

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»  
(назва факультету)

Кафедра «Електронні обчислювальні машини»  
(повна назва кафедри)

Пояснювальна записка  
до кваліфікаційної роботи  
бакалавра  
(ступінь вищої освіти)

на тему: Розробка програмних засобів веб-системи дистанційного обліку навчальної успішності

за освітньою програмою Комп'ютерна інженерія

зі спеціальності: 123 Комп'ютерна інженерія  
(шифр і назва спеціальності)

Виконав: студент групи: КС1811

(підпис студента)

/ Дмитро КРИКУНОВ /

(Ім'я ПРІЗВИЩЕ)

Керівник:

(підпис)

/ ст. викладач, Володимир ДЗЮБА /

(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер:

(підпис)

/ ст. викладач, Володимир ДЗЮБА /

(посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

(підпис)

Дніпро – 2022 рік

**Ministry of Education and Science of Ukraine**  
**Ukrainian State University of Science and Technologies**

Computer Technologies and Systems

(faculty)

Electronic Computers

(department)

**Explanatory Note**  
**to Bachelor's Thesis**  
(higher education degree)

on the topic: Development of web-based software for remote accounting of academic performance

according to educational curriculum Computer Engineering

in the Speciality: 123 Computer Engineering

(speciality and its code)

Done by the student of the group: KC1811 / Dmytro Krykunov /  
(name, surname)

Scientific Supervisor: B DSV / Sr. Lecturer, Volodymyr Dziuba /  
(position, name, surname)

Normative controller: B DSV / Sr. Lecturer, Volodymyr Dziuba /  
(position, name, surname)

**Supervisors**

\_\_\_\_\_  
(Chapter title heading) / (position, name, surname) /

\_\_\_\_\_  
(Chapter title heading) / (position, name, surname) /

\_\_\_\_\_  
(Chapter title heading) / (position, name, surname) /

\_\_\_\_\_  
(Chapter title heading) / (position, name, surname) /

**Міністерство освіти і науки України**  
**Український державний університет науки і технологій**

Факультет: Комп'ютерні технології і системи  
Кафедра: Електронні обчислювальні машини  
Рівень вищої освіти: Перший (бакалаврський)  
Освітня програма: Комп'ютерна інженерія  
Спеціальність: 123 Комп'ютерна інженерія  
(шифр та назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ЕОМ

Ігор ЖУКОВИЦЬКИЙ  
(Ім'я ПРІЗВИЩЕ)

(підпис)

Дата \_\_\_\_\_

**ЗАВДАННЯ**

на кваліфікаційну роботу

бакалавра

(ступінь вищої освіти)

студенту Крикунову Дмитру Валентиновичу

(Прізвище, Ім'я По батькові)

1. Тема роботи: Розробка програмних засобів веб-системи дистанційного обліку навчальної успішності.

Керівник роботи: Дзюба Володимир Володимирович

(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від

"07" грудня 2021 р. № 68ст

2. Строк подання студентом роботи: 17.06.2022 р.

3. Вихідні дані до роботи: Приклад навчальної відомості поточного та модульного контролів

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Огляд сучасних систем обліку навчальної успішності

4.2 Основна частина:

Огляд сучасних систем обліку навчальної успішності


Вибір технологічного стеку для розробки системи

Розробка системи обліку навчальної успішності

## КАЛЕНДАРНИЙ ПЛАН


№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ та постановка завдання	4.03.2022	5%
2	Огляд сучасних систем обліку навчальної успішності	18.03.2022	20%
3	Вибір технологічного стеку для розробки системи	15.04.2022	30%
4	Розробка системи обліку навчальної успішності	20.05.2022	40%
5	Висновки та рекомендації	27.05.2022	5%
6	Подання кваліфікаційної роботи до кафедри	17.06.2022	
7	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	22.06.2022	

Студент

  
(підпис)

Дмитро КРИКУНОВ  
(Ім'я ПРІЗВИЩЕ)

Керівник роботи

  
(підпис)

Володимир ДЗЮБА  
(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра:  
47с., 18 рис., 1 табл., 3 додатки, 15 джерел.

Об'єкт розробки – веб-системи дистанційного обліку навчальної успішності, її алгоритм та програмна реалізація.

Мета роботи – поліпшення процесу передачі інформації щодо навчальних відомостей на кафедру, збереження дати та часу відправки відомості викладачем та її отримання на кафедрі.

Методи дослідження – аналіз існуючого програмного забезпечення, подібного до нашого, його порівняння та тестування програмної реалізації.

Одержані результати – розроблено веб-додаток по технології SPA. Архітектура програми влаштована так, що при початковому запуску користувач бачить основний контент сайту в браузері, а нові дані завантажуються на ходу при необхідності, наприклад, при прокручуванні або кліку на кнопку. Використання цієї технології дозволило нам досягти максимальної швидкодії та кращого досвіду використання додатку. Для розробки веб-системи було використано сучасний JavaScript фреймворк, а саме React.

Результати роботи можуть бути застосовані в університеті та поліпшити процес передачі відомостей на навчальну кафедру.

Ключові слова: ВЕБ-СИСТЕМА, ОБЛІК УСПІШНОСТІ СТУДЕНТІВ, ДИСТАНЦІЙНЕ НАВЧАННЯ, НАВЧАЛЬНА КАФЕДРА, ВИКЛАДАЧ.

