

Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Кваліфікаційна робота
магістра
з теми: «МОДЕЛЬ ЕЛЕКТРОННОГО ОНЛАЙН СЕРЕДОВИЩА
«РЕЙТИНГ НПП»»

Виконав: студент групи KN1-M23
спеціальності 122 Комп'ютерні науки

Чорний Ярослав Сергійович

Керівник:

Слободянюк О. В., доцент кафедри
комп'ютерних наук

Рецензент:

Оптасюк С. В., кандидат фізико-
математичних наук, доцент,
доцент кафедри фізики

Кам'янець-Подільський 2024

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ФОРМУВАННЯ ЕЛЕКТРОННОГО ОНЛАЙН СЕРЕДОВИЩА.....	6
1.1. Аналіз сучасних підходів до оцінювання науково-педагогічних працівників.....	6
1.2. Теоретичні основи створення електронних онлайн середовищ	13
1.3. Вимоги до моделі рейтингового оцінювання НПП.....	18
Висновки до 1 розділу	23
РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДЕЛІ ЕЛЕКТРОННОГО ОНЛАЙН СЕРЕДОВИЩА.....	24
2.1 Огляд існуючих мобільних застосунків для створення електронних кабінетів НПП.....	24
2.2 Проектування моделі рейтингового середовища	31
2.3 Опис мобільного додатку для визначення рейтингу науково- педагогічних працівників	35
Висновки до 2 розділу	47
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК.....	51

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій сприяє значному вдосконаленню процесів управління в усіх сферах діяльності, включаючи освіту. У контексті цифрової трансформації особливе значення набувають автоматизовані системи, що дозволяють ефективно оцінювати якість науково-педагогічної діяльності. Формування рейтингу науково-педагогічних працівників (НПП) є важливим завданням, яке має забезпечити прозорість, об'єктивність і стимулювати професійний розвиток.

Наразі у багатьох закладах вищої освіти України відсутні сучасні інструменти, що дозволяють проводити комплексний моніторинг та рейтингування НПП. Це знижує ефективність управління кадровими ресурсами, уповільнює процес вдосконалення освітніх програм і не сприяє формуванню конкурентоспроможності закладу освіти на національному та міжнародному рівнях.

Розробка електронного онлайн середовища для рейтингового оцінювання НПП дозволить оптимізувати цей процес, зробивши його прозорим і доступним для всіх зацікавлених сторін. Удосконалення методів рейтингового оцінювання відповідає потребам реформування вищої освіти, орієнтованої на європейські стандарти якості.

Таким чином, актуальність теми обумовлена необхідністю впровадження інноваційних підходів до оцінювання діяльності НПП через використання сучасних інформаційних технологій, що сприятиме підвищенню якості освіти та наукових досліджень.

Мета дослідження полягає у створенні моделі електронного онлайн середовища для об'єктивного оцінювання та формування рейтингу науково-педагогічних працівників, що забезпечує прозорість і сприяє підвищенню ефективності їхньої діяльності.

Для досягнення мети роботи поставлено такі **завдання**:

1. Провести аналіз існуючих методів і систем рейтингового оцінювання НПП, визначивши ключові критерії, метрики та алгоритми для

створення моделі рейтингу.

2. Розробити концептуальну модель електронного онлайн середовища та реалізувати програмну частину розробленої моделі.
3. Провести тестування системи на практичних прикладах.
4. Оцінити ефективність розробленого середовища та визначити перспективи його використання.

Об'єкт дослідження – процес рейтингового оцінювання науково-педагогічних працівників у системі вищої освіти.

Предмет дослідження – модель електронного онлайн середовища, призначена для автоматизації та оптимізації рейтингового оцінювання НПП.

Методи дослідження. У роботі використано такі методи:

- Аналіз літературних джерел для вивчення сучасних підходів до оцінювання НПП.
- Моделювання для розробки концепції онлайн середовища.
- Програмна розробка для створення прототипу електронної системи.
- Методи тестування програмного забезпечення для перевірки працездатності та відповідності системи поставленим вимогам.
- Статистичний аналіз для оцінки результатів роботи середовища та порівняння його ефективності з існуючими підходами.

Наукова новизна роботи полягає у розробці нової моделі електронного середовища для об'єктивного і прозорого рейтингового оцінювання НПП з використанням сучасних інформаційних технологій та адаптації алгоритмів ранжування до специфіки освітньої сфери.

Практичне значення роботи полягає у можливості впровадження розробленої системи у закладах вищої освіти для:

- підвищення прозорості оцінювання працівників,
- стимулювання професійного зростання НПП,
- оптимізації процесів управління кадровими ресурсами.

Структура роботи. Робота складається зі вступу, двох розділів, висновків, списку використаних джерел та додатків. У першому розділі

висвітлюються теоретичні засади створення електронного онлайн середовища, розглядаються існуючі підходи до оцінювання НПП та визначаються вимоги до розроблюваної моделі.

Другий розділ присвячений практичній реалізації моделі, включаючи проектування, програмну розробку, тестування та оцінку ефективності системи. У висновках підбиваються підсумки дослідження, узагальнюються результати та визначаються перспективи подальшої роботи у цьому напрямку.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ФОРМУВАННЯ ЕЛЕКТРОННОГО ОНЛАЙН СЕРЕДОВИЩА

1.1. Аналіз сучасних підходів до оцінювання науково-педагогічних працівників

Оцінювання ефективності праці науково-педагогічних працівників (НПП) є одним із важливих інструментів підвищення якості освіти та наукових досліджень у закладах вищої освіти (ЗВО). Сучасні методики оцінювання спрямовані на забезпечення об'єктивності, прозорості та стимулювання професійного розвитку НПП у закладах вищої освіти. У цьому параграфі ми проводимо аналіз ключових підходів до оцінки ефективності праці НПП у вітчизняних та міжнародних практиках, розкриваємо їх переваги, та досліджуємо недоліки та можливості адаптації.

Проаналізуємо основні принципи оцінювання ефективності праці НПП в Україні.

Оцінка діяльності НПП в закладах вищої освіти в Україні побудована за такими основними принципами:

1. Прозорість, як важливий принцип, за яким усі критерії мають бути зрозумілими та доступними для працівників.
2. Комплексність: оцінка повинна враховувати всі аспекти діяльності такі як: викладання, наукову роботу, управлінську активність, участь у проєктах тощо.
3. Регулярність, щодо процесу оцінювання проводиться систематично, що дозволяє фіксувати динаміку розвитку.
4. Об'єктивність для використання кількісних та якісних показників, що мінімізують суб'єктивність. [4]

Розглянемо класифікацію методик оцінювання НПП у закладах вищої освіти. [2]

Методики оцінювання поділяються на декілька важливих категорій:

Кількісні методи:

- Вимірювання публікаційної активності (індекс цитування, кількість наукових статей, участь у конференціях).
- Аналіз викладацького навантаження (кількість навчальних годин, розробка навчально-методичних матеріалів).
- Оцінка залучення до міжнародних проєктів.

Якісні методи:

- Експертне оцінювання (оцінка роботи НПП комісією або зовнішніми експертами).
- Опитування студентів щодо якості викладання.
- Оцінка впливу на наукову спільноту (участь у наукових радах, редколегіях).

Змішані методи:

- Комплексні рейтингові системи, що поєднують кількісні та якісні показники.
- Використання балансового підходу: збір балів за різні види діяльності. Розглянемо міжнародні практики оцінювання ефективності праці НПП. У різних країнах Європи та Азії використовуються свої підходи до оцінки ефективності НПП [7-10]. Наприклад:

США:

- Основна увага приділяється публікаційній активності.
- Використовуються спеціалізовані бази даних, такі як Scopus та Web of Science, для оцінки індексів цитування.
- Студентські опитування є важливим інструментом оцінки якості викладання.

Європейський Союз:

- Важливим аспектом є участь НПП у міжнародних проєктах і грантах.
- Значну увагу приділяють якості викладання, що перевіряється через незалежні зовнішні аудити.

Азійські країни (Японія, Китай):

- Інтеграція академічних і прикладних досліджень є ключовим критерієм.
- Велике значення має розробка інновацій та їх практичне впровадження.

Проаналізуємо вітчизняні підходи до оцінювання праці НПП в Україні.

В Україні також існує декілька підходів до оцінювання НПП [2-5]:

Публікаційна активність.

- Значний акцент на кількості публікацій у виданнях, включених до баз Scopus і Web of Science.

- Оцінка наукової роботи за кількістю цитувань.

Методики Міністерства освіти і науки України.

- Використання уніфікованих критеріїв, таких як участь у проєктах, захист дисертацій, навчальна діяльність.

- Оцінка викладацького навантаження відповідно до державних стандартів.

Внутрішні системи ЗВО.

- У багатьох університетах впроваджуються власні системи рейтингового оцінювання.

- Урахування специфіки навчальних дисциплін і профілю закладу освіти.

Розглянемо переваги та недоліки які можуть виникнути до існуючих підходів оцінювання НПП у вищих навчальних закладах України.

Що до основних переваг це прозорість та можливість порівняння результатів між різними НПП. Стимулювання наукової активності через мотивацію до публікацій та участі у грантах. Можливість адаптації під конкретні потреби закладу вищої освіти.

До основних недоліків можна віднести це залежність від зовнішніх баз даних, таких як Scopus або Web of Science, що обмежує можливості локальних публікацій. Суб'єктивність студентських опитувань, які можуть не завжди

об'єктивно відображати якість викладання. Недостатня увага до прикладної наукової діяльності та інновацій.

Які на думку науковців існують перспективи удосконалення методик оцінювання НПП у ВНЗ України.

Для підвищення ефективності оцінювання необхідно [3]:

1. Інтегрувати інструменти автоматизації, зокрема електронні системи збору та аналізу даних.
2. Враховувати особливості міждисциплінарної діяльності НПП.
3. Розширити спектр критеріїв для оцінки, зокрема через оцінювання соціального впливу наукової діяльності.
4. Забезпечити адаптивність системи оцінювання до змін у вимогах та викликах часу.

Грунтовний аналіз існуючих методик оцінювання праці НПП демонструє широкий спектр підходів, які базуються на кількісних та якісних показниках. Водночас для забезпечення об'єктивності та прозорості необхідно впроваджувати нові моделі, що використовують можливості сучасних інформаційних технологій. Розробка електронного онлайн середовища для рейтингового оцінювання є важливим кроком у цьому напрямку, що дозволить врахувати переваги існуючих методик і подолати їх недоліки [7].

Сучасні рейтингові системи оцінювання науково-педагогічних працівників (НПП) використовуються як у вітчизняних, так і в закордонних закладах освіти для підвищення якості освіти, стимулювання професійного розвитку викладачів та забезпечення конкурентоспроможності університетів. Цей параграф присвячено аналізу підходів до рейтингової оцінки НПП в українських та зарубіжних університетах, визначенню ключових подібностей і відмінностей, а також вивченню перспектив адаптації кращих міжнародних практик в умовах України [2].

