

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТА
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра комп'ютерних наук

ПРОЄКТ ДРУГОГО РІВНЯ

НАЗВА:

«ЕЛЕКТРОННА ВІДОМІСТЬ СТУДЕНТА УНІВЕРСИТЕТУ «КРОК»

Ступінь вищої освіти – бакалавр
Спеціальність – 122 «Комп'ютерні науки»
Освітня програма «Комп'ютерні науки»

ЗВІТ З ПРОЄКТУ

Команда проєкту
Владислав АРХИПЕНКОВ
Віктор БАЦУРА

Керівник проєкту
Олег МУШИНСЬКИЙ

Київ – 2025 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

ЗАТВЕРДЖУЮ:
 завідувач кафедри комп'ютерних наук
 _____ **Сергій МІЧКІВСЬКИЙ**
 «28» квітня 2025 р.

ЗАВДАННЯ НА ПРОЄКТ ДРУГОГО РІВНЯ

Назва проекту	SharePoint Client-Server Web-Part: «Електронна відомість студента»
Мета проекту	Створити застосунок який надає можливість студентам навчального закладу швидко отримувати інформацію про виставлені академічні оцінки та наявні борги по дисциплінам
Опис проекту	Забезпечує швидкий та зручний доступ студентів до інформації про їхні оцінки та заборгованості без необхідності звертатися до викладачів або адміністрації. Зменшує навантаження на співробітників навчального закладу, автоматизуючи процес інформування студентів.
Очікувані результати	Створити застосунок який надає можливість студентам навчального закладу швидко отримувати інформацію про виставлені академічні оцінки та наявні борги по дисциплінам
Ролі та відповідальність виконавців	Віктор Бацура – Frontend-розробник; Дизайнер Владислав Архипенков – Backend-розробник
Критерії успішності проекту	Інтерфейс зрозумілий, адаптивний та зручний для студентів Web-Part стабільно працює в середовищі SharePoint, підтримує авторизацію та відображається відповідно до прав доступу користувача. Доступ до інформації реалізовано відповідно до ролей користувачів, немає витоків або несанкціонованого доступу до відомостей. Проект протестований, не має критичних помилок, та виконує всі запити без збоїв.
<i>Технологічний стек, що використовується, та його обґрунтування</i>	
Технологія/Інструмент	SharePoint, SPFx, React, TS, ASP.NET Core, C#
Опис	SPFx, React, TS – використовуються для створення Web-Part для SharePoint, SPFx це основний фреймворк для їх побудови, React надає можливість створювати динамічні користувацькі інтерфейси. ASP.NET Core, C# - веб-фреймворк для побудови серверної логіки (API), буде відповідати за авторизацію та пошук даних.

Причини вибору	Ми обрали цей стек, оскільки він ідеально підходить для інтеграції з SharePoint та екосистемою Microsoft 365. SPFx дозволяє створювати сучасні клієнтські Web-Part'и, а React і TypeScript забезпечують зручний, динамічний та надійний інтерфейс. Для реалізації серверної логіки ми використовуємо ASP.NET Core і C#, які дають гнучкість, безпеку та хорошу продуктивність.
----------------	--

Дата видачі завдання 28 квітня 2025р.

Команда

Владислав АРХИПЕНКОВ,

Віктор БАЦУРА

Керівник проєкту

Олег МУШИНСЬКИЙ

АННОТАЦІЯ

Під час проєкту другого рівня розроблено застосунок «Електронна відомість студента» для відображення академічної інформації про студентів Університету економіки та права «КРОК». Застосунок забезпечує доступ до поточних оцінок, екзаменів, заліків, а також до архіву завершених результатів і інформації про академічні борги. Фронт-енд частина реалізована з використанням SPFx, а бек-енд - на базі C# та ASP.NET Core. На відміну від попереднього рішення на PowerApps, новий застосунок дозволяє досягти кращої масштабованості, швидкодії та зручності користування. На цьому етапі застосунок призначений виключно для студентів і не передбачає можливості коригування оцінок з боку викладачів.

ЗМІСТ

АННОТАЦІЯ	4
ВСТУП	6
1. КОНЦЕПТУАЛІЗАЦІЯ ПРОДУКТУ	8
1.1. Опис ідеї продукту	8
1.2. Аналіз предметної області та конкурентів	9
1.2.1. Порівняльна таблиця конкурентів.....	9
1.2.2. SWOT-аналіз розроблюваного продукту	9
1.2.3. Бізнес-модель Lean Canvas продукту	10
1.3. Опис продукту	11
1.3.1. Мета створення продукту	11
1.3.2. Вимоги до продукту.....	11
1.3.3. Вимоги до функцій.....	12
1.3.4. Системні вимоги	12
2. ПРОЄКТУВАННЯ ПРОДУКТУ	14
2.1. Моделювання поведінки продукту	14
2.2. Моделювання структури продукту	15
2.3. Дизайн продукту	16
3. РЕАЛІЗАЦІЯ ПРОДУКТУ	19
3.1. Особливості реалізації клієнтської частини	19
3.1.1. Дизайн інтерфейсу	19
3.1.2. Авторизація користувача	21
3.1.3. Взаємодія з серверною частиною	21
3.2. Особливості реалізації серверної частини.....	22
3.2.1. Обробка запитів на Web API	23
3.2.2. Авторизація студента.....	24
3.2.3. Отримання оцінок	25
3.2.4. Повернення результатів	25
3.3. Тестування	25
3.3.1. Тестування клієнтської частини	25
3.3.2. Тестування серверної частини.....	26
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
ДОДАТКИ	31

ВСТУП

Актуальність теми. У сучасних умовах цифровізації університету «КРОК» (завдяки Центру Автоматизації та Розробки) освіти особливої важливості набуває створення ефективних електронних інструментів для взаємодії студентів з академічною інформацією. Багато закладів вищої освіти все ще використовують застарілі або обмежені за функціональністю рішення, які ускладнюють доступ до навчальних даних. Хоча в складі кабінета студента вже є наявне рішення цієї проблеми, але з розробницької та користувацької точки зору воно не є ідеальним. Актуальність теми зумовлена потребою в сучасному, гнучкому та масштабованому застосунку, що забезпечить студентам зручний доступ до результатів навчання, сприятиме оперативному виявленню академічних боргів та кращій організації освітнього процесу.

Мета проєкту. Метою даного проєкту є створення сучасного електронного застосунку для відображення студентських оцінок, який забезпечує зручний інтерфейс, швидкий доступ до навчальної інформації та можливість подальшого розширення функціоналу відповідно до потреб навчального закладу.

Об'єктом дослідження є інформаційна система управління академічними даними студентів у вищому навчальному закладі.

Предметом дослідження є процес розробки та впровадження клієнт-серверного застосунку для візуалізації результатів навчання студентів з використанням технологій SPFx та ASP.NET Core.

Завдання проєкту:

1. Проаналізувати попередню реалізацію системи на базі PowerApps.
2. Визначити функціональні та нефункціональні вимоги до нового застосунку.
3. Реалізувати Front-end частину застосунку за допомогою SPFx.
4. Розробити серверну частину API з використанням C# та ASP.NET Core.