## ЗМІСТ

РЕФЕРАТ .....	5
ВСТУП ТА ПОСТАНОВКА ЗАВДАННЯ .....	6
1 ОГЛЯД СУЧАСНИХ СИСТЕМ ОБЛІКУ НАВЧАЛЬНОЇ УСПІШНОСТІ .....	8
1.1 Застосування та можливості систем обліку навчальної успішності.....	8
1.2 Огляд систем обліку навчальної успішності.....	8
1.2.1 Електронний журнал «Щоденник» .....	9
1.2.2 Електронний журнал «Additio App» .....	10
1.2.3 Електронний журнал «TeacherKit» .....	11
1.3 Висновки огляду сучасних систем обліку навчальної успішності .....	13
2 ВИБІР ТЕХНОЛОГІЧНОГО СТЕКУ ДЛЯ РОЗРОБКИ СИСТЕМИ.....	14
2.1 Вибір технологічного стеку .....	14
2.1.1 Вибір технологій для розробки клієнтської частини .....	14
2.1.2 Вибір технологій для розробки серверної частини .....	17
2.1.3 Вибір бази даних .....	18
2.2 Висновки .....	19
3 РОЗРОБКА СИСТЕМИ ОБЛІКУ НАВЧАЛЬНОЇ УСПІШНОСТІ.....	20
3.1 Опис функцій системи.....	20
3.1.1 Створення облікового запису та вхід до системи.....	20
3.1.2 Управління доступом на основі ролей.....	25
3.1.3 Створення профілів студентів та викладачів .....	27
3.1.4 Створення навчальних груп та предметів .....	29
3.1.5 Заповнення й відправка відомостей на навчальну кафедру .....	31
3.2 Висновки .....	32
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ.....	33
ПЕРЕЛІК ПОСИЛАНЬ.....	34
ДОДАТОК А.....	<b>ПОМИЛКА! ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.</b>
ДОДАТОК Б .....	<b>ПОМИЛКА! ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.</b>
ДОДАТОК В.....	<b>ПОМИЛКА! ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.</b>

## ВСТУП ТА ПОСТАНОВКА ЗАВДАННЯ

У грудні 2019 року пандемія COVID-19 втрутилася в життя всього світу. Ця сумнозвісна подія безсумнівно внесла зміни в повсякденність кожного з нас. Фахівці ще й досі радять усім залишатися вдома, та вже понад два роки більшість університетів світу працюють онлайн використовуючи в освітньому контексті різноманітні цифрові пристрої, а саме комп'ютери, ноутбуки та планшети. Інтеграція інформаційно-комунікаційних технологій (ІКТ) та використання різноманітних програм показує, що якщо їх вміло використовувати, вони можуть допомогти подолати багато бар'єрів, що виникають у традиційних вузах.

У цей час як ніколи гостро стоїть проблема організації навчального процесу та обліку навчальної успішності. Студент не завжди має змогу відвідати університет і принести викладачу квиток контрольного терміну, що потім буде внесений до інформаційної системи ВНЗ. Також іноді виникають ситуації коли асистент, або лектор не має можливості зі стану здоров'я заповнити навчальну відомість, і це робить спеціально визначений для цього контрольний викладач кафедри. У цьому випадку інформацію щодо відомостей можна передати за допомогою інтернету, але її необхідно зберігати, та фіксувати час внесення цієї інформації, щоб позбавити відомості помилок. Надати такі можливості можуть так звані системи обліку навчальної успішності.

Таким чином, дослідження та розробка даної системи є досить актуальною темою.

В цій системі передбачаються наступні функції: створення облікового запису та вхід до системи; реалізація управління доступом на основі ролей; створення профілів студентів та викладачів; створення навчальних груп та предметів; заповнення й відправка відомостей на навчальну кафедру.

Завдання, що потрібно виконати:

- огляд сучасних систем обліку навчальної успішності;
- вибір технологічного стеку для розробки;
- опис функцій розроблюваної системи;
- розробка програмного забезпечення клієнтської частини;
- розробка програмного забезпечення серверної частини.

# 1 ОГЛЯД СУЧАСНИХ СИСТЕМ ОБЛІКУ НАВЧАЛЬНОЇ УСПІШНОСТІ

Для розробки системи обліку навчальної успішності, необхідно розглянути подібні системи, які вже існують на ринку.

## 1.1 Застосування та можливості систем обліку навчальної успішності

Система обліку навчальної успішності - це засіб спілкування, створений для сприяння комунікації між викладачами, кураторами груп, адміністрацією університету та студентами. Використання системи обліку навчальної успішності суттєво допомагає керівництву університету, при чому не лише ректору або працівникам деканату, а й звичайним викладачам, які займаються заповненням різноманітних статистичних даних. Система обліку навчальної успішності замінює всі заходи, пов'язані з адмініструванням університету. У той час як у традиційному підході до навчального процесу викладач заповнює усі необхідні данні вручну, система обліку навчальної успішності пропонує заздалегідь визначені послуги, які автоматично перевіряються та обчислюються програмою. Порівняно з традиційним підходом адміністрування, електронний доступ значно швидший, тому викладачі та куратори груп можуть витратити більше часу на роботу зі студентами та планування викладацької діяльності, ніж на офіційну сторону заповнення різноманітних відомостей.