Розкриємо ключові критерії рейтингової оцінки НПП у вітчизняних та зарубіжних системах

Наукова діяльність.

- Вітчизняна практика базується, що на основному акценті робиться на кількість публікацій у журналах, індексованих у Scopus та Web of Science, монографіях, участі у міжнародних та всеукраїнських конференціях.

- Зарубіжна практика базується окрім кількості публікацій, враховується індекс Хірша, якість видань (Q-квартиль), міжнародне співробітництво, залученість до грантових проєктів.

Викладацька діяльність.

- Вітчизняна практика, що оцінюється навантаженням (кількість лекцій, практичних занять), створення навчально-методичних матеріалів, оцінки студентів.

- Зарубіжна практика, що використовується при студентському опитуванні, відгуки випускників, оцінка інтерактивних методів викладання та впровадження інновацій у навчальний процес.

Інноваційна діяльність.

- Вітчизняна практика яка часто не є обов'язковим критерієм або враховується в загальних показниках.

- Зарубіжна практика яка оцінює комерціалізацію наукових досліджень, патентів, впровадження інновацій у суспільне життя.

Участь у міжнародних проєктах.

- Вітчизняна практика яка має обмежене застосування, здебільшого через недостатнє фінансування і слабку інтеграцію до міжнародного академічного середовища.

- Зарубіжна практика де високо цінується участь у проєктах Horizon 2020, Erasmus+, дослідницьких консорціумах.

Розглянемо особливості систем рейтингового оцінювання у зарубіжних університетах. [9-12]

Розглянемо на прикладі США де основна увага приділяється науковим досягненням через такі метрики, як індекс цитування, участь у грантових програмах, викладацька діяльність та оцінка студентів. Основний акцент на міждисциплінарності досліджень та їхньому практичному значенні.

У Європейському Союзі використовуються інтегровані підходи до оцінки НПП: вагоме значення мають соціальний вплив досліджень, комерціалізація результатів, співпраця з бізнесом і державними органами. Використання європейських рейтингів, таких як Times Higher Education та QS, які враховують продуктивність НПП як один із ключових показників.

В Азійських країнах таких як Японія, Китай, Південна Корея великий акцент на технологічному прогресі та впровадженні інновацій у суспільство. Підтримка державою публікацій у високореєтингових журналах як показника конкурентоспроможності університету.

Розглянемо основні переваги та недоліки вітчизняних систем рейтингового оцінювання НПП у вищих навчальних закладах України:

До переваг, на нашу думку, можна віднести:

- Спрямованість на розвиток наукової діяльності, публікацій і викладацької роботи.

- Формування внутрішніх рейтингів у багатьох університетах.

- Спрощена система звітності для моніторингу досягнень НПП.

До недоліків, на нашу думку, можна віднести:

- Недостатня увага до комерціалізації досліджень.

- Обмежена інтеграція до міжнародних рейтингових систем через низький рівень залучення до грантових програм.

- Мало уваги приділяється якісним показникам, зокрема студентським опитуванням чи соціальному впливу.

Які ж існують важливі відмінності між вітчизняними та зарубіжними системами оцінювання ПНН:

Ступінь автоматизації.

- Вітчизняні системи переважно базуються на ручному введенні даних, що підвищує ймовірність помилок.

- Зарубіжні системи активно використовують автоматизовані платформи для збору й аналізу даних.

Кількісні та якісні показники.

- В Україні основний акцент робиться на кількісних показниках.
- За кордоном вагомість якісних показників, таких як інноваційність та міждисциплінарність, значно вища.

Фінансова підтримка.

- Вітчизняні університети мають обмежений доступ до фінансових ресурсів для стимулювання наукової діяльності.
- У зарубіжних ЗВО значна увага приділяється грантовим програмам.

Розкриємо вплив рейтингів на професійний розвиток НПП.

У зарубіжних ЗВО рейтинги часто використовуються як інструмент мотивації до наукової діяльності та викладання через надання грантів, премій, можливостей для стажування.

Вітчизняні ЗВО переважно використовують рейтинги як механізм контролю, що може сприйматися викладачами як формальний процес, без значного впливу на кар'єрне зростання.

Які ж перспективи адаптації міжнародного досвіду в Україні, на нашу думку, можна виокремити?

Для підвищення ефективності вітчизняних рейтингів НПП доцільно:

- Автоматизувати процеси оцінювання через впровадження сучасних онлайн платформ.
- Розширити перелік показників, включивши критерії соціального впливу, інноваційності та міждисциплінарності.
- Підвищити залучення до міжнародних програм, що дозволить отримати доступ до кращих практик і фінансування.
- Мотивувати НПП до участі у грантових проєктах через створення систем преміювання.

Порівняння систем рейтингового оцінювання у вітчизняних та зарубіжних ЗВО свідчить про суттєві відмінності в підходах, орієнтованості на кількісні та якісні показники, рівні автоматизації. Використання кращих міжнародних практик у поєднанні з урахуванням місцевих умов може сприяти вдосконаленню систем оцінювання праці НПП в Україні.

1.2. Теоретичні основи створення електронних онлайн середовищ

Інформаційні системи (ІС) є однією з невід'ємних частин сучасного суспільства, що забезпечує ефективну обробку, зберігання та передачу даних. Вони, в свою чергу охоплюють широкий спектр сфер діяльності, включаючи управління бізнесом, освіту, медицину та науку. Для забезпечення такої ефективності, гнучкості та стійкості інформаційних систем важливо дотримуватися низки фундаментальних принципів, які визначають їхню архітектуру, функціональність і взаємодію з користувачами.

До основних принципів побудови інформаційних систем можна віднести наступні[12]:

Принцип системності:

- Інформаційна система розглядається як єдине ціле, що складається з взаємопов'язаних компонентів: апаратного забезпечення, програмного забезпечення, баз даних, користувачів і процесів.
- Забезпечується координація всіх елементів для досягнення загальної мети.

Принцип відкритості:

- ІС повинна мати можливість інтегруватися з іншими системами та адаптуватися до змін у зовнішньому середовищі.
- Наприклад, інтеграція з API або підключення до зовнішніх баз даних.

Принцип гнучкості:

- Система має бути здатна до модифікації та розширення без значних витрат і часу.
- Це досягається за рахунок модульної архітектури, що дозволяє додавати або замінювати окремі компоненти.

Принцип безпеки:

- Інформаційна система повинна гарантувати захист даних від несанкціонованого доступу, втрати чи модифікації.
- Використання шифрування, багаторівневої аутентифікації та регулярного аудиту системи.

Принцип надійності:

- Забезпечення стабільної роботи системи навіть у разі збою окремих компонентів.
- Реалізація резервного копіювання, відновлення даних і дублювання критично важливих елементів.

Принцип ергономічності:

- ІС повинна бути зручною для користувачів, із простим і зрозумілим інтерфейсом.
- Це сприяє зменшенню помилок і підвищенню ефективності роботи з системою.

Принцип економічності:

- Побудова системи має бути обґрунтована з точки зору витрат на її розробку, впровадження та експлуатацію.
- Доцільність використання ресурсів оцінюється на етапі проектування.

Принцип масштабованості:

- Система повинна мати можливість працювати із збільшеним обсягом даних або кількістю користувачів без значного зниження продуктивності.
- Наприклад, використання хмарних технологій для зберігання даних.

Розглянемо основні компоненти, які реалізують описані принципи побудови інформаційних систем [13]:

Апаратне забезпечення – сервери, робочі станції, мережеве обладнання забезпечують фізичну інфраструктуру для роботи системи.

Програмне забезпечення – операційні системи, прикладні програми та інтерфейси користувача реалізують функціональність системи: приклад: платформи управління базами даних (MySQL, Oracle).

Бази даних – організоване зберігання інформації, що забезпечує швидкий доступ і обробку даних; дотримання принципу надійності завдяки резервному копіюванню.

Користувачі – люди, які взаємодіють із системою, забезпечують її продуктивне використання.

Алгоритми та процеси реалізують логіку обробки інформації, автоматизують повторювані завдання.

Розробка інформаційних систем із використанням сучасних технологій складається з наступних основних важливих процесів[16]:

Хмарні обчислення які дозволяють реалізувати принципи масштабованості, надійності та економічності на прикладі Amazon Web Services (AWS), Microsoft Azure.

Модульна архітектура для застосування мікросервісів для створення гнучких і адаптивних систем.

Машинне навчання та штучний інтелект які використовуються для автоматизації аналізу великих обсягів даних, прогнозування та оптимізації процесів.

Інтернет речей (IoT) за допомогою якої відбувається інтеграція пристроїв із системою для збору й обробки даних у реальному часі.

Блокчейн – це реалізація принципу безпеки через створення незмінної бази даних транзакцій.

Деякі науковці [9-14] вказують на виклики які виникають при побудові інформаційних систем. Такі як:

Комплексність інтеграції – це взаємодія з іншими системами може вимагати значних витрат часу і ресурсів.

Кіберзагрози – це коли постійна еволюція загроз вимагає впровадження сучасних методів захисту.

Швидкий розвиток технологій потребує постійно є необхідність модернізації систем для відповідності сучасним стандартам.

Принципи побудови інформаційних систем визначають їхню ефективність, зручність та адаптивність до змін. Дотримання цих принципів забезпечує створення систем, що відповідають сучасним потребам суспільства та бізнесу. Урахування новітніх технологій, таких як хмарні обчислення, штучний інтелект і блокчейн, дозволяє створювати гнучкі, надійні та економічно ефективні рішення.

Електронне середовище в освітніх установах виконує важливу роль у забезпеченні якості освітнього процесу, зручності управління, доступності інформації для студентів і викладачів. Кам'янець-Подільський національний університет імені Івана Огієнка є прикладом сучасного ЗВО, де впроваджено низку цифрових інструментів для підтримки навчання та організації освітньої діяльності.

Розглянемо основні складові електронного середовища освітніх установ.

Сучасне електронне середовище в Українських університетах включає [15]:

1. Системи управління навчальним процесом (LMS):

- Використовуються для організації онлайн-навчання, ведення розкладу, оцінювання студентів.

- У Кам'янець-Подільському національному університеті імені Івана Огієнка застосовується платформа Moodle, яка дозволяє створювати курси, інтегрувати мультимедійні матеріали та проводити дистанційне навчання.

2. Електронні бібліотеки та репозиторії:

- Доступ до наукових публікацій, підручників, монографій.
- В університеті функціонує електронна бібліотека, що містить навчальні матеріали та роботи викладачів.