5. Забезпечити відображення поточних, архівних оцінок та академічних боргів.

6. Перевірити ефективність та стабільність роботи застосунку.

Практична цінність проєкту полягає у створенні застосунку, який може бути впроваджений у навчальному процесі для покращення взаємодії студентів з академічною інформацією. Застосунок може стати основою для подальшого розширення, зокрема додавання можливостей для викладачів та інтеграції з іншими навчальними сервісами.

1. КОНЦЕПТУАЛІЗАЦІЯ ПРОДУКТУ

1.1. Опис ідеї продукту

Ідея створення вебсервісу «Електронна відомість студента» виникла через проблеми, які є в наявному рішенні на основі Power Apps. Наразі студенти університету переглядають свої оцінки та академічні борги через спеціальний застосунок Power Apps на «Кабінеті студента». Але цей застосунок працює повільно, особливо коли потрібно обробляти великі обсяги інформації — у списках оцінок університету зберігаються сотні тисяч записів. Крім того, існують ризики безпеки, оскільки студенти змушені мати прямий доступ до списків оцінок. Саме тому було прийнято рішення створити веб-парту для платформи SharePoint, яка б вирішувала ці проблеми. Ідея полягала в перенесенні функціоналу відображення оцінок у спеціалізований модуль із власним API. Це дозволяє:

- значно прискорити завантаження інформації;
- осилити захист особистих даних студентів;
- створити зручний, зрозумілий і адаптивний до різних пристроїв інтерфейс.

Перевірка ідеї відбулася через аналіз наявних проблем та недоліків існуючого рішення на основі Power Apps. На практиці було помічено, що студенти часто стикаються з повільною роботою застосунку, особливо при великих обсягах інформації. Крім того, була очевидною проблема безпеки особистих даних студентів, оскільки їм доводилося мати прямий доступ до списків із оцінками. Таким чином, стало зрозуміло, що необхідно створити більш швидке, зручне і безпечне рішення, адаптоване під сучасні стандарти веб-інтерфейсів та захисту даних.

1.2. Аналіз предметної області та конкурентів

1.2.1. Порівняльна таблиця конкурентів

Зараз на ринку існує кілька типів продуктів, які вирішують схожі завдання щодо управління оцінками та особистими даними студентів. Нижче наведена порівняльна таблиця конкурентів, адаптована до умов університету. (див. табл. 1.1)

Таблиця 1.1 - Порівняльна таблиця конкурентів

<i>Конкуренти</i>	<i>Ключові функції</i>	<i>Переваги продукту</i>	<i>Недоліки продукту</i>
Power Apps (від Microsoft)	Перегляд оцінок; інтеграція з SharePoint; налаштовувані форми та списки	Легка інтеграція з Office365; проста налаштовуваність	Повільна робота з великими обсягами даних; ризики безпеки (прямий доступ користувачів до списків)
Google Classroom	Система оцінок; інтеграція з Google Suite; мобільні додатки	Простий інтерфейс; хороша адаптація під мобільні пристрої	Потрібна окрема інфраструктура (поза SharePoint); немає глибокої інтеграції з SharePoint
Moodle	Платформа для навчання; управління оцінками; інтеграція з іншими сервісами	Багатий функціонал для навчання; активна спільнота користувачів	Складна система налаштування; потребує додаткових серверів і адміністрування

1.2.2. SWOT-аналіз розроблюваного продукту

Для оцінки можливостей розроблюваного продукту було виконано SWOT-аналіз, який дозволяє визначити сильні та слабкі сторони системи, а також потенційні можливості розвитку і можливі загрози під час її впровадження. SWOT-аналіз наведений у табл. 1.2.

Таблиця 1.2 – SWOT-аналіз

<i>Сильні сторони</i>	<i>Слабкі сторони</i>
1. Швидка робота навіть з великими обсягами даних. 2. Високий рівень безпеки даних. 3. Простий, зрозумілий інтерфейс, що адаптується до різних пристроїв.	1. Початково обмежений функціонал (тільки перегляд оцінок). 2. Залежність від роботи зовнішнього API-сервера. 3. Відсутність власної адміністративної панелі (управління через SharePoint).
<i>Можливості</i>	<i>Загрози</i>
1. Подальше розширення функціоналу: додавання нових можливостей, модулів і функцій. 2. Популярність SharePoint у навчальних закладах, що може сприяти поширенню продукту.	1. Можливі зміни політики безпеки чи API у Microsoft, що можуть вплинути на роботу. 2. Поява більш комплексних рішень конкурентів, наприклад, нових платформ для навчання.

1.2.3. Бізнес-модель *Lean Canvas* продукту

Для визначення основних складових функціонування продукту було використано модель *Lean Canvas*, яка дозволяє коротко описати цільових користувачів, їхні проблеми, запропоноване рішення та ключові переваги системи. Бізнес-модель розроблюваного продукту наведена у табл. 1.3.

Таблиця 1.3 – *Lean-Canvas* продукту

<i>Сегменти споживачів</i>	<i>Проблеми</i>	<i>Унікальна пропозиція</i>
Студенти університету	Повільна робота наявного рішення, ризику безпеки	Швидкий та безпечний доступ до персональних оцінок
Викладачі та адміністрація	Складність в управлінні доступом до оцінок студентів	Просте управління доступом через централізований API

<i>Канали взаємодії</i>	<i>Рішення</i>	<i>Джерела доходу</i>
Веб-парт на SharePoint-сторінці	Розробка власного SharePoint веб-парту з API інтеграцією	Продукт є внутрішнім і не передбачає прямого доходу (розробляється для університету)
<i>Витрати</i>	<i>Ключові метрики</i>	<i>Конкурентні переваги</i>
Витрати на підтримку сервера, Azure App, адміністрування API	Час завантаження оцінок, безпека даних, зворотній зв'язок користувачів	Безпека даних, швидкодія, інтеграція з існуючими рішеннями

1.3. Опис продукту

1.3.1. Мета створення продукту

Головною метою створення веб-парту «Електронна відомість студента» є надання студентам швидкого, безпечного та зручного способу перегляду власних оцінок, академічних заборгованостей та іншої важливої персоналізованої інформації на платформі SharePoint. Важливим аспектом є покращення продуктивності та безпеки в порівнянні з поточним рішенням на базі Power Apps, а також забезпечення сучасного і адаптивного користувацького інтерфейсу.

1.3.2. Вимоги до продукту

Продукт повинен відповідати таким вимогам:

- продуктивність. Швидке завантаження даних навіть при роботі зі списками, які містять понад 100 тисяч записів;
- безпека. Відсутність прямого доступу студентів до списків оцінок; використання проміжного API, що авторизується через Azure AD;
- адаптивність. Інтерфейс має бути зручним і коректно відображатися на всіх пристроях (ПК, смартфони, планшети);

- інтегрованість. Веб-парта повинна легко інтегруватися в екосистему SharePoint університету без додаткових складних налаштувань з боку користувача.