## 1.2 Огляд систем обліку навчальної успішності

Найрозповсюдженіший тип систем обліку навчальної успішності це так звані електронні щоденники та електронні журнали, тому ми розглянемо саме їх у якості потенційних конкурентів.

## 1.2.1 Електронний журнал «Щоденник»

Електронний журнал «Щоденник» – це сучасна безкоштовна освітня мережа, створена в Україні, яка формує зручне електронне середовище для викладачів, учнів та батьків. Проект був ініційований Міністерством освіти і науки, молоді та спорту України. Метою цієї мережі є об'єднання усіх учасників навчального процесу України в єдину спільноту та впровадження сучасних комп'ютерних технологій у школах [1]. «Щоденник» надає велику кількість різноманітних функцій для учасників навчального процесу та значно покращую його якість. Приклад інтерфейсу електронного журналу «Щоденник» наведений на рисунку 1.1.

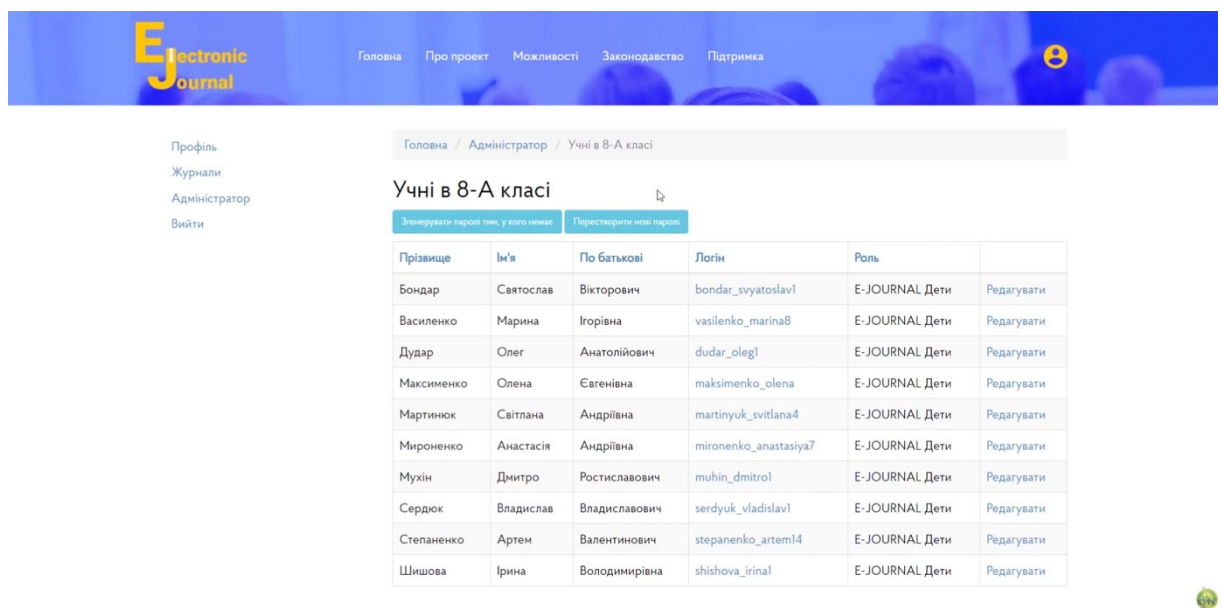


Рисунок 1.1 – Інтерфейс електронного журналу «Щоденник»

Позитивними можливостями «Щоденника» є:

- «Розклад уроків» (розклад уроків для учнів і викладачів, доступний у всіх видах звітних періодів);
- «Електронний щоденник» (надається всім учням, наявне сортування оцінок за певний предмет або період часу);

- «Домашні завдання» (наявний функціонал за допомогою якого викладач може задавати домашні завдання та контролювати їх виконання).

Негативними особливостями «Щоденника» є:

- діяльність цього журналу спрямована лише для шкіл;
- відсутність налаштувань інтерфейсу;
- значна кількість функцій ще розробляється.

## 1.2.2 Електронний журнал «Additio App»

Електронний журнал «Additio App» – це простий у використанні журнал оцінювання успішності, що дозволяє викладачеві зручно день у день керувати навчальним процесом учнів [2]. Приклад інтерфейсу електронного журналу «Additio App» наведений на рисунку 1.2.

Mathematics 4A Lab 107		Activities and Exam						
		Exercise 1	Exercise 2 Problems	Average exercises 30%	Comments	Exams topic 1 50%	Attention in class	Final average
1.	Hurtado, Rafael	3.9	6 / 10 ↑	5	The student has a good attitude, it is trying hard to improve little by little.	6.2	😊	5.7
2.	Vidal, Sergi	6.5	6.9 / 10	6.75	It stays in line, could improve.	6.2	😐	6.43
3.	Losada, Omar	9.2	8.7 / 10	9	It deserves to be congratulated - good progress and behaviour.	9	😊	8.6
4.	Casanova, Teresa	10	10 / 10	10	The student likes and is engaged with the subject. Help their colleagues.	9.8	😊	9.9
5.	Hinojosa, Andrés	2.5	5 / 10	3.75	Do not pay attention It is necessary to talk with parents.	4.3	😐	4.28
6.	Trias, Lucas	5	5.3 / 10	5.25	It has improved, is happier than before.	5.9	😐	6.03
7.	Cortés, Marta	7	6.9 / 10	7	It is attentive, works well and helps their colleagues.	6.5	😊	6.95
8.	Gibert, Karina	5.3	5.1 / 10 ↓	5.25	It annoys their companions constantly. We need to have a chat.	4.9	😐	5.13

Рисунок 1.2 – Інтерфейс електронного журналу «Additio App»

Електронний журнал «Additio App» має такі основні можливості:

- має інструмент для оцінювання студентів, який легко налаштовується і може бути адаптованим до будь-якої системи оцінювання;
- можливість перенесення оцінок студентів в Excel за допомогою генератора формул;
- можливість використання без підключення до Інтернету.