3. Інструменти для адміністративного управління:

- Автоматизовані системи обліку студентів, формування розкладу, звітності.
- Реалізована система деканату для відстеження успішності студентів та ведення документації.

4. Комунікаційні платформи:

- Інструменти для швидкої взаємодії між студентами, викладачами та адміністрацією (наприклад, Google Workspace, Microsoft Teams).

- Кам'янець-Подільський університет активно використовує електронну пошту та внутрішній портал для спілкування.

5. Інструменти для моніторингу та оцінювання знань:

- Автоматизовані системи тестування та оцінювання знань.
- Викладачі університету активно використовують Moodle для створення тестів, контрольних і практичних робіт.

Розкриємо основні особливості впровадження електронного середовища у Кам'янець-Подільському національному університеті імені Івана Огієнка.

1. Інтеграція традиційних і цифрових методів навчання:

- Університет поєднує традиційне навчання з онлайн-інструментами. Наприклад, лекції проводяться офлайн, але студенти мають доступ до записів і додаткових матеріалів на Moodle.

2. Адаптація під час пандемії COVID-19 та війни з росією:

- У період пандемії та військової агресії росії наш університет швидко перейшов до дистанційного навчання завдяки готовності електронного середовища.

- Активно використовувалися відеоконференції через Zoom, Google Meet, а також платформа Moodle для організації повноцінного навчального процесу.

3. Цифровізація адміністративних процесів:

- Впроваджено систему «Електронний кабінет студента», де доступна інформація про розклад, оцінки, борги та стан навчання.
- Для викладачів доступні електронні журнали, що спрощують оцінювання та звітність.

4. Розвиток електронної бібліотеки:

- Університет надає доступ до електронних ресурсів, зокрема до наукових баз даних та репозиторіїв.
- Електронна бібліотека сприяє підвищенню якості підготовки студентів і викладачів до навчального процесу.

5. Підтримка академічної доброчесності:

- Застосовуються системи перевірки текстів на плагіат, такі як Unicheck, для контролю якості наукових і навчальних робіт.
- Це дозволяє підтримувати високий рівень академічної доброчесності.

Які основні переваги електронного середовища для Кам'янець-Подільського університету ми можемо виокремити:

- Доступність навчальних ресурсів – студенти та викладачі мають постійний доступ до матеріалів незалежно від місця перебування.
- Ефективність комунікації – швидкий обмін інформацією через електронну пошту та внутрішні системи.
- Прозорість та контроль – впровадження електронних журналів та тестування знижує ймовірність суб'єктивності оцінювання.
- Індивідуалізація навчання – студенти можуть самостійно працювати з додатковими матеріалами, коригувати темп навчання.
- Скорочення паперової роботи – адміністративні процеси значно спрощені через автоматизацію документів.

Які ж головні перспективи розвитку електронного середовища сучасного вищого навчального закладу в Україні ми вбачаємо:

1. Інтеграція штучного інтелекту – використання AI для автоматизації процесів навчання та оцінювання.
2. Розширення доступу до міжнародних платформ – надання студентам доступу до міжнародних освітніх програм і наукових баз.
3. Мобільні додатки – створення мобільного застосунку для управління навчальним процесом.
4. Аналіз даних – впровадження систем аналітики для відстеження прогресу студентів і ефективності викладання.

Електронне середовище Кам'янець-Подільського національного університету імені Івана Огієнка є прикладом ефективного впровадження сучасних інформаційних технологій у навчальний процес. Воно забезпечує доступність, прозорість і гнучкість освіти.

1.3. Вимоги до моделі рейтингового оцінювання НПП

Ефективність науково-педагогічної діяльності (НПД) є одним із ключових чинників забезпечення високої якості освіти та науки в закладах вищої освіти України. Оцінка ефективності діяльності науково-педагогічних

працівників (НПП) базується на показниках, що відображають результати їхньої роботи у навчальній, науковій, інноваційній і громадській сферах. Розуміння цих показників є важливим як для самого НПП, так і для адміністрації ЗВО, що прагне підвищити конкурентоспроможність закладу.

Розглянемо основні категорії показників ефективності НПД у вищих навчальних закладах України.

Показники ефективності НПД можна умовно поділити на декілька категорій:

1. Наукова діяльність:

- Кількість публікацій. Враховується кількість статей у фахових виданнях, зокрема індексованих у Scopus, Web of Science.
- Індекс цитування. Відображає науковий вплив через цитування робіт іншими дослідниками.
- Участь у грантових проєктах. Показує здатність залучати фінансування для проведення досліджень.
- Розробка нових знань. Оцінюється через патенти, монографії, наукові розробки.

2. Викладацька діяльність:

- Навчальне навантаження. Враховується кількість лекційних і практичних занять.
- Рівень задоволеності студентів. Опитування студентів дозволяють оцінити якість викладання.
- Розробка навчально-методичних матеріалів. Оцінюється створення курсів, посібників, програм.

3. Інноваційна діяльність:

- Впровадження інновацій у навчальний процес. Наприклад, створення інтерактивних курсів або використання нових методів викладання.
- Комерціалізація наукових розробок. Приклади впровадження нових технологій у промисловість.

4. Громадська діяльність:

- Участь у роботі кафедр і факультетів. Організаційна робота, участь у вчених радах.

- Популяризація науки. Проведення відкритих лекцій, участь у конференціях.

- Робота з абітурієнтами. Участь у профорієнтаційних заходах.

5. Особистий професійний розвиток:

- Участь у семінарах, тренінгах, підвищенні кваліфікації.

- Міжнародна мобільність. Стажування в іноземних університетах, участь у міжнародних програмах.

Які кількісні показники ефективності НПД можна виокремити:

1. Наукові показники:

- Кількість статей у журналах: Scopus, Web of Science, Index Copernicus.

- Індекс Хірша (h-index), який поєднує кількість публікацій і цитувань.

- Кількість доповідей на конференціях.

2. Педагогічні показники:

- Обсяг навчального навантаження у годинах.

- Кількість студентів, які успішно захистили дипломні та магістерські роботи.

3. Інноваційні показники:

- Кількість впроваджених у навчальний процес методик, цифрових технологій.

- Обсяг залученого фінансування через гранти або проекти.

4. Організаційні показники:

- Кількість заходів (семінарів, конференцій), організованих НПП.

- Кількість опублікованих навчально-методичних матеріалів.

Кількісні показники мають об'єктивний характер, але не враховують якість результатів.

Якісні показники ефективності НПД такі як :

1. Якість викладання:

- Відгуки студентів про якість викладання, наприклад, за допомогою анонімних анкетувань.

- Аналіз інтерактивних методів, використаних у навчанні.

2. Науковий вплив:

- Цитованість у провідних журналах і вплив наукових праць на розвиток галузі.

- Участь у редколегіях наукових видань.

3. Інноваційність:

- Ступінь впровадження нових методик викладання або технічних засобів.

- Вплив інновацій на рівень засвоєння матеріалу студентами.

4. Соціальний вплив:

- Участь у соціальних проєктах, пов'язаних із наукою чи освітою.

- Проведення відкритих лекцій, вебінарів, публікацій у популярних медіа.

Якісні показники надають глибший аналіз, але можуть бути суб'єктивними.

Сучасні системи оцінювання, зокрема у закладах вищої освіти в Україні, все частіше використовують комплексні підходи.

Рейтингові системи де НПП набирають бали за різні види діяльності (публікації, викладання, організаційна робота). Приклад: внутрішні рейтинги кафедр або факультетів.

Моделі збалансованих показників (Balanced Scorecard), що розробляються метрики для оцінки всіх сфер діяльності НПП; враховується взаємозв'язок між показниками (наприклад, наука і викладання).

Електронні платформи для автоматизації збору та аналізу даних про ефективність роботи; університети використовують LMS (наприклад, Moodle) для аналізу участі викладачів у навчальному процесі.

Проблеми визначення показників ефективності НПД: суб'єктивність оцінювання (якісні показники можуть бути суб'єктивно трактовані).

Обмеженість ресурсів як недостатнє фінансування обмежує можливості НПП для участі у міжнародних проєктах.

Складність інтеграції нових методик: впровадження інновацій може потребувати часу і технічної підтримки.

Формалізація оцінювання: надмірний акцент на кількісних показниках може знижувати мотивацію до якісної роботи.

Які, на нашу думку, перспективи вдосконалення системи оцінювання НПД в сучасному ЗВО:

Інтеграція автоматизованих систем.

- Використання електронних платформ для збору даних і створення прозорих рейтингів.

Врахування соціального впливу.

- Оцінка внеску НПП у розвиток місцевої спільноти, популяризацію науки.

Гнучкі критерії оцінювання.

- Адаптація показників до специфіки діяльності НПП і цілей університету.

Міжнародна співпраця.

- Запозичення кращих практик оцінювання з провідних університетів світу.

Визначення показників ефективності НПД є складним і багатогранним процесом. Ефективна система оцінювання повинна враховувати як кількісні, так і якісні показники, а також бути прозорою, гнучкою та адаптованою до потреб конкретного ЗВО. Підвищення ефективності оцінювання сприятиме стимулюванню професійного розвитку НПП та підвищенню конкурентоспроможності університетів.

Висновки до 1 розділу

В першому розділі проведено аналіз сучасних підходів до оцінювання науково-педагогічних працівників. Обґрунтована оцінка діяльності НПП в закладах вищої освіти в Україні побудована за такими основними принципами:

1. Прозорість, як важливий принцип, за яким усі критерії мають бути зрозумілими та доступними для працівників.
2. Комплексність: оцінка повинна враховувати всі аспекти діяльності такі як: викладання, наукову роботу, управлінську активність, участь у проєктах тощо.
3. Регулярність, щодо процесу оцінювання проводиться систематично, що дозволяє фіксувати динаміку розвитку.
4. Об'єктивність для використання кількісних та якісних показників, що мінімізують суб'єктивність.

Розкриті теоретичні основи створення електронних онлайн середовищ.

Електронне середовище в освітніх установах виконує важливу роль у забезпеченні якості освітнього процесу, зручності управління, доступності інформації для студентів і викладачів. Кам'янець-Подільський національний університет імені Івана Огієнка є прикладом сучасного ЗВО, де впроваджено низку цифрових інструментів для підтримки навчання та організації освітньої діяльності. Виокремлені вимоги до моделі рейтингового оцінювання НПП.

РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДЕЛІ ЕЛЕКТРОННОГО ОНЛАЙН СЕРЕДОВИЩА

2.1 Огляд існуючих мобільних застосунків для створення електронних кабінетів НПП

КNU Online (КНУ ім. Тараса Шевченка)

Універсальний мобільний додаток із функціоналом для студентів і викладачів. **Електронний кабінет викладача** дозволяє завантажувати навчальні матеріали (тексти лекцій, презентації, завдання), створювати тести та кейси для занять. Також додаток інтегрує цифрову бібліотеку, онлайн-звітність і документообіг, що спрощує роботу НПП та взаємодію зі студентами

КNU Online: Як працює мобільний додаток

КNU Online — це мобільний додаток Київського національного університету імені Тараса Шевченка, розроблений для автоматизації освітніх процесів. Він забезпечує доступ до електронних кабінетів викладачів і студентів, цифрової бібліотеки, електронного документообігу, а також полегшує комунікацію між усіма учасниками освітнього процесу.

Основний функціонал для НПП

1. Електронний кабінет викладача:

- **Завантаження матеріалів:** викладач може додавати презентації, текст лекцій, завдання, тести й інші матеріали.
- **Оцінювання студентів:** простий інтерфейс для виставлення оцінок у цифровому журналі.
- **Відстеження успішності:** перегляд статистики успішності та відвідуваності студентів.
- **Екзаменаційні відомості:** доступ до розкладів і результатів екзаменів.
- **Комунікація:** месенджер для взаємодії зі студентами та адміністрацією.

2. Планування навчального процесу:

- Викладачі отримують персоналізований розклад занять.

- Можливість реєструвати заміни викладачів.

3. Цифрова бібліотека:

- Доступ до наукової та навчальної літератури, яку можна використовувати у курсах.

- Функція швидкого пошуку по бібліотечному каталогу.

4. Онлайн-звітність:

- Викладач може завантажувати звіти про виконану роботу та аналізувати свої результати.

- Інтеграція з системою цифрового документообігу університету.

Інтеграція з іншими системами

Електронний деканат:

- Автоматичний облік навчальних годин.
- Розподіл навантаження між викладачами.

Онлайн-звіти:

- Прямий зв'язок із адміністрацією університету для надання звітності.

Оцінювання студентами:

- Викладачі можуть бачити анонімні відгуки про якість свого викладання.

JetIQ (Вінницький національний технічний університет)

Система пропонує електронний кабінет викладача як частину комплексної платформи для управління освітнім процесом. Функціонал включає:

- Електронний журнал успішності студентів.
- Управління тестами, навчальними матеріалами та корпоративною поштою.

- Модуль комунікації зі студентами через вбудований месенджер.

- Інструменти для створення електронних книг та матеріалів для самостійного навчання.

JetIQ: Як працює система

JetIQ — це цифрова система управління освітнім процесом, створена у Вінницькому національному технічному університеті (ВНТУ). Вона забезпечує автоматизацію навчального процесу як для студентів, так і для викладачів, інтегруючи в собі електронний деканат, модуль оцінювання, бібліотеки та інші функціональні можливості.

Основний функціонал JetIQ

1. Електронний кабінет викладача

Електронний журнал:

- Викладач може виставляти оцінки студентам за дисциплінами.
- Відстежувати відвідуваність занять і аналізувати успішність.

Управління навчальними матеріалами:

- Завантаження лекцій, тестів, презентацій та іншого контенту в персональний репозиторій.
- Модуль "Test-IQ" дозволяє створювати автоматизовані тести для студентів.

Комунікація зі студентами:

- Можливість надсилати повідомлення через вбудований месенджер.
- Доступ до загальної корпоративної електронної пошти.

Звітування:

- Формування звітів для адміністрації.
- Автоматичне обчислення навчального навантаження.

2. Електронний деканат

Автоматизоване управління навчальними потоками:

- Розклад занять та екзаменів.
- Реєстрація дисциплін та оцінок.

Моніторинг роботи викладачів:

- Відстеження виконання навчального плану.
- Оцінка активності викладачів.

3. Для студентів

Доступ до навчальних матеріалів:

- Всі завдання, тести, лекції доступні в електронному вигляді.
- Персоналізоване середовище для дистанційного навчання.

Розклад занять:

- Інтегрований календар із зазначенням дисциплін, викладачів і локацій.

Електронна залікова книжка:

- Поточні результати та підсумкові оцінки за дисциплінами.

4. Інтеграція з іншими сервісами

Наукометрія:

- Система підтримує інтеграцію з наукометричними базами, що дозволяє відображати наукову активність викладачів.

Управління кафедрою:

- Адміністрація може оновлювати інформацію про навчальні ресурси кафедри, вести звітність і комунікувати з викладачами.

АСУ ВНЗ (Автоматизована система управління ВНЗ)

Ця система пропонує електронний кабінет викладача для перегляду та редагування журналів, відміток про присутність студентів і виставлення оцінок. Кабінет також надає доступ до персоналізованого розкладу занять і екзаменаційних відомостей, що зручно для управління навчальним процесом

Можливості АСУ ВНЗ (Автоматизованої системи управління ВНЗ)

АСУ ВНЗ — це програмно-інформаційна система, створена для автоматизації освітніх процесів та управління діяльністю закладів вищої освіти. Вона включає модулі для роботи з навчальним процесом, документами та комунікацією між адміністрацією, викладачами та студентами.

Основний функціонал АСУ ВНЗ

1. Електронний кабінет викладача

Робота з журналами:

- Виставлення оцінок студентам.
- Відмітки про присутність на заняттях.
- Примітки для студентів із рекомендаціями або коментарями.

Розклад занять:

- Доступ до персоналізованого розкладу із зазначенням аудиторій та часу.
- Інструмент для заміни занять або зміни графіків.

Формування навчальних матеріалів:

- Завантаження лекцій, презентацій, завдань і тестів.

Аналітика:

- Відображення успішності та відвідуваності студентів.

2. Електронний кабінет студента

Розклад занять:

- Інтеграція з календарем для зручного перегляду дисциплін.

Електронна залікова книжка:

- Поточні та підсумкові оцінки.
- Результати тестів і екзаменів.

Доступ до навчальних матеріалів:

- Завантаження лекцій, презентацій і завдань від викладачів.

Комунікація:

- Обмін повідомленнями з викладачами та деканатом.

3. Адміністративний модуль

Електронний деканат:

- Автоматизація розподілу навчального навантаження.
- Управління документами (довідки, заяви, запити).

Моніторинг роботи викладачів:

- Оцінка виконання навчального плану.
- Звіти про навантаження та продуктивність.

Документообіг:

- Автоматизація обробки внутрішніх документів, включно з електронними відомостями.

Підтримка сесій:

- Ведення екзаменаційних відомостей.
- Організація перездач і оцінювання.

4. Додаткові можливості

Статистика і звітність:

- Формування детальних звітів для адміністрації.

Інтеграція:

- Сумісність із іншими системами, такими як ЄДЕБО (Єдина державна електронна база з питань освіти).

Безпека даних:

- Захист інформації через централізоване управління доступом.

Цифровий кампус Університету Грінченка

Додаток з інтуїтивним інтерфейсом, де викладачі мають доступ до матеріалів курсу, онлайн-занять, бібліотеки, а також можливість підвищувати свою цифрову компетентність через спеціальні модулі навчання.

Цифровий кампус Університету Грінченка – це сучасна платформа, яка перетворює традиційний освітній процес на інтерактивний та доступний досвід. Завдяки використанню найновіших технологій, студенти та викладачі отримують широкий спектр можливостей для навчання, спілкування та співпраці.

Основні можливості Цифрового кампусу:

Система електронного навчання:

- **Електронні навчальні курси:** Широкий вибір курсів з різноманітних дисциплін, доступних онлайн у будь-який зручний час.
- **Персональні кабінети:** Кожен студент та викладач має свій особистий кабінет, де можна переглядати розклад, оцінки, завдання та спілкуватися з іншими учасниками навчального процесу.

- **Журнал оцінок:** Зручний інструмент для відстеження власних досягнень та прогресу в навчанні.
- **Каталог вибіркових дисциплін:** Можливість обрати додаткові курси за інтересами.
- **Мобільний додаток:** Доступ до всіх функцій системи з будь-якого пристрою.

Цифровий хаб інноваційних рішень:

- **Віртуальна та доповнена реальність:** Занурення в інтерактивні 3D-моделі та симуляції, що дозволяє глибше розуміти складні поняття та процеси.
 - **Лабораторії робототехніки, вбудованих систем та 3D-моделювання:** Можливість практичного застосування отриманих знань та розробки власних проектів.
 - **Студії дистанційного навчання:** Проведення онлайн-занять, вебінарів та конференцій.
 - **Мультимедійні простори:** Створення та презентація навчальних матеріалів у різних форматах.
- Інші можливості:
- **Сучасна цифрова бібліотека:** Швидкий доступ до величезної кількості електронних ресурсів.
 - **Інтегрований простір для спілкування та співпраці:** Форуми, чати, відеоконференції для обміну знаннями та ідеями.
 - **Індивідуальна траєкторія навчання:** Можливість скласти власний навчальний план відповідно до інтересів та потреб.

Принцип роботи Цифрового кампусу

Цифровий кампус працює на основі сучасних інформаційних технологій, що забезпечують:

- **Доступність:** Можливість навчатися в будь-який час і в будь-якому місці, маючи підключення до інтернету.
- **Інтерактивність:** Активну участь студентів у навчальному процесі через різноманітні інструменти та завдання.

- **Персоналізація:** Створення індивідуальної траєкторії навчання для кожного студента.
- **Ефективність:** Оптимізацію навчального процесу та підвищення його результативності.

Цифровий кампус Університету Грінченка – це не просто онлайн-платформа, а потужний інструмент для розвитку сучасного освітнього середовища. Завдяки своїм можливостям, він сприяє підвищенню якості освіти, розвитку творчих здібностей студентів та підготовці висококваліфікованих фахівців.

Ці системи показують приклади сучасних платформ для підтримки викладацької діяльності. Вони можуть бути основою для розробки вашого мобільного додатку, орієнтованого на конкретні потреби кафедри або університету.

2.2 Проектування моделі рейтингового середовища

Проектування моделі рейтингового середовища є важливим етапом розробки інформаційної системи (ІС), яка забезпечує об'єктивність, прозорість і автоматизацію оцінки науково-педагогічних працівників (НПП). Цей процес охоплює визначення функціональних компонентів, архітектури системи та способів взаємодії користувачів із системою. Розробка такої моделі вимагає врахування специфіки освітнього закладу, основних критеріїв оцінювання і технологічних можливостей.

Наведемо опис основних етапів проектування моделі рейтингового середовища НПП.

Для початку необхідно виконати аналіз вимог до майбутньої системи. Визначити потреби користувачів (адміністрація, викладачі, студенти), визначити основні функції додатку: збір даних, оцінювання, формування рейтингів, узгодити критеріїв і метрик для ранжування.

Наступним кроком буде формування концептуальної моделі, з визначенням ключових компонентів системи: бази даних, інтерфейсу

користувача, модулів аналізу, провести моделювання інформаційних потоків між компонентами.