1.3.3. Вимоги до функцій

Функціональні вимоги до веб-парти включають:

- персоналізоване відображення інформації. Веб-парта повинна автоматично розпізнавати поточного користувача (студента) та виводити інформацію саме для нього;
- виведення списку оцінок. Інтерфейс повинен показувати поточні оцінки, академічні заборгованості та вже здані предмети;
- оновлення інформації в реальному часі. Підтримка актуальності даних завдяки регулярним запитам до API;
- інтерактивність інтерфейсу. Наявність зручних елементів управління, таких як фільтри, перемикання між типами оцінок, сортування та пошук за назвою предметів;
- можливість подальшого масштабування. Архітектура веб-парти має дозволяти легко додавати нові функції, модулі або додаткові сервіси.

1.3.4. Системні вимоги

Технічні вимоги до середовища для роботи вебсервісу:

- фреймворк SharePoint Framework (SPFx), React з використанням Fluent UI для компонентів інтерфейсу;
- API-сервер: .NET (C#) Web API, розгорнутий в Azure App Service, для отримання даних зі списків SharePoint. Авторизація: AadHttpClient для захищених запитів до серверного API через Azure Active Directory;
- сумісність. Веб-парта повинна стабільно працювати на всіх популярних сучасних браузерях: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari;

- інфраструктурні вимоги: Для розгортання веб-парти необхідний доступ до адміністративної панелі SharePoint для завантаження і налаштування пакету (.sppkg-файл).

Таким чином, продукт «Електронна відомість студента» розробляється з урахуванням усіх сучасних стандартів і технологій, що дозволяє досягнути високої ефективності, зручності та безпеки користування.

2. ПРОЄКТУВАННЯ ПРОДУКТУ

2.1. Моделювання поведінки продукту

Для розуміння роботи веб-парті «Електронна відомість студента» використано діаграму варіантів використання (див. рис. 2.1). Цей тип діаграми дозволяє чітко продемонструвати взаємодію користувачів (акторів) з основними функціями системи.

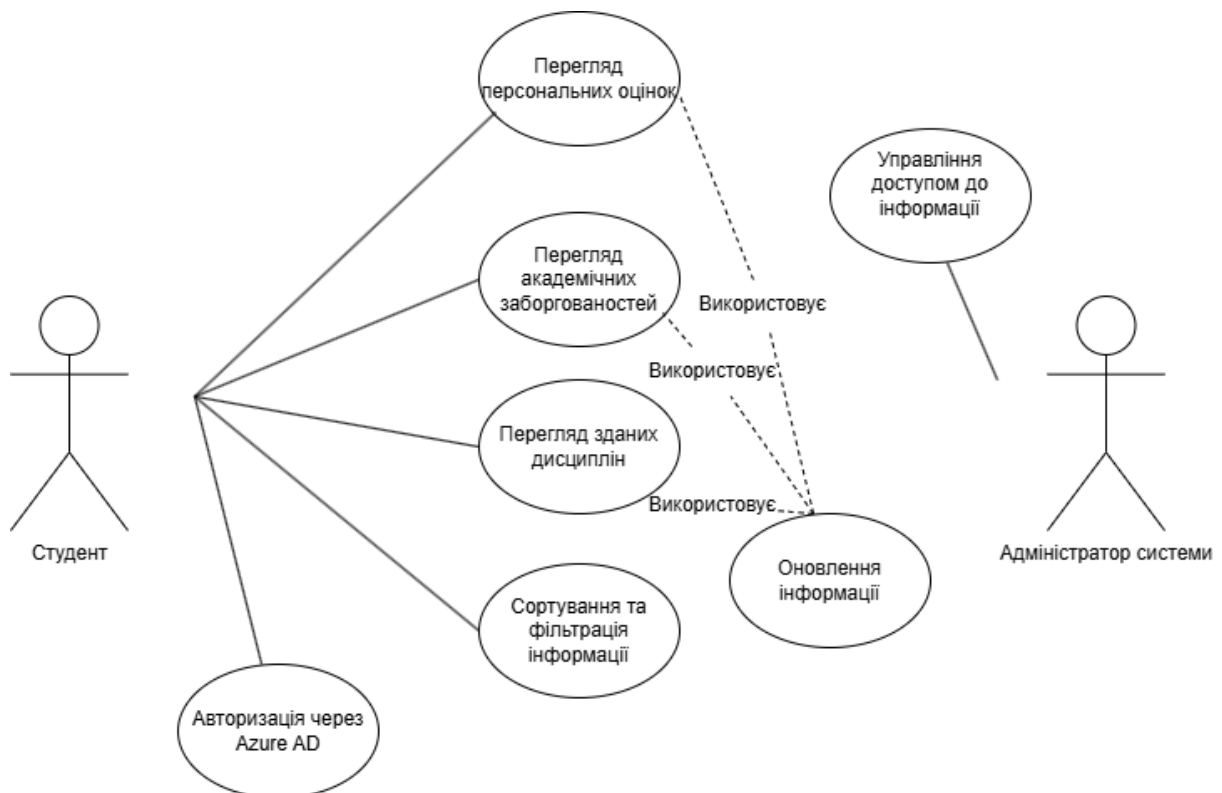


Рисунок 2.1 - Діаграма варіантів використання

Джерело: розроблено автором

Опис діаграми:

- перегляд персональних оцінок: студент має можливість бачити поточні оцінки за всіма предметами;
- перегляд академічних заборгованостей: студент бачить список предметів, за якими він має академічні борги;

- перегляд зданих дисциплін: студент бачить оцінки за завершені дисципліни;
- сортування та фільтрація інформації: студент може сортувати і фільтрувати дані за дисциплінами, семестрами або типом оцінки;
- авторизація через Azure AD: веб-парта автоматично отримує дані про студента через авторизацію у SharePoint;
- оновлення інформації: система регулярно оновлює дані автоматично через API;
- управління доступом до інформації (адміністратор): налаштування дозволів, доступу до списків оцінок та налаштування API (ця дія для адміністратора системи).

2.2. Моделювання структури продукту

Веб-парта «Електронна відомість студента» отримує інформацію виключно з існуючих списків SharePoint. Використовуються чотири основні списки:

- Оцінки Поточні – Університет;
- Оцінки Поточні – Коледж;
- Оцінки Архів – Університет;
- Оцінки Архів – Коледж.