Однак, електронний журнал «Additio App» має й ряд недоліків, до них належать:

- відсутність української мови;
- електронний журнал є платним.

### **1.2.3 Електронний журнал «TeacherKit»**

Електронний журнал «TeacherKit» – це персональний органайзер у сфері навчально-виховного та освітнього процесу [3]. Цей додаток доступний на багатьох системах, таких як Android, Mac OS та Windows. Він являється найпопулярнішим електронним журналом у світі, та користується попитом у більше ніж одного мільйона викладачів. Приклад інтерфейсу електронного журналу «TeacherKit» наведений на рисунку 1.3.

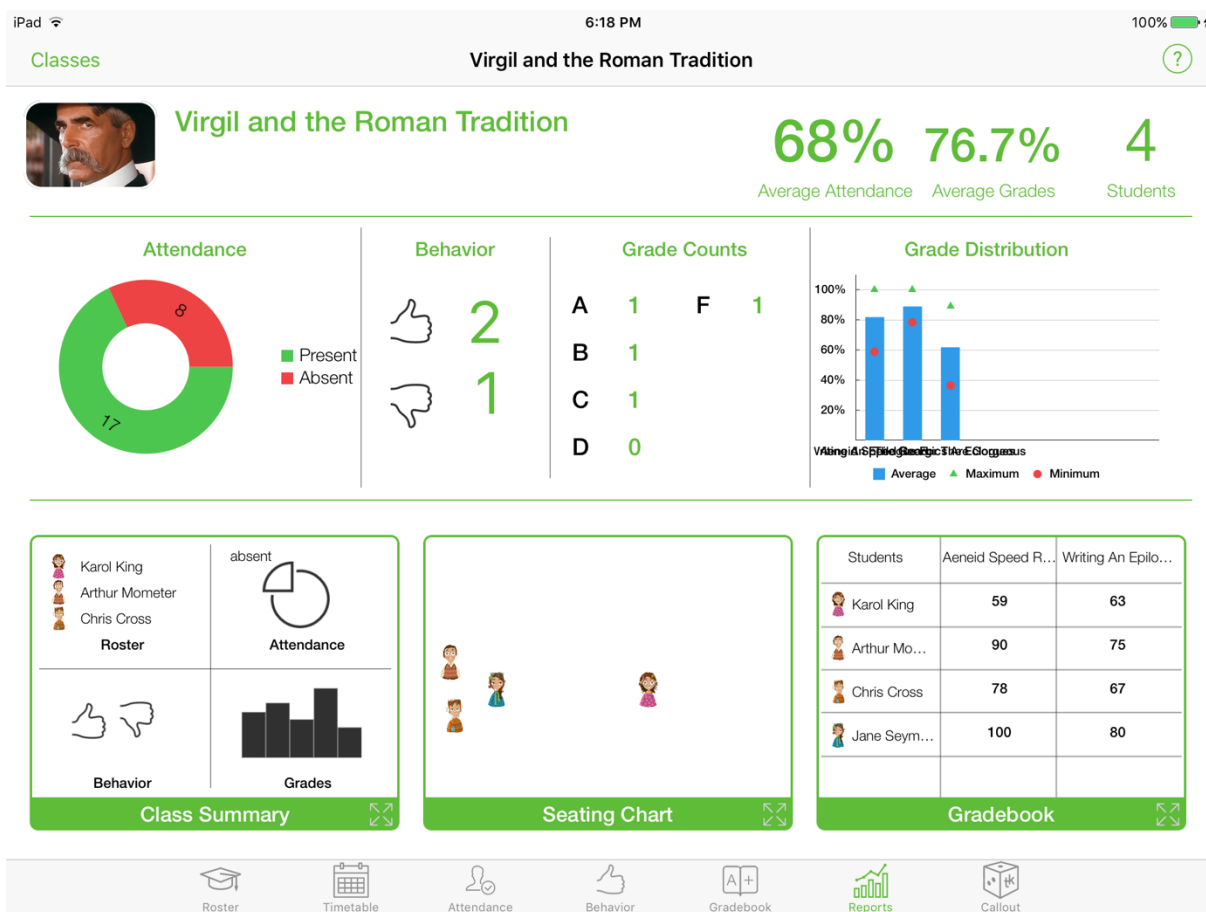


Рисунок 1.3 – Інтерфейс електронного журналу «TeacherKit»

Суттєвими перевагами електронного журналу «TeacherKit» є:

- можливість додавання студентів у журнал за допомогою синхронізації контактів з іншим гаджетом;
- автоматичне резервне копіювання даних щодня;
- створення шкали оцінювання залежно від потреб користувача;
- перегляд відвідуваності та успішності студентів у діаграмах.