Далі відбувається вибір технологій реалізації нашого застосунку. Потрібно брати мову програмування (наприклад, Python для аналітики, JavaScript для веб-інтерфейсу). Визначитись з використанням баз даних (наприклад, MySQL або PostgreSQL), з розробкою мобільного інтерфейсу.

Наведемо архітектуру інформаційної системи.

Архітектура рейтингового середовища може бути побудована за модульним підходом, що забезпечує гнучкість і масштабованість. До них відносять:

Клієнт-серверна архітектура.

Клієнтська частина (Frontend): Інтерфейс користувача (веб- або мобільний додаток). Реалізується з використанням HTML, CSS, JavaScript або фреймворків, таких як React чи Angular.

Серверна частина (Backend): Обробка запитів, обчислення рейтингів, управління даними. Реалізується на Python (Django/Flask), PHP або Node.js.

Рівні архітектури:

Презентаційний рівень: відповідає за взаємодію користувачів із системою.

Логічний рівень: виконує бізнес-логіку, наприклад, розрахунок рейтингу.

Рівень даних: зберігає інформацію про НПП, метрики та результати оцінювання.

Основні компоненти архітектури:

База даних: таблиці для зберігання інформації про НПП, критерії, оцінки, звіти, використання SQL або NoSQL баз даних.

Модуль збору даних: інтеграція із зовнішніми джерелами (Scopus, Web of Science) для автоматичного збору публікаційної активності.

Модуль аналізу: алгоритми обчислення рейтингу, зокрема нормалізація даних, вагові коефіцієнти.

Модуль звітності: генерація таблиць і графіків із результатами рейтингів.

Розкриємо інформаційні потоки у запропонованій системі:

Вхідні дані: дані від користувачів (адміністрація вводить критерії, НПП додають інформацію про наукову діяльність), автоматичний збір публікацій із наукових баз.

Обробка даних: алгоритми розрахунку рейтингу, нормалізація показників для забезпечення рівності умов.

Вихідні дані: рейтингові таблиці, графіки, звіти, боступ до результатів через веб-інтерфейс.

Особливості проектування інтерфейсу користувача:

1. Зручність і доступність:
 - Інтуїтивний дизайн для всіх категорій користувачів.
 - Адаптивність для мобільних пристроїв.
2. Розподіл доступу:
 - Різні рівні доступу для адміністрації, викладачів і студентів.
3. Інструменти візуалізації:
 - Таблиці, діаграми та графіки для представлення рейтингів і динаміки змін.

Виклики у проектуванні та шляхи їх вирішення

1. Складність збору даних:
 - Необхідність інтеграції з різними базами даних та системами.

Рішення – це використання API зовнішніх систем.

2. Безпека даних:
 - Захист конфіденційної інформації про НПП.

Рішення – це шифрування, багаторівнева аутентифікація.

3. Масштабованість:
 - Забезпечення роботи системи зі зростанням обсягу даних.

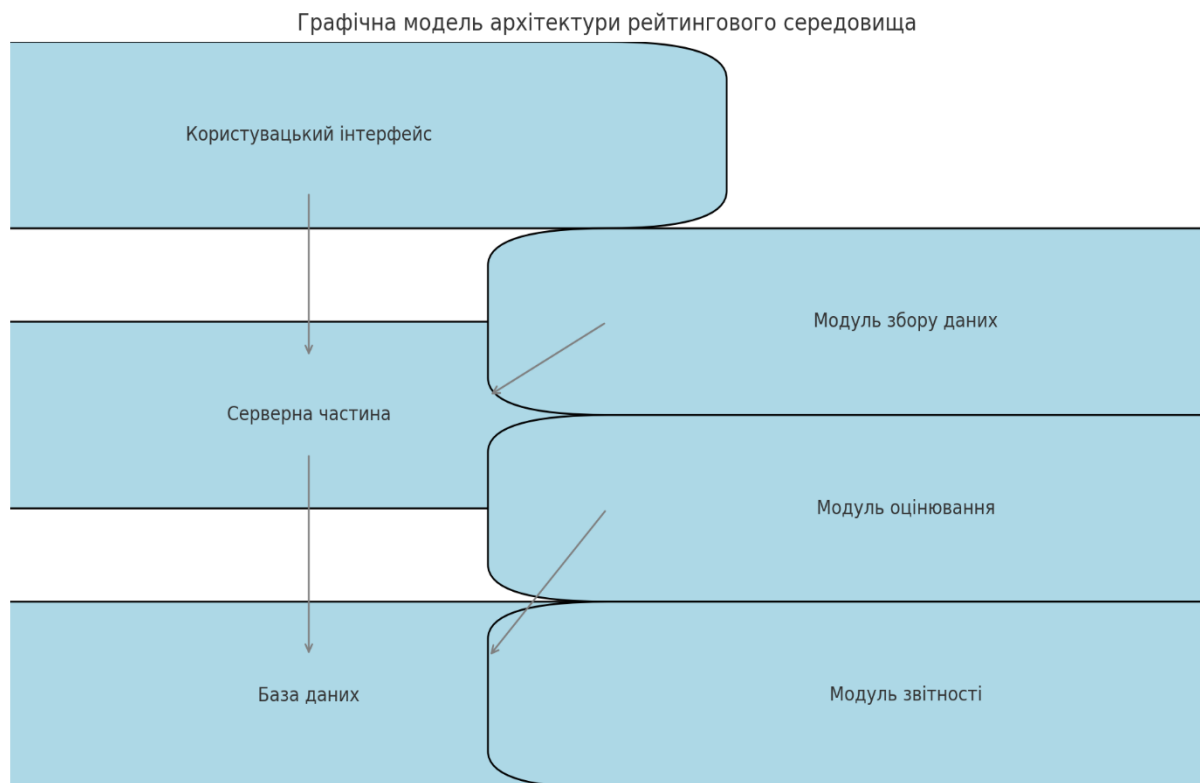
Рішення – це хмарні технології для зберігання та обробки інформації.

Проектування моделі рейтингового середовища базується на чіткій

структуризації інформаційних потоків, визначенні основних функцій системи та виборі відповідних технологій. Грамотно розроблена архітектура забезпечує гнучкість, масштабованість і безпеку системи, що робить її ефективним інструментом для оцінювання роботи НПП.

Електронне онлайн середовище «Рейтинг НПП» для Кам'янець-Подільського національного університету імені Івана Огієнка (КПНУ) має на меті забезпечити прозоре, автоматизоване та об'єктивне оцінювання роботи науково-педагогічних працівників. Система підтримує збір, обробку та аналіз даних, надаючи рейтинги за ключовими показниками, що враховують наукову, викладацьку, інноваційну та громадську діяльність НПП.

Запропонована модель електронного онлайн середовища «Рейтинг НПП» для КПНУ забезпечує автоматизацію процесу оцінювання, підвищує прозорість і об'єктивність, а також сприяє розвитку науково-педагогічної діяльності. Впровадження цієї системи дозволить університету ефективно управляти ресурсами та підвищувати конкурентоспроможність на національному і міжнародному рівнях.



Вище наведено графічне представлення основних компонентів

архітектури системи рейтингу. На графіку показані:

1. Користувацький інтерфейс для взаємодії з системою.
2. Серверна частина для обробки запитів і бізнес-логіки.
3. База даних для зберігання інформації.
4. Модулі збору даних, оцінювання та звітності, які взаємодіють між собою.

Таблиця з описом компонентів

Компонент	Опис
Користувацький інтерфейс (Frontend)	Інтерактивний веб-інтерфейс для роботи з системою.
Серверна частина (Backend)	Обробка запитів, реалізація бізнес-логіки, формування рейтингу.
База даних	Зберігання даних про НПП, оцінки та рейтинги.
Модуль збору даних	Імпорт даних із наукових баз та внутрішніх систем.
Модуль оцінювання	Обчислення рейтингу за заданими метриками та алгоритмами.
Модуль звітності	Генерація звітів, графіків та динаміки рейтингу.

Дана модель відображає взаємозв'язки між компонентами системи та їхніми функціями. Це дозволяє зрозуміти, як дані проходять через систему — від введення користувача до створення звітів і рейтингів.

2.3 Опис мобільного додатку для визначення рейтингу науково-педагогічних працівників

Назва застосунку **EduRank** — мобільний застосунок для автоматизації процесу збору, аналізу та відображення рейтингових показників науково-

педагогічних працівників (НПП).

Цільова аудиторія – адміністрація навчальних закладів; викладачі, які бажають оцінити власні результати; студенти, які можуть брати участь у створенні рейтингу через оцінювання викладачів.

Мета застосунку:

1. Надати зручний інструмент для автоматизації рейтингової системи викладачів.
2. Забезпечити прозорість і об'єктивність рейтингу.
3. Полегшити аналіз результатів викладацької діяльності за заданими критеріями.

Функціональні можливості:

1. Реєстрація та авторизація

Ролі користувачів:

- Адміністратор (створення критеріїв, управління системою).
- Викладач (додавання інформації про свою діяльність).
- Студент (оцінювання викладачів).

Методи входу:

- Електронна пошта + пароль.
- Використання корпоративного облікового запису (університетський email).

Критерії оцінювання

Викладачі отримують бали за:

- Наукову діяльність (публікації, конференції).
- Методичну діяльність (навчальні матеріали, курси).
- Студентське оцінювання (анонімні анкети студентів).
- Організаційну діяльність (участь у комісіях, організація заходів).

Кожен критерій має вагові коефіцієнти, які можна налаштовувати.

Особистий кабінет користувача

Для викладачів:

- Перегляд власних результатів.

- Завантаження доказових матеріалів (сертифікатів, публікацій тощо).

Для студентів:

- Анкетування викладачів із встановленими запитаннями.

Для адміністрації:

- Повний доступ до даних про викладачів.
- Формування фінального рейтингу.

Модуль анкетування

Система анонімних опитувань для студентів:

- Зручний інтерфейс для відповіді на запитання.
- Вбудована логіка автоматичного підрахунку балів.

Запитання можуть включати:

- Оцінку професійності викладача.
- Якість навчальних матеріалів.
- Пунктуальність, доступність для консультацій тощо.

Рейтинг викладачів

Автоматично розраховується за формулою:

$$\text{Рейтинг} = \sum (\text{Критерій}_i \times \text{Вага}_i)$$

Відображення даних:

- Загальний рейтинг викладачів.
- Рейтинг за категоріями (наукова, методична, організаційна діяльність).

Графічний інтерфейс: графіки, діаграми для візуалізації.

Адміністративні функції

Управління критеріями оцінювання – додавання, зміна, видалення.

Експорт даних: формування звітів у форматах PDF/Excel; надсилання результатів викладачам.