Структура даних цих списків є ідентичною. (див. табл. 2.1)

Таблиця 2.1 – Структура моделі оцінки

<i>Назва поля</i>	<i>Тип поля</i>	<i>Опис</i>
Ідентифікатор	Число	Унікальний номер запису
Назва дисципліни	Текст	Назва дисципліни, яка виводиться студенту
Назва дисципліни (англ.)	Текст або Null	Назва дисципліни англійською мовою
Email викладача	Текст або Null	Адреса електронної пошти викладача
ПІБ викладача	Текст або Null	Повне ім'я викладача
Семестр	Текст	Семестр, в якому дисципліна вивчається

<i>Назва поля</i>	<i>Тип поля</i>	<i>Опис</i>
Дата закінчення сесії	Дата/час (ISO)	Дата завершення сесії у форматі ISO
Дата завершення безкоштовного перекладання	Дата/час (ISO)	Останній день безкоштовного перекладання
Дата виставлення оцінки	Дата/час (ISO або Null)	Дата, коли оцінка була виставлена студенту
Кількість кредитів	Число	Кількість кредитів ECTS за дисципліну
Оцінка (0-100)	Число	Оцінка за шкалою від 0 до 100
Оцінка за 12-бальною шкалою	Число	Оцінка за 12-бальною шкалою
Тип оцінювання	Текст	Тип оцінювання (наприклад, «Залік», «Іспит»)
Перездача платна	Логічний (Так/Ні)	Показник, чи перездача дисципліни платна
До кінця безкоштовної перездачі ≤ 5 днів	Логічний (Так/Ні)	Позначає, чи залишилось 5 або менше днів до кінця терміну безкоштовної перездачі
Перезарахування	Логічний (Так/Ні)	Показник перезарахування дисципліни
Складено протягом сесії	Логічний (Так/Ні)	Чи була дисципліна здана в рамках сесії

Дані списків SharePoint отримуються через розроблений API, який працює через авторизацію Azure AD. Це гарантує безпеку і стабільність отримання інформації та дозволяє уникнути прямого доступу студентів до вихідних даних.

Таким чином, забезпечується чітка структура і прозора модель роботи веб-парти з оцінками, яка легко підтримується та масштабується за необхідності.

2.3. Дизайн продукту

Під час створення дизайну веб-парти «Електронна відомість студента» важливим орієнтиром було поточне рішення на базі Power Apps. Основна мета полягала в тому, щоб не втратити функціонал, який вже використовувався студентами, та забезпечити відображення всієї необхідної інформації у зрозумілому та звичному вигляді.

На першому етапі було зроблено аналіз існуючого застосунку (див. рис. 2.2). Він показав, що інтерфейс потребує удосконалення в таких моментах:

- покращення зручності використання;
- чітке виділення важливої інформації;
- адаптація під сучасні мобільні пристрої та монітори з різною роздільною здатністю.

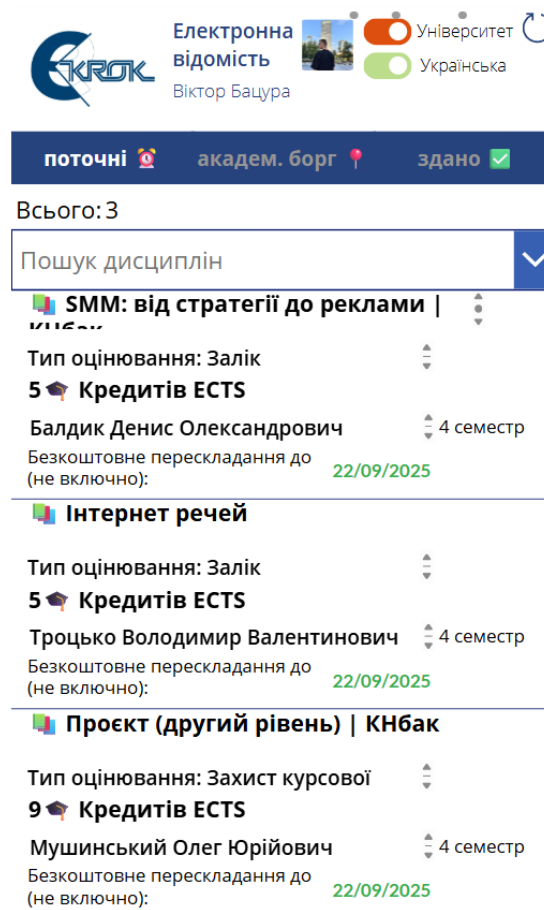


Рисунок 2.2 – PowerApps застосунок

Джерело: Кабінет студента

На основі цього аналізу було створено макет у Figma, який враховує попередній досвід використання, але водночас є більш сучасним, інтуїтивно зрозумілим і зручним. Головними аспектами нового дизайну є:

- простий та лаконічний вигляд оцінок і дисциплін;
- інтуїтивна навігація, зручні елементи управління для сортування та фільтрації інформації;

- покращена адаптивність і підтримка різних типів пристроїв.

Фінальний макет дизайну представлений на наступному зображенні (див. рис. 2.3).

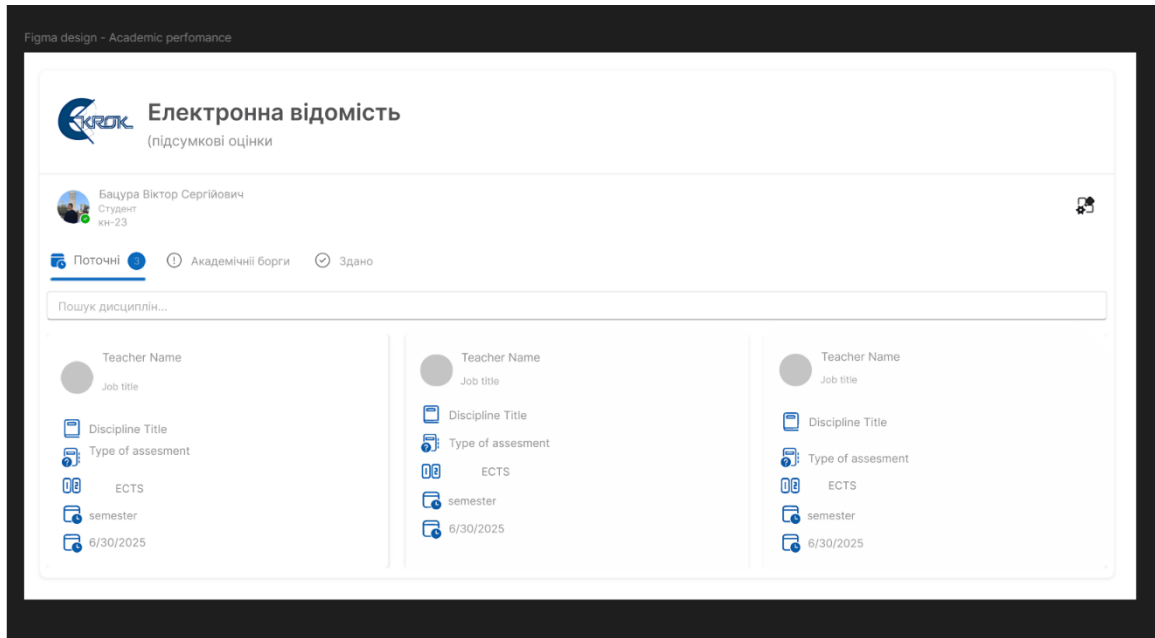


Рисунок 2.3 – Новий дизайн застосунку

Джерело: розроблено авторами

Цей дизайн дозволяє зробити взаємодію студентів з інформацією швидкою, зрозумілою і комфортною.

3. РЕАЛІЗАЦІЯ ПРОДУКТУ

3.1. Особливості реалізації клієнтської частини

Клієнтська частина застосунку «Електронна відомість студента» реалізована з використанням SharePoint Framework (SPFx) у поєднанні з React, TypeScript та бібліотекою інтерфейсних компонентів Fluent UI. Такий технологічний стек є сучасним і рекомендованим підходом для розробки інтерактивних, продуктивних та естетично узгоджених застосунків у середовищі Microsoft 365.