Негативними рисами «TeacherKit» є:

- відсутність української мови;
- звіти, які створюються за допомогою цього додатку є платними.

### 1.3 Висновки огляду сучасних систем обліку навчальної успішності

Основні параметри сучасних систем обліку навчальної успішності зведемо до порівняльної таблиці 1.1.

Таблиця 1.1 – Порівняння сучасних систем обліку навчальної успішності

	Можливість налаштування інтерфейсу	Доступність використання без сплати коштів	Підтримка імпорту та експорту	Мова інтерфейсу	Візуалізація інформації за допомогою діаграм
Щоденник	немає	повна	відсутня	укр. мова	немає
Additio App	доволі гнучка	відсутня	є експорт у форматі Excel	немає укр. мови	немає
TeacherKit	доволі гнучка	майже повна	синхронізація та резервне копіювання	немає укр. мови	є

Отже, провівши огляд, було виявлено перелік проблем, які є на даний час у сучасних систем обліку навчальної успішності. Зазвичай це малий перелік доступних функцій, відсутність україномовного інтерфейсу користувача та відсутність візуалізації інформації. Варто відзначити, що сучасні продукти поступово намагаються вирішити їх, але це доволі складне завдання.

Аналіз сучасних систем дозволить нам прийняти набагато точніше рішення щодо шляхів подальшої розробки власної системи обліку навчальної успішності.

Вся система буде складатися з серверної та клієнтської частин. Клієнтська частина підтримуватиме постійний зв'язок з серверною, посилаючи запити до серверної частини. Сервер у цьому випадку буде аналізувати запит, перевіряти чи має даний користувач до читання або редагування запрошеної інформації. Після цього будуть зроблені потрібні зміни у базі даних, згенерована та надіслана відповідь назад до клієнтської частини.

## **2 ВИБІР ТЕХНОЛОГІЧНОГО СТЕКУ ДЛЯ РОЗРОБКИ СИСТЕМИ**

Технологічний стек кардинально впливає на будь-який ІТ-проект, будь-то серйозна банківська система, або звичайний додаток для збереження заміток. Залежно від технологічного стека змінюється час, якість та вартість розробки. Більше того, він впливає на масштабованість, тому якщо почати розробку на погано підбраному технологічному стеку, це може спричинити низку проблем у майбутньому. Саме тому варто придати цій темі особливе значення.

### **2.1 Вибір технологічного стеку**

Технологічний стек – це набір мов програмування, фреймворків та ПЗ, необхідних для розробки програми [4].

Оскільки усі веб-додатки складаються з клієнтської та серверної частин, вимоги до їх функціональності суттєво впливають на вибір мови програмування, фреймворків та програмного забезпечення, за допомогою яких буде вестися розробка [5].

#### **2.1.1 Вибір технологій для розробки клієнтської частини**

Клієнтська сторона – це видима частина веб-програми, з якою взаємодіють користувачі. Вона складається з 3 основних частин:

- JavaScript – мова програмування, яка відповідає за інтерактивну частину веб-програми.
- HTML – мова розмітки документів, яка потрібна для правильного відображення веб-програми в браузері.
- CSS – формальна мова, яка потрібна для правильної стилізації веб-програми.

Важливу роль також відіграє обраний JavaScript фреймворк, адже хоч кодування й можливе без його використання, але правильно підібране середовище може значно полегшити роботу й зменшити час і ціну розробки.

Фреймворки JS – це бібліотеки програмування JavaScript, в яких є попередньо написаний код для використання у стандартних функціях та завданнях програмування. Це основа для створення веб-сайтів або веб-застосунків. Більшість фреймворків являються безкоштовними та маюць відкритий вихідний код.

Фреймворки дійсно значно підвищують продуктивність, з їх використанням розробникам доводиться писати меншу кількість коду вручну, тому що вже є заздалегідь написані і готові до використання функції та шаблони. Багато компонентів веб-сайту зазвичай не є унікальними з функціональної точки зору, тому дуже часто ми маємо можливість просто розширювати заздалегідь створені компоненти, та створювати власні за потреби.

Найбільш поширеними фреймворками є React, Angular та Vue.js. Ми не будемо детально зосереджуватись на огляді кожного з них, адже вони усі роблять майже одну й ту ж саму роботу, але трохи різними способами. Іноді навіть трапляються ситуації коли при виборі фреймворку основним критерієм стає досвід роботи команди з тим чи іншим фреймворком. Наразі найбільш популярним та перспективним із цих фреймворків являється React, тому ми обрали для розробки саме його.

React – це UI-бібліотека, яка була створена компанією-гігантом Facebook і яка більш ніж за 7 років існування зібрала навколо себе багатомільйонну спільноту [6]. Вона базується на компонентах багаторазового використання. Простіше кажучи, це блоки коду, які можуть бути написані за допомогою класового або функціонального підходу, та в подальшому перезастосовані у багатьох частинах додатку. Кожен компонент представляє певну окрему частину сторінки, таку як логотип, кнопку або поле введення. Вигляд та функціонал компонентів залежить від параметрі, що їм передаються, тому один компонент може мати різний вигляд у різних місцях, що додає розробнику ще більше

гнучкості. Також особливістю React є використання JSX, синтаксису XML, який поєднує JavaScript і HTML. Це не просто шаблон JavaScript, це повний JavaScript який дозволяє використовувати усі мови цієї мови програмування.