Додаткові можливості

Push-сповіщення:

- Нагадування про необхідність заповнити анкету.
- Інформація про оновлення рейтингу.

Мультимовність:

- Українська, англійська.

Технологічний стек:

Frontend

React Native для створення мобільного застосунку.

React Native – це фреймворк для розробки мобільних додатків за допомогою JavaScript і React, що дозволяє створювати кросплатформні застосунки (Android і iOS) з одним кодовим базисом.

Етапи використання нами React Native:

1. Налаштування середовища - встановлюємо Node.js: завантажуюмо та встановлюємо Node.js; встановлюємо Expo CLI:

```
bash  
  
npm install -g expo-cli
```

Експо допомагає швидко створювати й запускати React Native проєкти.

Створюємо новий проєкт:

```
bash  
  
expo init MyApp  
cd MyApp  
expo start
```

Після запуску ми отримуємо QR-код. Скануємо його за допомогою мобільного додатку **Експо Го** для тестування нашого застосунку на смартфоні.

2. Створення структури додатку

Головний файл – `App.js`:

Файл, з якого починається виконання застосунку. В ньому налаштовується основна логіка.

Додавання папок:

`/components` – для UI-компонентів.

`/screens` – для окремих екранів.

`/assets` – для зображень, шрифтів тощо.

Створення екранів відбувається наступним чином:

```
import React from 'react';
import { View, Text, StyleSheet } from 'react-native';

export default function HomeScreen() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Ласкаво просимо до EduRank!</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#fff',
  },
  text: {
    fontSize: 18,
    color: '#333',
  },
});
```

Додаємо навігацію між екранами та встановлюємо бібліотеку навігації:

```
npm install @react-navigation/native react-native-screens react-native-safe-area-c
```

Створюємо навігаційний стек:

```
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import HomeScreen from './screens/HomeScreen';
import DetailsScreen from './screens/DetailsScreen';

const Stack = createStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Details" component={DetailsScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

Додаємо бібліотеку форм:

```
npm install react-hook-form
```

Приклад форми оцінювання:

```
import React from 'react';
import { useForm, Controller } from 'react-hook-form';
import { View, Text, TextInput, Button, StyleSheet } from 'react-native';

export default function EvaluationForm() {
  const { control, handleSubmit } = useForm();
  const onSubmit = data => console.log(data);

  return (
    <View style={styles.container}>
      <Text>Оцініть викладача:</Text>
      <Controller
        name="rating"
        control={control}
        defaultValue=""
        render={({ field: { onChange, value } }) => (
          <TextInput
            style={styles.input}
            placeholder="Введіть оцінку"
            value={value}
            onChangeText={onChange}
          />
        )}
      />
      <Button title="Надіслати" onPress={handleSubmit(onSubmit)} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
  },
  input: {
    borderWidth: 1,
    borderColor: '#ccc',
    padding: 10,
    marginVertical: 10,
  },
});
```

Підключення до бази даних:

Використання Firebase. Встановлюємо бібліотеку Firebase:

```
npm install firebase
```

Ініціалізація Firebase у проєкті:

```
import { initializeApp } from 'firebase/app';

const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_AUTH_DOMAIN",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_STORAGE_BUCKET",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID",
};

const app = initializeApp(firebaseConfig);
```

Збереження оцінок у Firestore:

```
import { getFirestore, collection, addDoc } from 'firebase/firestore';

const db = getFirestore();

async function saveRating(rating) {
  try {
    const docRef = await addDoc(collection(db, 'ratings'), {
      rating: rating,
      date: new Date(),
    });
    console.log('Оцінка збережена з ID:', docRef.id);
  } catch (e) {
    console.error('Помилка при збереженні:', e);
  }
}
```

Деплой додатку.

Збірка додатку для Android:

Запускаємо команду:

```
expo build:android
```

Отримуємо APK-файл для тестування або публікації у Google Play.

Для iOS:

Використовуйте Mac і команду

```
expo build:ios
```

Backend – ініціалізація проєкту за допомогою **Node.js** (Express.js) для серверної логіки. Створюємо новий проєкт і встановлюємо Express.js

```
mkdir server-app
cd server-app
npm init -y
npm install express
```

Створюємо файл `server.js`

```
const express = require('express');
const app = express();
const PORT = 3000;

// Middleware для обробки JSON
app.use(express.json());

// Головний маршрут
app.get('/', (req, res) => {
  res.send('Ласкаво просимо до сервера!');
});

// POST-запит для збереження даних
app.post('/rating', (req, res) => {
  const { teacher, rating } = req.body;
  res.json({ message: `Оцінка ${rating} для ${teacher} збережена!` });
});

// Запуск сервера
app.listen(PORT, () => {
  console.log(`Сервер працює на http://localhost:${PORT}`);
});
```

База даних

Для створення бази даних для нашого мобільного застосунку ми використовуватимемо **PostgreSQL** — надійну реляційну базу даних. Вона чудово підходить для збереження інформації про викладачів та рейтинги.

Підключаємось до PostgreSQL та створюємо нову базу даних:

```
sql

CREATE DATABASE teacher_ratings;
```

Створюємо основні таблиці: **викладачі, оцінки, користувачі.**

```
sql

-- Таблиця викладачів
CREATE TABLE teachers (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    department VARCHAR(100),
    position VARCHAR(100)
);

-- Таблиця користувачів
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL
);

-- Таблиця оцінок
CREATE TABLE ratings (
    id SERIAL PRIMARY KEY,
    teacher_id INT REFERENCES teachers(id) ON DELETE CASCADE,
    user_id INT REFERENCES users(id) ON DELETE CASCADE,
    rating INT CHECK (rating >= 1 AND rating <= 5),
    comment TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Наповнення тестовими даними відбувається:

```
INSERT INTO teachers (name, department, position)
VALUES ('Слободянюк Олександр Васильович', 'Кафедра комп'ютерних наук', 'Доцент');

INSERT INTO users (username, password)
VALUES ('student1', 'hashed_password123');

INSERT INTO ratings (teacher_id, user_id, rating, comment)
VALUES (1, 1, 5, 'Чудовий викладач, добре пояснює матеріал!');
```

Створюємо запити до бази даних.

Отримати всі оцінки викладача:

```
SELECT t.name, r.rating, r.comment, r.created_at
FROM ratings r
JOIN teachers t ON r.teacher_id = t.id
WHERE t.id = 1;
```

Обчислити середній рейтинг:

```
SELECT t.name, AVG(r.rating) AS average_rating
FROM ratings r
JOIN teachers t ON r.teacher_id = t.id
GROUP BY t.name;
```

Підключення до серверної логіки:

З'єднаємо базу даних із сервером **Node.js (Express.js)** за допомогою **pg** (PostgreSQL клієнт):

```
const { Pool } = require('pg');

const pool = new Pool({
  user: 'username',
  host: 'localhost',
  database: 'teacher_ratings',
  password: 'password',
  port: 5432,
});

pool.query('SELECT * FROM teachers', (err, res) => {
  if (err) throw err;
  console.log(res.rows);
});
```

Ця структура бази даних зберігає інформацію про викладачів, студентів та їхні оцінки. Завдяки реляційним зв'язкам можна ефективно виконувати запити для відображення рейтингу викладачів та відгуків у нашому мобільному застосунку.

Далі наведемо приклад тестування POST-запиту для нашого додатку з використанням Postman. Для цього ми відкриваємо Postman та вибираємо метод POST. Вводимо на URL <http://localhost:3000/rating>. У розділі Body

вибираємо Raw та формат JSON.

Вводимо дані:

```
json

{
  "teacher": "Слободянюк Олександр Васильович",
  "rating": 5,
  "comment": "Чудовий викладач!"
}
```

Натискаємо Send і перевіряємо відповідь сервера.

Використання curl. У терміналі виконаємо наступну команду:

```
curl -X POST http://localhost:3000/rating \
-H "Content-Type: application/json" \
-d '{"teacher":"Слободянюк Олександр Васильович", "rating":5, "comment":"Чудовий викладач!"}
```

Очікувана відповідь:

```
json

{
  "message": "Оцінка 5 для Слободянюк Олександр Васильович збережена!"
}
```

Це підтвердить, що наш сервер коректно обробляє POST-запити.

Проведемо опис мобільного застосунку:

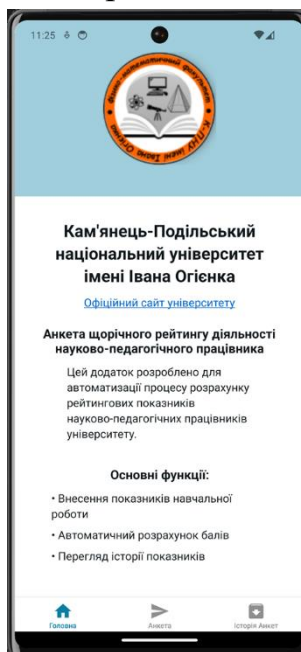


Рис 2.1 Титульна сторінка мобільного застосунку

Після проведення реєстрації проводимо анкетування НПП:

11:25

Анкета Рейтингу НПП

№	Зміст	Критерії	Б
1.1	Проведення навчальних занять зі спеціальних дисциплін іноземною мовою зі здобувачами вищої освіти освітніх ступенів «бакалавр», «магістр», «доктор філософії»	2 бали за 1 годину фактичного виконаного навчального навантаження	
1.2	Викладання за програмами академічної мобільності / Участь в освітніх проектах	10/20 балів за 1 годину фактичного викладання	
1.3	Результати опитування здобувачів вищої освіти через анкету «Викладач очима студентів»	10 балів × середнє арифметичне оцінок	

Загальна кількість балів: 0

ІсторіяЗберегти

Головна

Анкета

Історія Анкет

11:25

Анкета Рейтингу НПП

льних занять зі спеціальних дисциплін іноземною мовою зі здобувачами вищої освіти освітніх ступенів «бакалавр», «магістр», «доктор філософії»	2 бали за 1 годину фактичного виконаного навчального навантаження	2
дання за програмами академічної мобільності / Участь в освітніх проектах	10/20 балів за 1 годину фактичного викладання	3
тати опитування здобувачів вищої освіти через анкету «Викладач очима студентів»	10 балів × середнє арифметичне оцінок	0
едення вступних іспитів	1 бал за 1 годину (макс. 8 годин на день)	0