SPFx (SharePoint Framework) - це офіційна платформа від Microsoft для створення клієнтських вебчастин (web parts), розширень та застосунків для SharePoint Online. Головними перевагами SPFx є:

- повна інтеграція з Microsoft 365 (зокрема, з SharePoint Online);
- підтримка сучасних вебтехнологій (React, TypeScript, Webpack);
- робота у контексті поточного користувача без потреби в серверному рендерінгу;
- можливість доступу до Microsoft Graph API, SharePoint REST API та інших сервісів.

В нашому проєкті SPFx забезпечує розгортання вебчастин безпосередньо у SharePoint-середовище, з підтримкою контексту автентифікації, темізації, мовної локалізації та обліку платформи користувача.

3.1.1. Дизайн інтерфейсу

Для побудови інтерфейсу використовувалася бібліотека Fluent UI яка є частиною дизайн-системи Microsoft. Ця бібліотека забезпечує:

- сумісність із виглядом і поведінкою елементів Microsoft 365;
- уніфікований стиль компонентів (кнопки, таблиці, меню, дропдауни тощо);
- підтримку accessibility (доступності) та адаптивного дизайну;

- автоматичну підтримку тем оформлення, що відповідає темі SharePoint.

Центральний елемент застосунку складається з трьох вкладок, які відповідають різним статусам академічних оцінок.

Вкладка «*Поточні*» містить дисципліни, які зараз вивчаються студентом або ще не завершені. У кожній картці в цій вкладці виводиться:

- назва предмета;
- викладач;
- бали ECTS;
- семестр;
- гранична дата складання;
- тип контролю (залік, екзамен).

Вкладка «*Академічні борги*» відображає дисципліни, які були не складені у визначений термін. Для таких записів інтерфейс містить:

- назва предмета;
- викладач;
- бали ECTS;
- семестр;
- термін безкоштовного перескладання;
- чи почалося платне перескладання;
- тип контролю (залік, екзамен).

Вкладка «*Здано*» містить завершені дисципліни, оцінки з яких вже зафіксовані. Для таких записів інтерфейс відображає:

- назва предмета;
- викладач;
- семестр;
- дата здачі;
- чи перезарахування;
- тип контролю (залік, екзамен).

3.1.2. Авторизація користувача

Однією з ключових особливостей клієнтської частини застосунку є її глибока інтеграція з Microsoft 365 через платформу SPFx. Завдяки цьому застосунок має змогу працювати в контексті поточного користувача, отримувати від імені цього користувача авторизаційні токени та передавати їх до серверної частини (Web API) для безпечного обміну даними.

Після завантаження веб-частини (web part) у SharePoint, SPFx автоматично надає доступ до контексту середовища. Цей контекст надає змогу не лише відображати персоналізований інтерфейс, а й виконувати авторизовані HTTP-запити до зовнішніх API, зареєстрованих у Azure AD.

Для отримання access token з Microsoft Entra ID та подальших авторизованих запитів на серверну частину, SPFx надає сервіс AADHttpClientFactory. Це частина API SPFx, яка спрощує автентифіковану взаємодію з захищеними веб-сервісами. Принцип роботи:

У manifest.json вебчастини заздалегідь реєструється зовнішній ресурс API з його Azure AD Application ID URI, та під час виконання запиту використовується `this.context.aadHttpClientFactory.getClient()`, де SPFx автоматично отримує access token для зареєстрованого ресурсу (API), кешує його на сесію, та додає його у заголовок Authorization для кожного запиту.

Клієнтська частина не зберігає жодних облікових даних користувача, натомість весь процес автентифікації та авторизації виконується прозоро для користувача за рахунок інтеграції SPFx з Azure AD.

3.1.3. Взаємодія з серверною частиною

Далі, нам потрібна послідовна взаємодію з API.

Після ініціалізації вебчастини у SharePoint, першим кроком є запит до ендпоінту для отримання метаданих користувача:

```
GET /api/v1/academic-performance/prefetch/me
```

Цей запит надсилається автоматично після отримання access token від Microsoft через AADHttpClientFactory, і передбачає визначення **контексту користувача**. Серверна частина на основі токена встановлює:

- а) чи є користувач студентом університету або коледжу;
- б) чи має користувач академічну заборгованість.

Отримана відповідь містить структуровані метадані, які зберігаються у стані React-компонента і визначають подальшу поведінку інтерфейсу (відображення сповіщення про борги та налаштування фільтрів).

Після визначення профілю користувача, SPFx-застосунок формує основний запит для завантаження оцінок:

```
GET /api/v1/academic-performance/grades/me?filter=Debt&isCollege=true
```

Після отримання відповіді у форматі JSON, застосунок:

- а) перевіряє, чи не виникла помилка або порожній результат;
- б) зберігає масив оцінок у стан (useState);
- в) ініціює візуалізацію даних у вигляді карток оцінок.

3.2. Особливості реалізації серверної частини

Серверна частина застосунку реалізована у вигляді REST Web API, створеного на основі сучасного фреймворку ASP.NET Core 8 із використанням мови програмування C#. Цей вибір зумовлений не лише високим рівнем інтеграції технологій Microsoft із платформою SharePoint, а й перевагами, які надає ASP.NET Core як міжплатформне, продуктивне та масштабоване середовище для створення веб-сервісів. Однією з ключових особливостей ASP.NET Core є висока продуктивність - він демонструє одні з найкращих результатів у тестах навантаження серед усіх фреймворків для веб-розробки. З головних переваг фреймворку є:

- вбудована підтримка DI (ін'єкції залежностей), яка дозволяє створювати гнучкі та легко тестовані архітектури;

- розширена підтримка безпеки, включаючи аутентифікацію, авторизацію та інтеграцію з зовнішніми провайдерами (наприклад, Microsoft Entra ID);
- підтримка асинхронного програмування, що дозволяє ефективно обробляти одночасні запити;
- гнучка маршрутизація та middleware-пайплайн, що забезпечують контроль над обробкою HTTP-запитів;
- інтеграція з сучасними стандартами API - OpenAPI (Swagger), REST, JSON, CORS.

У контексті даного проєкту ASP.NET Core дозволяє побудувати стабільний, продуктивний і безпечний серверний застосунок.

3.2.1. Обробка запитів на Web API

Серверна частина застосунку «Електронна відомість студента» побудована відповідно до REST-архітектурного стилю (Representational State Transfer). Це означає, що API розроблено таким чином, щоб кожен запит від клієнта був ідемпотентним, чітко структурованим, орієнтованим на ресурси та використовував стандартні HTTP-методи (GET, POST, PUT, DELETE тощо) для взаємодії з цими ресурсами.

У реалізації даного застосунку REST-інтерфейси слугують мостом між фронтенд частиною, створеною на SPFx, і внутрішніми даними, що отримуються із SharePoint через CSOM.