Клієнтська частина нашого додатку має дуже тісно взаємодіяти з серверною, так як усі дані зберігаються саме там. Щоб усунути у користувача почуття того, що дані завантажуються дуже повільно, було вирішено використати бібліотеку `react-query` [7]. Головні переваги які надає ця бібліотека – це кешування та усунення дублікації запитів. Це дозволяє покращити оптимізацію додатку та зменшити навантаження на мережу. Приклад використання `react-query` при завантаженні профілю наведено на рисунку 2.1.

```
1 import React from "react";
2 import { useQuery } from "react-query";
3 import axios from "axios";
4
5 export function Profile() {
6   const { isLoading, error, data, isFetching } = useQuery( queryKey: "profileData", queryFn: () =>
7     axios.get( url: "https://api.com/user/1").then((result : AxiosResponse<any> ) => result.data),
8   );
9
10  if (isLoading) return "Завантаження...";
11
12  if (error) return "Сталася помилка: " + error.message;
13
14  return (
15    <div>
16      <h1>{data.name}</h1>
17      <p>{data.description}</p>
18      <div>{isFetching ? "Оновлення..." : ""}</div>
19    </div>
20  );
21 }
22
```

Рисунок 2.1 – Приклад використання `react-query` при завантаженні профілю

З прикладу вище ви також можете помітити що для здійснення запитів ми використовуємо `axios` [8] замість сучасного `fetch` API браузера [9]. Це пов'язано з тим сучасний API підтримується не всіма браузерами та може спровокувати помилку в старіших версіях, чого ми допустити не можемо. Окрім широкої підтримки браузерів `axios` також пропонує нам більш зручний та інтуїтивно зрозумілий синтаксис.

## 2.1.2 Вибір технологій для розробки серверної частини

Серверна частина веб-застосунку – це те, що не бачить користувач, тому що вона знаходиться під клієнтською частиною.

Так як ми вже маємо JavaScript на клієнтській стороні і додаток буде розроблятися одним розробником – буде доцільним розглянути використання цієї мови програмування також і при написанні серверної частини. Щоб цього досягнути, ми вирішили використовувати Node.js.

Node.js – це міжплатформне середовище виконання JavaScript із відкритим вихідним кодом, яке працює на движку V8 і виконує код JavaScript за межами веб-браузера [10]. Node.js дозволяє нам створювати всі види серверних інструментів і програм на JavaScript. Простими словами, Node.js представляє собою парадигму «JavaScript всюди», що об'єднує розробку веб-додатків навколо однієї мови програмування, замість звичного підходу з використанням різних мов для сценаріїв на стороні сервера та клієнта. У більшості випадків, при написанні додатків на Node.js, у парі з ним використовують Express.

Express є найпопулярнішим веб-фреймворком Node та водночас базовою бібліотекою для багатьох інших популярних веб-фреймворків [11]. Він забезпечує механізми для:

- Написання обробників HTTP запитів для різних URL-шляхів (маршрутів).
- Інтеграція з механізмами візуалізації, генеруючи відповіді, вставляючи дані в шаблони.
- Встановлення загальних для додатку параметрів, таких як порт для підключення або розташування шаблонів, які використовуються для генерування відповіді.
- Створення додаткового «проміжного програмного забезпечення» для обробки запитів у будь-якій точці конвеєра обробки запитів.

Ми можемо зобразити схему взаємодії серверної та клієнтської частин (рисунок 2.2) таким чином:

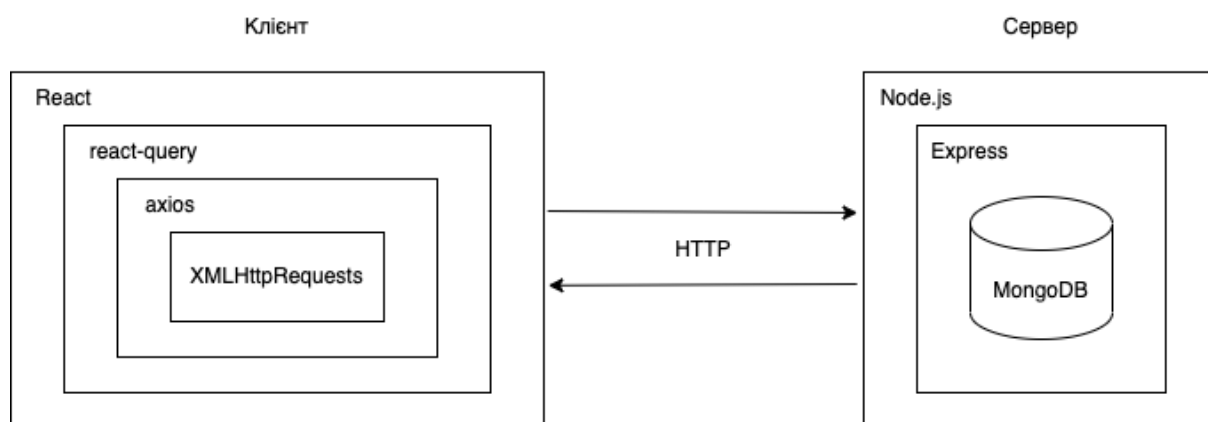


Рисунок 2.2 – Схема взаємодії серверної та клієнтської частин

Хоча сам Express є досить мінімалістичним, він має дуже багато вже сумісних пакетів проміжного програмного забезпечення для вирішення практично будь-яких проблем веб-розробки. Існують бібліотеки для роботи з файлами куки, сеансами, параметрами URL-адреси, заголовками безпеки та багатьма іншими.