Загальна кількість балів: 5

ІсторіяЗберегти

Головна

Анкета

Історія Анкет

11:25

Історія анкет

10.12.2024

5 балів

Переглянути деталі →

10.12.2024

0 балів

Переглянути деталі →

Успіх

Анкету збережено успішно

OK

Головна

Анкета

Історія Анкет

11:26

Історія анкет

10.12.2024

5 балів

Переглянути деталі →

10.12.2024

0 балів

Переглянути деталі →

Головна

Анкета

Історія Анкет

Рис. 2.2-2.5 Анкета Рейтинг НПП

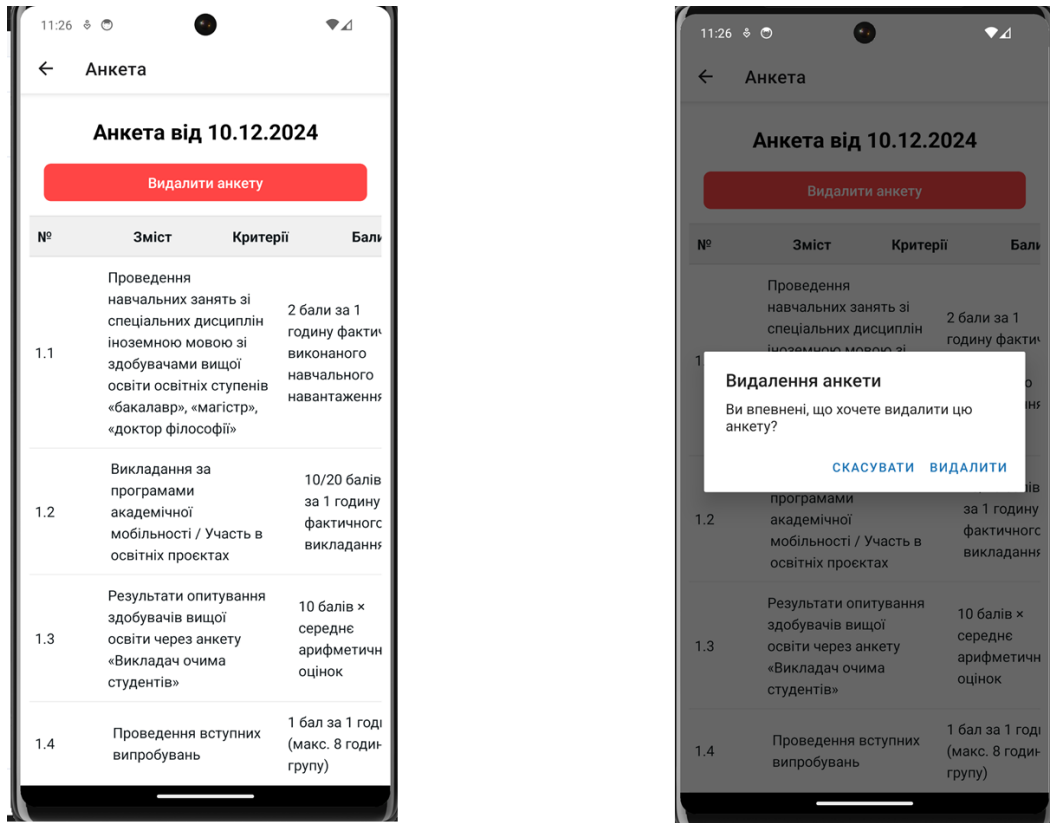


Рис. 2.6-2.7 Анкета Рейтинг НПП

Висновки до 2 розділу

У 2 розділі проведено огляд існуючих мобільних застосунків для створення електронних кабінетів науково-педагогічних працівників, Проведено проектування моделі рейтингового середовища та проведено опис мобільного додатку для визначення рейтингу науково-педагогічних працівників.

ВИСНОВКИ

По завершенню виконання магістерської роботи було проаналізовано літературу та вже існуюче програмне забезпечення для аналізу та формування рейтингу викладацького складу кафедри, з якого було зроблено здобуто наступну інформацію:

- з чого складається рейтинг кафедри;
- вплив рейтингу викладача на загальний рейтинг;
- оптимізовано інформацію, необхідну для створення автоматизованої системи аналізу викладацького складу.

Було розглянуто існуючі аналогічні системи для аналізу викладацького складу кафедри та сформульовано наступні завдання:

- проаналізовано можливості сучасних систем аналогів;
- проведено аналіз методів опрацювання рейтингів НПП;
- удосконалено використання методів розрахунку порівняно із системами попередниками;
- спроектовано архітектури нової системи;
- розроблено спроектовану систему.

Було проведено аналіз властивостей та закономірностей проблем при зборі інформації та формуванні кінцевих рейтингових результатів

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Введення в REST API — RESTful веб-сервіси від Хабр [Електронний ресурс]. режим доступу до ресурсу: <https://habr.com/ru/articles/483202/>
2. Документація по ASP.NET MVC від Metanit [Електронний ресурс] режим доступу до ресурсу: <https://metanit.com/sharp/aspnet5/3.1.php>
3. Фрімен А. Entity Framework Core 2 для ASP.NET Core MVC для професіоналів. Діалектика, 2019. 626 с
4. Документація по CanvasJS від CanvasJS [Електронний ресурс] режим доступу до ресурсу: <https://canvasjs.com/>.
5. Документація по Bootstrap від Bootstrap [Електронний ресурс] режим доступу до ресурсу: <https://bootstrap-4.ru/docs/5.0/gettingstarted/introduction/>
6. Офіційний веб-сайт Кам'янець-Подільського національного університету імені Івана Огієнка URL: <https://kpnu.edu.ua/>
7. Kai-Fu Lee. "AI Superpowers: Mobile Innovation" Houghton Mifflin Harcourt. 2018. 352 p.
8. John Smith. "Mobile App Development". Wiley. 2019. 450 p.
9. Elizabeth L. Gerber, Michael S. Casey. "Designing for Education" MIT Press. 2020. 200 p.
10. Martin Fowler. "Refactoring: Improving Mobile Applications".. Addison-Wesley. 2020. 450 p.
11. Michael Hartl. "Learning Mobile and Web Development". Pearson. 2021. 350 p.
12. Kyle Simpson. "You Don't Know JS: Mobile Edition". O'Reilly Media 2021. 320 p.
13. Andrea Resmini, Luca Rosati. "Information Architecture for the Web and Beyond" . Morgan Kaufmann. 2019. 450 p.
14. Max Kanat-Alexander. "Code Simplicity: Mobile Applications". No Starch Press. 2020. 250 p.
15. Daniel J. Barrett. "Linux Pocket Guide for Mobile App Testing". O'Reilly Media. 2021. 200 p.

16. Brent C. James. "Performance Evaluation for Mobile Applications". Springer. 2022. 310 p.

ДОДАТОК

Текст програмного модуля

openpyxl: for working with data from Ексель files InvalidFileException:
for handling input file names errors

numpy: for menu option arrays termcolor: for colored output in Windows
shell colorama: for colored text output os: for clearing console shell

tabulate: for output data from Ексель file as table

re: for subtract strings

"""

import openpyxl as Ексель

from openpyxl.utils.exceptions import InvalidFileException from numpy

import any, arange, array from termcolor import colored, cprint from colorama

import init from os import system from tabulate import tabulate

from re import sub

def inputNumber(msg):

"""

Displays msg and asks user to input a number.

Repeats until user inputs a valid number.

:param msg: input message

:return: valid integer number

"""

while True: try:

 number = int(input(msg))

 break except ValueError:

 print(colored("Invalid input!!", 'red')) pass

 return number

def displayMenu(options):

"""

Displays a menu of options, ask the user to choose an option and returns
the number of the menu item chosen. :param options: array of menu options

```

        :return: integer value of chosen option
        """

    # Display menu options    for i in range(len(options)):
        print(f"{i + 1}. {options[i]}")

    # Get a valid menu choice
    choice = 0    while not (any(choice == arange(len(options)) + 1)):
choice = inputNumber("Please choose a menu item: ")    if not (any(choice ==
arange(len(options)) + 1) == 1):        cprint("Invalid menu item!!", "red")
    return choice

    # 1st menu item - load data from Ексель file def loadFile():
        """

        Loads file and return loaded workbook object with entered name

        :return: string of file name and workbook object
        """

    while True:
        file_name = input("Input file name (q to exit): ")    if file_name.lower() ==
"q":
            file_name = "None"    return file_name, 0    try:        print("Try
to read...")        workbook = Ексель.load_workbook("../" + file_name + ".xlsx")
        cprint("Done!", "green")        break    except FileNotFoundError:
        cprint("File not found!!", 'red')    except InvalidFileException:
        cprint("Invalid file name!!", 'red')    return file_name, workbook

    # 2nd menu item - output data as table in console def outputData(wb):
        """

        Output data from Ексель file as table

        :param wb: current workbook
        :return: none
        """

```

```

if not wb:
    cprint("Load Ексель file first!!", 'red')    return

ws = wb.active
table = ws.iter_rows(values_only = True)    print(tabulate(table, headers =
'firstrow'))    return
# 3th menu item - correct data def correctData(wb):
    """

    Correct invalid spaces in each workbook cell

    param wb: current workbook
    :return: none
    """

    if not wb:
        cprint("Load Ексель file first!!", 'red')    return

    ws = wb.active
    teachers_id = set()

    print ("Start correcting data...")
    for row in ws.iter_rows(min_row = 2):
        changed = False    for cell in row:
            if type(cell.value) is not int:
                cell_content = str(cell.value)
                new_cell_content = sub('([:.,!?!?()])', r'\1', " ".join(cell_content.split()))
                if cell_content != new_cell_content:
                    changed = True

                    cell.value = new_cell_content
                    cell.value = str(cell.value).lower()

    if changed: teachers_id.add(row[2].value)
        cprint("Spaces was deleted!", 'green')

    print ("Finding duplicates of works...")
    duplicates = set()    for row in ws.iter_rows(min_col = 14, max_col = 19):
        if (row[0].value, row[5].value) in duplicates:            ws.delete_rows(row[0].row)
        teachers_id.add(row[5].value)    else:
            duplicates.add((row[0].value, row[5].value))            cprint("Duplicates was
deleted!", 'green')

```

```

        outputTeachers(wb, teachers_id)
        return
def outputTeachers(wb, teachers):
    """
        Output data about corrected teachers
    param wb: current workbook
        param teachers: list of corrected teachers
        return: none
    """
    if not teachers:
        cprint("No corrected teachers!!", 'red')        return
    ws = wb.active

    ids = list(teachers)
    teachers_info = [["Surname", "Name", "Patronymic", "Faculty"]]
    cprint("Corrected teachers:", 'cyan')

    for row in ws.iter_rows(min_row = 2):
        if row[2].value in ids:
            ids.remove(row[2].value)
            teachers_info.append([row[19].value, row[20].value, row[21].value,
row[24].value])

    print(tabulate(teachers_info, headers = 'firstrow'))    return
# 4th menu item - save data to a txt file
def writeData(wb):
    """
        Creates txt file or open existing and write    data from Ексель file
    param wb: current workbook
        return: none
    """
    if not wb:
        cprint("Load Ексель file first!!", 'red')        return
    ws = wb.active
    table = ws.iter_rows(values_only = True)