Основний запит, який ініціює SPFx-застосунок для завантаження академічних оцінок поточного користувача, має наступний вигляд:

```
GET /api/v1/academic-performance/grades/me
```

Запит обробляється контролером, що відповідає за обробку запитів до оцінок. У цьому запиті передаються query-параметри, які впливають на результат:

- `filter` - тип оцінок, який необхідно завантажити (значення перелічуються через `enum`);

- Current - поточні оцінки (наприклад, ще не завершені курси);
- Debt - предмети, за якими є академічна заборгованість;
- Completed - успішно завершені предмети, перенесені в архів;
- isCollege - булеве значення (true або false), яке вказує, чи потрібно шукати оцінки в списках, призначених для студентів коледжу.

Окремий запит, що викликається на етапі ініціалізації застосунку для отримання загальної інформації про студента, має такий вигляд:

```
GET /api/v1/academic-performance/prefetch/me
```

Цей запит потрібен для визначення типу студента (студент університету чи коледжу) та з'ясування наявності академічної заборгованості.

На кожному етапі обробки запиту виконується валідація вхідних даних, щоб запобігти некоректному використанню API. Якщо запит містить неочікуване значення або відсутні обов'язкові параметри - повертається відповідь з HTTP-кодом 400 Bad Request. У разі спроби звернення без дійсного токена - 401 Unauthorized, а при порушенні прав доступу - 403 Forbidden.

3.2.2. Авторизація студента

Для забезпечення контролю доступу до серверної частини застосунку використовується аутентифікація користувача яка реалізована за допомогою Azure AD. На сервері реалізовано ретельну перевірку дійсності JWT-токена перед тим, як надати доступ до ресурсів застосунку:

Спочатку токен перевіряється з використанням асиметричного ключа, який отримується з відкритого ключа Microsoft. Це дозволяє переконатися, що токен дійсно був виданий Microsoft Entra ID і не був змінений. Такий підхід критично важливий у випадку, якщо токен потрапив у руки злоумисника - будь-яке підроблення буде виявлено, оскільки підпис не пройде перевірку.

Токен має містити правильне значення aud, яке відповідає ідентифікатору бекенд-застосунку в Azure AD. Це гарантує, що токен призначений саме для цього API, а не був виданий для доступу до інших служб.

Додатково виконується перевірка ідентифікатора застосунку, щоб переконатися, що токен був виданий дозволеним клієнтським застосунком (SPFx), а не будь-яким іншим джерелом. Така перевірка виключає можливість зловживання токенами з інших застосунків навіть у межах тієї самої організації.

Якщо токен успішно проходить усі перевірки, з нього зчитується ідентифікатор користувача.

3.2.3. Отримання оцінок

Університет «КРОК» зберігає академічні оцінки студентів у списках SharePoint. Для їх отримання з серверної частини використовується бібліотека Microsoft.SharePoint.CSOM. Цей інструмент надає зручне і повнофункціональне API для взаємодії з об'єктами SharePoint, включно з можливістю запитувати та фільтрувати вміст списків.

Після успішної аутентифікації користувача, серверна частина використовує його унікальний ідентифікатор, витягнутий з JWT-токена, для пошуку відповідних записів в одному з SharePoint-списків.

Для вибірки оцінок застосовується CAML Query (Collaborative Application Markup Language) - це XML-подібна мова, що дозволяє фільтрувати та сортувати елементи списків у SharePoint.

3.2.4. Повернення результатів

Після виконання запиту серверна частина перетворює елементи списку на DTO-моделі (Data Transfer Object), які серіалізуються у формат JSON та повертаються клієнту.

3.3. Тестування

3.3.1. Тестування клієнтської частини

Для забезпечення надійності та коректної роботи інтерфейсу клієнтської частини застосунку було проведено мануальне тестування. У контексті

проєкту, де інтерфейс побудований з використанням React, SPFx і Fluent UI, ручне тестування дозволило оперативно перевірити ключові сценарії взаємодії користувача з системою, не потребуючи повноцінної автоматизованої інфраструктури.

Тестування проводилось у браузерях Microsoft Edge, Google Chrome та Firefox на різних розширеннях екранів, у тому числі мобільних режимах. Також перевірялась коректність роботи в темній та світлій темах.

Під час тестування використовувались облікові записи студентів, авторизованих через Azure AD, що дозволяло перевіряти повну інтеграцію з Microsoft 365 та серверною частиною застосунку.

Завантаження застосунку. Перевірка, що SPFx вебчастина коректно ініціалізується на сторінці та виконує первинний запит на `api/v1/academic-performance/prefetch/me`.

Вкладки. Перевірка перемикачів між «Поточні», «Академічні борги» та «Здано», коректного завантаження даних у кожній вкладці.

Карточки дисциплін. Перевірка структури та динамічного відображення полів в залежності від вкладки.

Пошук. Перевірка швидкості та точності пошуку за назвою предмета, у різних вкладках.

Фільтри запиту. Перевірка правильності параметрів `filter` і `isCollege` при зміні налаштувань.

Помилки. Перевірка реакції застосунку на мережеві помилки (відсутність підключення, 401 Unauthorized, порожні результати).

Мобільна адаптація. Перевірка зручності інтерфейсу на смартфонах та планшетах, перевірка responsive-верстки.

3.3.2. Тестування серверної частини

Для перевірки правильності реалованої логіки використовувалося юніт-тестування (unit testing), яке дозволяє виявити помилки на ранніх етапах розробки та гарантує коректність роботи окремих модулів.

У якості фреймворку для юніт-тестів було обрано xUnit - це сучасна, гнучка і широко підтримувана платформа для модульного тестування в середовищі .NET. xUnit дозволяє писати легкі, читаємі й ізольовані тести, що швидко виконуються та легко інтегруються з CI/CD-пайплайнами.

Для імітації зовнішніх залежностей, таких як сервіси, інтерфейси або доступ до SharePoint, використовувалась бібліотека Moq. Вона дозволяє створювати моки - підроблені об'єкти, які поведуться як справжні залежності, але з повним контролем з боку тесту.

Основною метою юніт-тестування було перевірити бізнес-логіку серверної частини.

Наприклад, як коректну обробку вхідних параметрів, або роботу в разі помилкових або граничних умов (некоректні значення, відсутні дані тощо)

Тестування покривало сервісні шари та контролери, які не мають прямої залежності від зовнішніх API (наприклад, CSOM), оскільки такі залежності замінювалися моками.

3.4. Документація продукту

Застосунок доступний безпосередньо через сайт SharePoint. Після авторизації у Microsoft 365 за вашим студентським обліковим записом:

1. Перейдіть на сторінку кабінет студента, де розміщено застосунок.
2. Зачекайте кілька секунд, поки інтерфейс завантажиться.
3. Ви автоматично будете авторизовані, і система отримає інформацію про ваші метадані та оцінки.

Інтерфейс складається з трьох вкладок, кожна з яких відображає різні типи академічної інформації:

1. Поточні. Виводяться дисципліни, які студент наразі вивчає. У картках вказано дату, до якої необхідно скласти предмет, щоб уникнути боргу. Показується тип контролю (екзамен, залік) та поточна оцінка, якщо вона вже є.

2. Академічні борги. Виводяться предмети, які не були складені вчасно. Вказано дату завершення періоду безкоштовного перескладання. Якщо предмет більше не викладається, це зазначається у коментарі.

3. Здано. Виводяться завершені предмети з оцінками. Вказано, чи була оцінка перезарахована. Зазначено, чи складено дисципліну в межах сесії.

У верхній частині інтерфейсу є рядок пошуку. Ви можете ввести повну або часткову назву предмета. Пошук застосовується до активної вкладки, а список автоматично оновлюється відповідно до запиту.

У правому верхньому куті інтерфейсу знаходиться іконка. Натиснувши на неї, ви відкриєте панель налаштувань, де можна: змінити мову інтерфейсу (українська / англійська) і перемкнути джерело даних з університету на коледж. Налаштування застосовуються миттєво без перезавантаження сторінки.

ВИСНОВКИ

У ході виконання проєкту було успішно розроблено веб-парту «Електронна відомість студента» на платформі SharePoint, яка вирішує актуальні проблеми доступу студентів до академічної інформації.

Основними результатами роботи є:

- створено інтуїтивно зрозумілий, адаптивний та швидкий інтерфейс, який суттєво покращує досвід взаємодії студентів із власними оцінками та академічними боргами;
- реалізовано стабільний та захищений механізм взаємодії між клієнтською частиною (на базі SharePoint Framework, React, TypeScript та Fluent UI) і серверною частиною (ASP.NET Core з використанням C#);
- забезпечено високий рівень безпеки та контролю доступу через інтеграцію з Azure Active Directory, що повністю відповідає сучасним стандартам захисту даних;
- архітектура продукту дозволяє легко розширювати функціонал, що відкриває потенціал для подальшого розвитку застосунку, включаючи інтеграцію додаткових функцій та модулів.

Водночас важливо зазначити, що поточна версія проєкту є лише першим етапом у його реалізації. Попереду ще багато роботи: планується додавання нового функціоналу, удосконалення інтерфейсу, оптимізація серверної частини, а також інтеграція з іншими системами університету. Незважаючи на це, вже зараз продукт має основний функціонал, готовий до практичного використання, і забезпечує вирішення поставлених завдань на даному етапі.

Таким чином, створений продукт ефективно відповідає поставленій меті та готовий до впровадження в освітній процес Університету «КРОК», сприяючи покращенню навчального середовища для студентів і зменшенню адміністративного навантаження на працівників закладу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кабінет Студента [Електронний ресурс]. URL: <https://livekrokedu.sharepoint.com/sites/KROK/student-cabinet/SitePages/ProjectHome.aspx> (дата звернення 18.06.2025.)
2. Power Apps [Електронний ресурс]. URL: <https://www.microsoft.com/en-us/power-platform/products/power-apps> (дата звернення 18.06.2025.)
3. Google Classroom [Електронний ресурс]. URL: <https://edu.google.com/workspace-for-education/products/classroom> (дата звернення 18.06.2025.)
4. Moodle [Електронний ресурс]. URL: <https://moodle.org/> (дата звернення 18.06.2025.)
5. Microsoft Entra ID (formerly Azure Active Directory) [Електронний ресурс]. URL: <https://www.microsoft.com/en-us/security/business/identity-access/microsoft-entra-id> (дата звернення 18.06.2025.)
6. Fluent UI – Get Started [Електронний ресурс]. URL: <https://developer.microsoft.com/en-us/fluentui#/> (дата звернення 18.06.2025.)
7. On-behalf-of flows with MSAL.NET [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/entra/msal/dotnet/acquiring-tokens/web-apps-apis/on-behalf-of-flow> (дата звернення 18.06.2025.)
8. What is RESTful API [Електронний ресурс]. URL: <https://aws.amazon.com/what-is/restful-api/> (дата звернення 18.06.2025.)

ДОДАТКИ

Додаток А

Скріншоти клієнтської частини застосунку

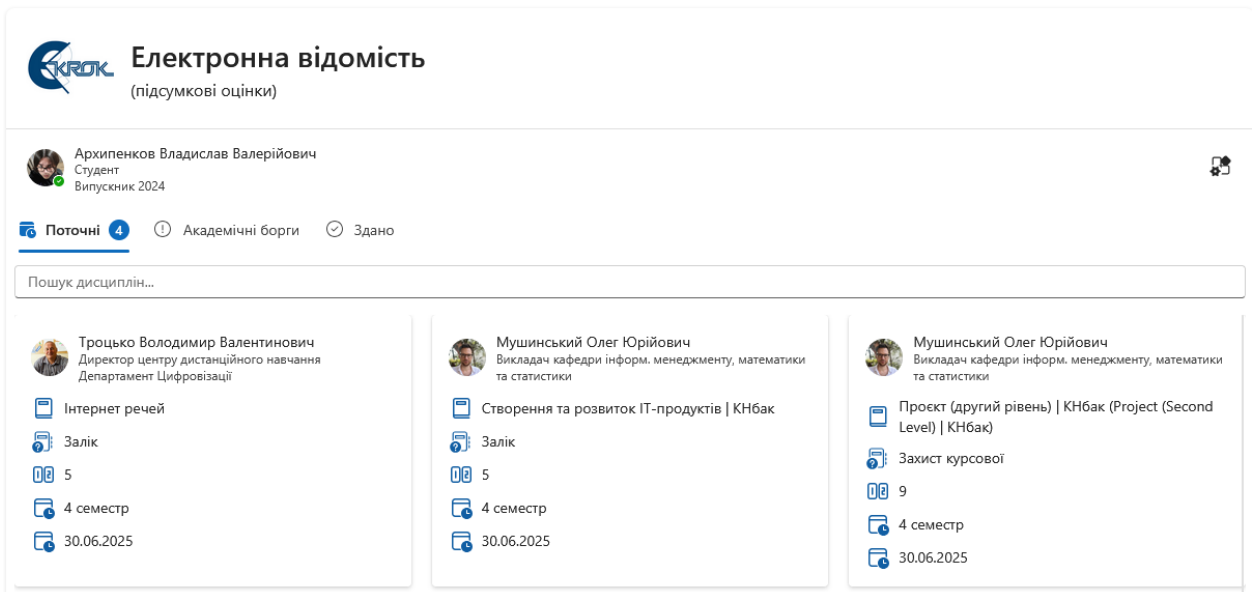


Рисунок А.1 – Вкладка «Поточні оцінки»