```

```

print ("Opening txt file...")
    file = open("table.txt", 'w', encoding = 'utf-8')    cprint("Done!", 'green')
print("Writing data to file...")
    file.write(tabulate(table, headers = 'firstrow'))    cprint("Done!", 'green')
file.close()
    return

# 5th menu item - save data to a new Ексель file
def saveFile(wb, name):
    """
        Saving workbook data to a new Ексель file
    param wb: current workbook
    param name: current workbook name
    return: none """
    if not wb:
        cprint("Load Ексель file first!!", 'red')    return
    print("Saving data to Ексель file...")    wb.save("new_" + name + ".xlsx")
    cprint("Done!", 'green')
    return

def main():
    init()
    current_file_name = "None"
    current_workbook = 0
    main_menu_items = array(["Load Ексель file", "Output data from file",
        "Correct data in file", "Write data to txt file",
        "Save file", "Exit"])
    while True:
        # Clear Windows shell
        system("cls")

        # Display current file name
        cprint("Current file: " + ("None" if current_file_name == "None" else
            f"{current_file_name}.xlsx"), "blue")

        # Display menu and get user choice    choice = display Menu
        (main_menu_items)    if choice == 1:
            current_file_name, current_workbook = loadFile()    elif choice == 2:

```

```

        outputData(current_workbook)        elif choice == 3:
        correctData(current_workbook)       elif choice == 4:
        writeData(current_workbook)         elif choice == 5:
        saveFile(current_workbook, current_file_name)    else:
        cprint("Bye, bye!", 'magenta')        break
        cprint("Enter to continue", 'yellow')
    input()

    return
if __name__ == "__main__":    main()"""
openpyxl: for working with data from Ексель files    InvalidFileException: for
handling input file names errors
numpy: for menu option arrays termcolor: for colored output in Windows shell
colorama: for colored text output
time: for count elapsed time while correcting data os: for clearing console shell
tabulate: for output data from Ексель file as table
re: for subtract strings
"""

import openpyxl as Ексель
from openpyxl.utils.exceptions import InvalidFileException from numpy
import any, arange, array from termcolor import colored, cprint from colorama
import init from os import system from time import perf_counter from tabulate
import tabulate

from re import sub
def inputNumber(msg):
    """
    Displays msg and asks user to input a number.
    Repeats until user inputs a valid number.

    param msg: input message
    return: valid integer number
    """

```

```

while True:    try:
    number = int(input(msg))
    break    except ValueError:
    print (colored("Invalid input!!", 'red'))    pass
    return number

def displayMenu(options):
    """
        Displays a menu of options, ask the user to choose    an option and returns
the number of the menu item chosen.
    param options: array of menu options
    return: integer value of chosen option
    """

    # Display menu options    for i in range(len(options)):
        print(f"{i + 1}. {options[i]}")

    # Get a valid menu choice
    choice = 0    while not (any(choice == arange(len(options)) + 1)):
choice = inputNumber("Please choose a menu item: ")    if not (any(choice ==
arange(len(options)) + 1) == 1):        cprint("Invalid menu item!!", "red")

    return choice
# 1st menu item - load data from Ексель file def loadFile():
    """

    Loads file and return loaded workbook object with entered name
    return: string of file name and workbook object
    """

    while True:

        file_name = input("Input file name (q to exit): ")    if file_name.lower() ==
"q":

        file_name = "None"    return file_name, 0    try:
        print ("Try to read...")    start = perf_counter()    workbook =
Ексель.load_workbook("../" + file_name + ".xlsx")    cprint("Done!", 'green')
        cprint(f"Time: {perf_counter() - start:.4f}", 'magenta')    break

```

```

        except FileNotFoundError:          cprint("File not found!!", 'red')      except
InvalidFileException:          cprint("Invalid file name!!", 'red')
        return file_name, workbook

# 2nd menu item - output data as table in console def outputData(wb):
    """

    Output data from Ексель file as table

    param wb: current workbook
    return: none

    """

    if not wb:
        cprint("Load Ексель file first!!", 'red')

        return

    ws = wb.active
    table = ws.iter_rows(values_only = True)    print(tabulate(table, headers =
'firstrow'))
    return

# 3th menu item - correct data def correctData(wb):
    """

    Correct invalid spaces in each workbook cell

    param wb: current workbook
    return: none

    """

    if not wb:
        cprint("Load Ексель file first!!", 'red')

        return

    ws = wb.active
    teachers_id = dict()

    # SPACES print("Start correcting data...")    start = perf_counter()    for row in
ws.iter_rows(min_row = 2):
        changed = False    for cell in row:        if type(cell.value) is not int:
cell_content = str(cell.value)        new_cell_content = sub('([:.,!?!?()])', r'\1', "
".join(cell_content.split()))        if cell_content != new_cell_content:
changed = True

```

```

    cell.value = new_cell_content

    cell.value = str(cell.value).lower()
    if changed and row[2].value not in teachers_id.keys():
        teachers_id[row[2].value] = {"Spaces"}    cprint("Spaces was deleted!", 'green')
        cprint(f"Time: {perf_counter() - start:.4f} sec", 'magenta')

    # DUPLICATES
    Print ("Finding duplicates of works...")    start = perf_counter()    duplicates
= set()    for row in ws.iter_rows(min_col = 14, max_col = 19):        if
(row[5].value, row[0].value) in duplicates:            if row[5].value not in
teachers_id.keys():                teachers_id[row[5].value] = {"Duplicate works"}
else:
    teachers_id[row[5].value].add("duplicate works")
ws.delete_rows(row[0].row)    else:
    duplicates.add((row[5].value, row[0].value))    del duplicates
    cprint("Duplicates was deleted!", 'green')
    cprint(f"Time: {perf_counter() - start:.4f} sec", 'magenta')

    # PERCENTS
    print("Working with percents...")    start = perf_counter()    for row in
ws.iter_rows(min_row = 2):
        if isinstance(row[9].value, str):            # Percent value is NULL
continue        elif row[9].value > 100:            # Percent value is > than 100
if row[5].value not in teachers_id.keys():                teachers_id[row[5].value] =
{"One-time % more than 100"}            else:
    teachers_id[row[5].value].add("one-time % more than 100")
ws.delete_rows(row[0].row)    else:
    # Percents by work > than 100            sum_percent = row[9].value
current_work = row[13].value            current_id = row[2].value
    for find in ws.iter_rows(min_col = 10, max_col = 19, min_row =
row[0].row):        if isinstance(find[0].value, str):
continue            elif find[4].value == current_work and find[9].value !=
current_id:                sum_percent += find[0].value            if sum_percent >

```

100:

```

    for delete in ws.iter_rows(min_col = 14, max_col = 19, min_row =
row[0].row):
        if delete[0].value == current_work:
            if
delete[5].value not in teachers_id.keys():
                teachers_id[delete[5].value]
= {"Total % more than 100"}
            else:
                teachers_id[delete[5].value].add("total % more than 100")
ws.delete_rows(delete[0].row)    cprint("Percentages have been corrected!",
'green')    cprint(f"Time: {perf_counter() - start:.4f} sec", 'magenta')
outputTeachers(wb, teachers_id)    return
def outputTeachers(wb, teachers):
    """

    Output data about corrected teachers
param wb: current workbook
    param teachers: list of corrected teachers
    return: none
    """

    if not teachers:
        print("No corrected teachers!!", 'red')    return

ws = wb.active
teachers_info = [{"Surname", "Name", "Patronymic", "Faculty", "Problem
with"}]
cprint("Corrected teachers:", 'cyan')
for row in ws.iter_rows(min_row = 2):
    if row[2].value in teachers:
        teachers_info.append([row[19].value,
                                row[20].value,
                                row[21].value, row[24].value, " - ".join(teachers[row[2].value])])
del
teachers[row[2].value]

# Print to console
    Print (tabulate(teachers_info, headers = 'firstrow'))

# Open txt    print("Opening txt file...")    start = perf_counter()
    file = open("teachers.txt", 'w', encoding = 'utf-8')    cprint("Done!", 'green')

    cprint(f"Time: {perf_counter() - start:.4f}", 'magenta')

# Write to txt

```

```

    Print ("Writing teachers data to file...")    start = perf_counter()
        file.write(tabulate(teachers_info, headers = 'firstrow'))    cprint("Done!",
'green')
        cprint(f"Time: {perf_counter() - start:.4}", 'magenta')    return

# 4th menu item - save data to a txt file def writeData(wb):
    """
        Creates txt file or open existing and write    data from Ексель file
    param wb: current workbook
    :return: none
    """

    if not wb:
        cprint("Load Ексель file first!!", 'red')    return
    ws = wb.active
        table = ws.iter_rows(values_only = True)
    print("Opening txt file...")    start = perf_counter()
        file = open("data.txt", 'w', encoding = 'utf-8')    cprint("Done!", 'green')
        cprint(f"Time: {perf_counter() - start:.4}", 'magenta')
    print("Writing data to file...")    start = perf_counter()
        file.write(tabulate(table, headers = 'firstrow'))    cprint("Done!", 'green')
        cprint(f"Time: {perf_counter() - start:.4}", 'magenta')
    file.close()
    return

# 5th menu item - save data to a new Ексель file def saveFile(wb, name):
    """
        Saving workbook data to a new Ексель file
    param wb: current workbook
    param name: current workbook name
    return: none
    """

    if not wb:
        cprint("Load Ексель file first!!", 'red')

```

```

        return

    print("Saving data to Ексель file...")    start = perf_counter()    wb.save("new_"
+ name + ".xlsx")    cprint("Done!", 'green')
        cprint(f"Time: {perf_counter() - start:.4}", 'magenta')    return

def main():    init()
    current_file_name = "None"
    current_workbook = 0

    main_menu_items = array(["Load Ексель file", "Output data from file",
        "Correct data in file", "Write data to txt file",
        "Save file", "Exit"])

    while True:
        # Clear Windows shell
        system("cls")

        # Display current file name
        cprint("Current file: " + ("None" if current_file_name == "None" else
f"{current_file_name}.xlsx"), "blue")

        # Display menu and get user choice        choice =
displayMenu(main_menu_items)        if choice == 1:
            current_file_name, current_workbook = loadFile()        elif choice == 2:
            outputData(current_workbook)        elif choice == 3:
            correctData(current_workbook)        elif choice == 4:
            writeData(current_workbook)        elif choice == 5:
            saveFile(current_workbook, current_file_name)        else:
            cprint("Bye, bye!", 'magenta')        break
            cprint("Enter to continue", 'yellow')        input()

    return

if __name__ == "__main__":    main()

```