Джерело: розроблено авторами

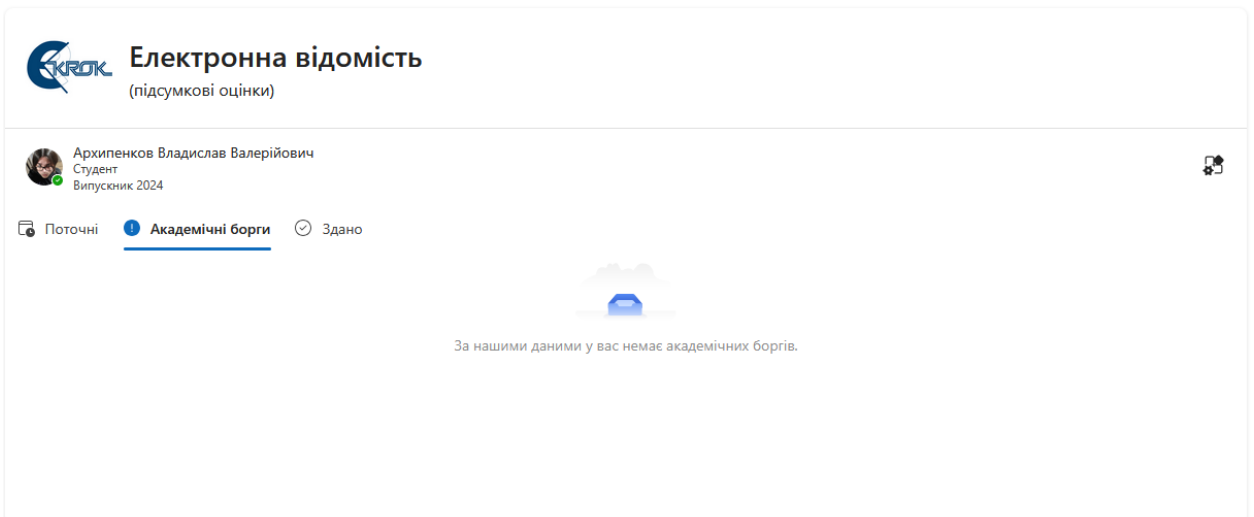


Рисунок А.2 – Вкладка «Борги» (пусто)

Джерело: розроблено авторами

Скріншоти клієнтської частини застосунку

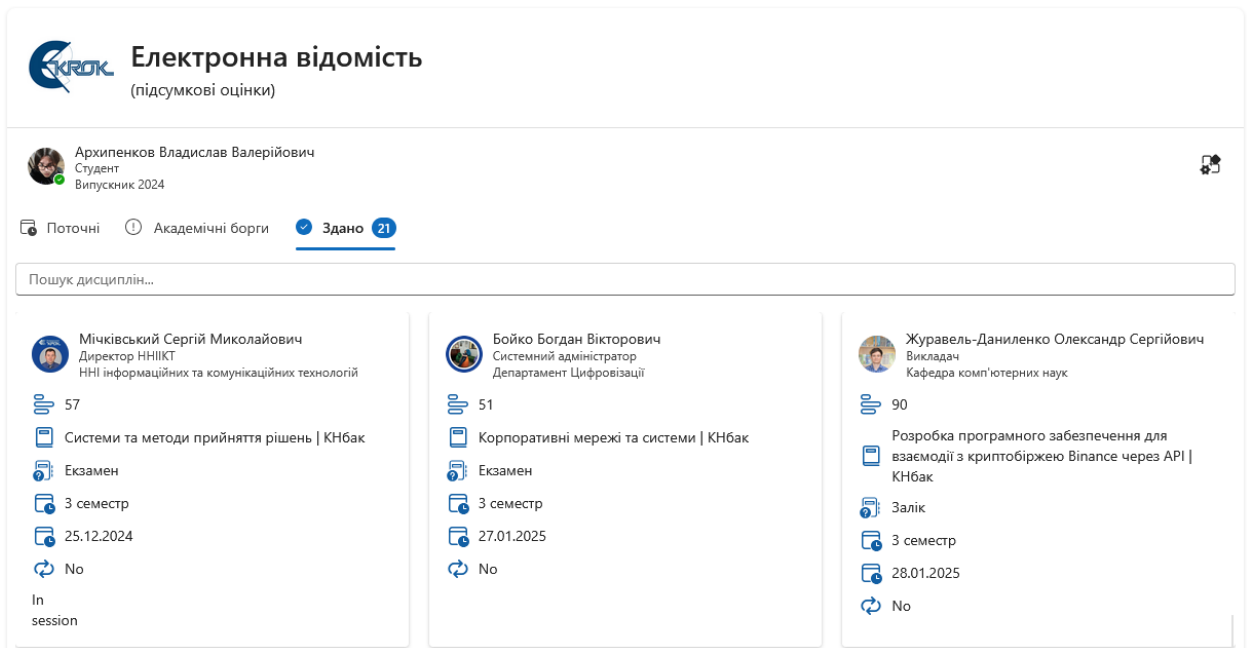


Рисунок Б.1 – Вкладка «Здано»

Джерело: розроблено авторами

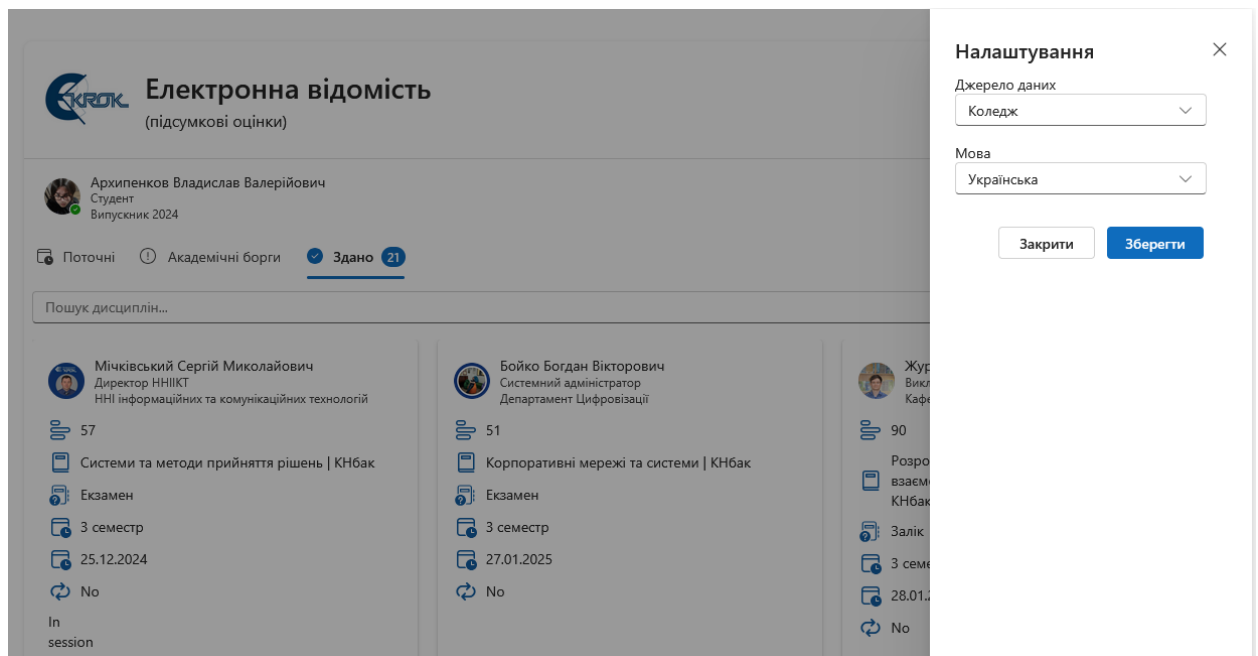


Рисунок Б.2 – Налаштування

Джерело: розроблено авторами