### 2.1.3 Вибір бази даних

Так як ми вже обрали основний стек нашого додатку – буде доцільно здійснювати вибір бази даних опираючись на нього. Виходячи з цього – найбільш підходящою для нас базою даних в нашій ситуації є MongoDB.

MongoDB – це документно-орієнтована NoSQL база даних, яка використовується для зберігання великих обсягів даних. На заміну таблиць і рядків, як у реляційних базах даних, вона представляє колекції та документи. Документи складаються з пар ключ-значення, які є основною одиницею даних у MongoDB. Колекції містять набори документів і функцій, що є еквівалентом таблиць реляційної бази даних [12].

Переваги використання MongoDB:

- Можливість масштабованості;
- Висока швидкість збереження та пошуку даних;
- Зрозумілий людині формат збереження даних;
- Шардінг (розділення даних між різними фізичними серверами);
- Просте налаштування середовища;
- Високий рівень технологічної підтримки від провайдера.

## 2.2 Висновки

Під час вибору технологічного стеку було обрано MERN стек (рисунок 2.3), що розшифровується як MongoDB, Express, React, Node, які є чотирма ключовими технологіями, з яких складається стек [13].

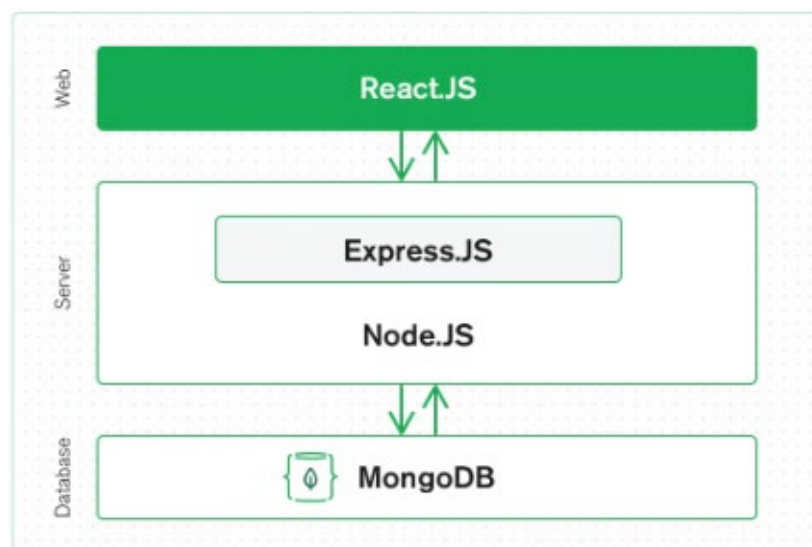


Рисунок 2.3 – Схематичне зображення MERN стеку

Цей стек є одним з найпопулярніших стеків технологій для написання веб додатків. Кожна з цих 4 потужних технологій вирішує окрему проблему, і відіграє велику роль у розробці додатку.

## **3 РОЗРОБКА СИСТЕМИ ОБЛІКУ НАВЧАЛЬНОЇ УСПІШНОСТІ**

У попередньому розділі ми здійснили вибір технологічного стеку, але подальша розробка не можлива без чітко сформульованих умов до продукту який ми реалізуємо. Вирішити цю проблему нам допоможе детальний опис функцій нашої системи.

### **3.1 Опис функцій системи**

В системі дистанційного обліку навчальної успішності передбачені наступні функції:

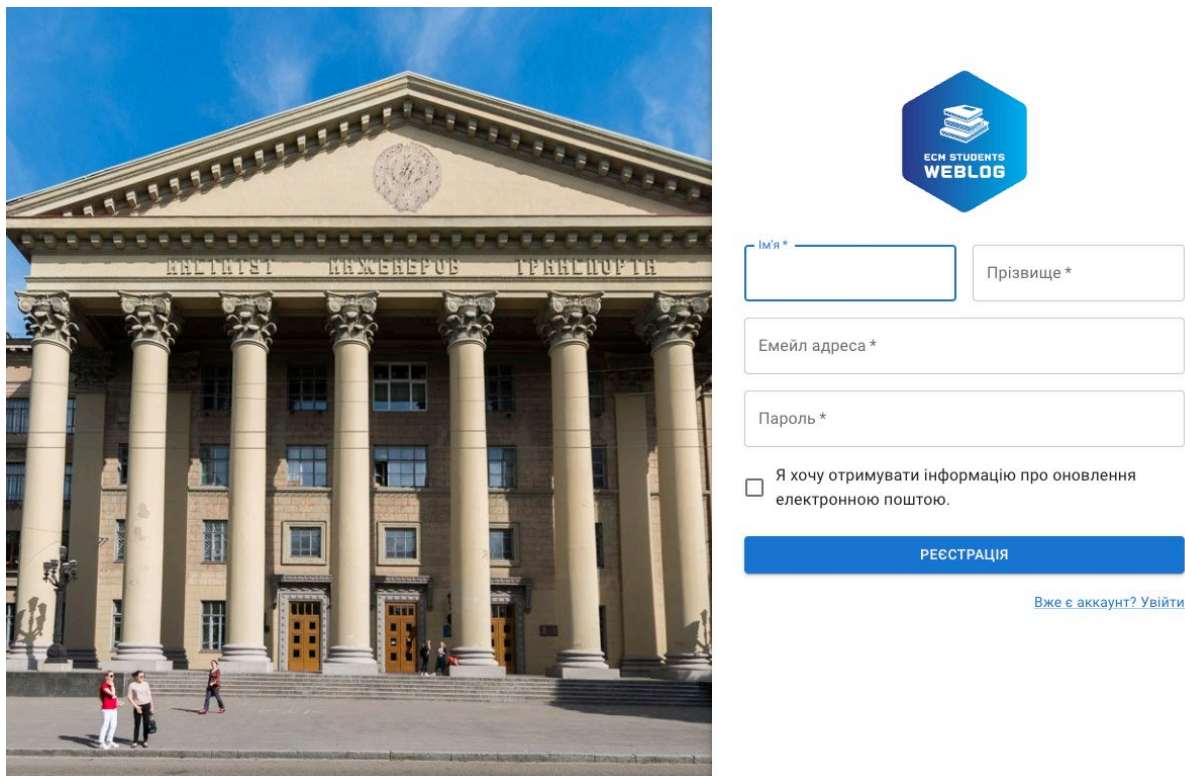
- створення облікового запису та вхід до системи;
- управління доступом на основі ролей;
- створення профілів студентів та викладачів;
- створення навчальних груп та предметів;
- заповнення й відправка відомостей на навчальну кафедру.

Кожній з функцій буде присвячено окремий розділ, у якому буде детально розглянуто її особливості та підхід до реалізації.

#### **3.1.1 Створення облікового запису та вхід до системи**

Кожен із нас повсякденно користується десятками, а можливо і сотнями інтернет ресурсів та має свій власний облікового запису у багатьох із них. Облікові записи необхідні для збереження даних про користувача, його розпізнавання (аутифікації) та надання доступу до його особистих даних та налаштувань. Наша система не є виключенням і також потребує наявності облікового запису для її використання.

В першу чергу для створення нового облікового запису в нашій системі користувач має перейти до сторінки реєстрації (рисунок 3.1) та заповнити обов'язкові поля «ім'я», «прізвище», «електронна пошта» та «пароль».



The registration page for ECM Students Weblog consists of a header with the logo, followed by a registration form. The form includes the following elements:

- Input field for Name (ім'я \*)
- Input field for Surname (Прізвище \*)
- Input field for Email address (Емейл адреса \*)
- Input field for Password (Пароль \*)
- Checkbox for receiving information about updates via email:  Я хочу отримувати інформацію про оновлення електронною поштою.
- A blue button labeled РЕЄСТРАЦІЯ.
- A link: [Вже є аккаунт? Увійти](#)

Рисунок 3.1 – Зовнішній вигляд сторінки реєстрації

Якщо усі поля заповнено коректно та користувач згоден з умовами використання, він може натиснути на кнопку «zareєstruvatisia». При натисканні на цю кнопку на сервер буде відправлено запит (рисунок 3.2) із вказаною інформацією.



The screenshot shows a REST client interface for a POST request to the endpoint `/auth/register` with the description "Register as user". The request body is a JSON object:

```
{
  "name": "fake name",
  "email": "fake@example.com",
  "password": "password1"
}
```

Рисунок 3.2 – Приклад запиту на реєстрацію нового користувача





Емейл адреса \*

Пароль \*

Запам'ятати мене

[УВІЙТИ](#)

[Забули пароль?](#) [Немає акаунту? Реєстрація](#)

Рисунок 3.4 – Зовнішній вигляд сторінки входу в систему

Але як зрозуміти що при наступному після реєстрації або входу в систему запиті до нас звертається саме цей користувач та що він має право на перегляд інформації? Процес перевірки достовірності користувача називається аутентифікацією і далі буде розглянуто один з методів її реалізацій – за допомогою JSON веб-токенів.

JSON веб-токен (англ. JSON Web Token, JWT) – це інтернет-стандарт для створення даних із додатковим підписом та/або необов'язковим шифруванням, корисне навантаження якого містить JSON. Токени підписуються або за допомогою приватного секрету, або публічного/приватного ключа [14].

JWT токен складається з трьох окремих частин: заголовка, корисного навантаження та підпису. Перші два елементи – це JSON об'єкти певної структури. Третій елемент обчислюється на основі перших і залежить від обраного алгоритму. Три частини кодуються окремо за допомогою Base64url і об'єднуються за допомогою крапок для створення JWT.

JWT токени в клієнт серверних додатках частіше за все поділяють на:

- Access-токен — це токен, який надає його власнику доступ до захищеної інформації сервера. Зазвичай він має короткий термін життя і може містити в собі додаткову інформацію, таку як IP-адреса сторони, що запитує даний токен.
- Refresh-токен — це токен, що дозволяє клієнтам вимагати нові access-токени після закінчення їх часу життя. Ці токени зазвичай видаються на тривалий термін.

Схематичне зображення типового процесу аутентифікації наведено на рисунку 3.5.

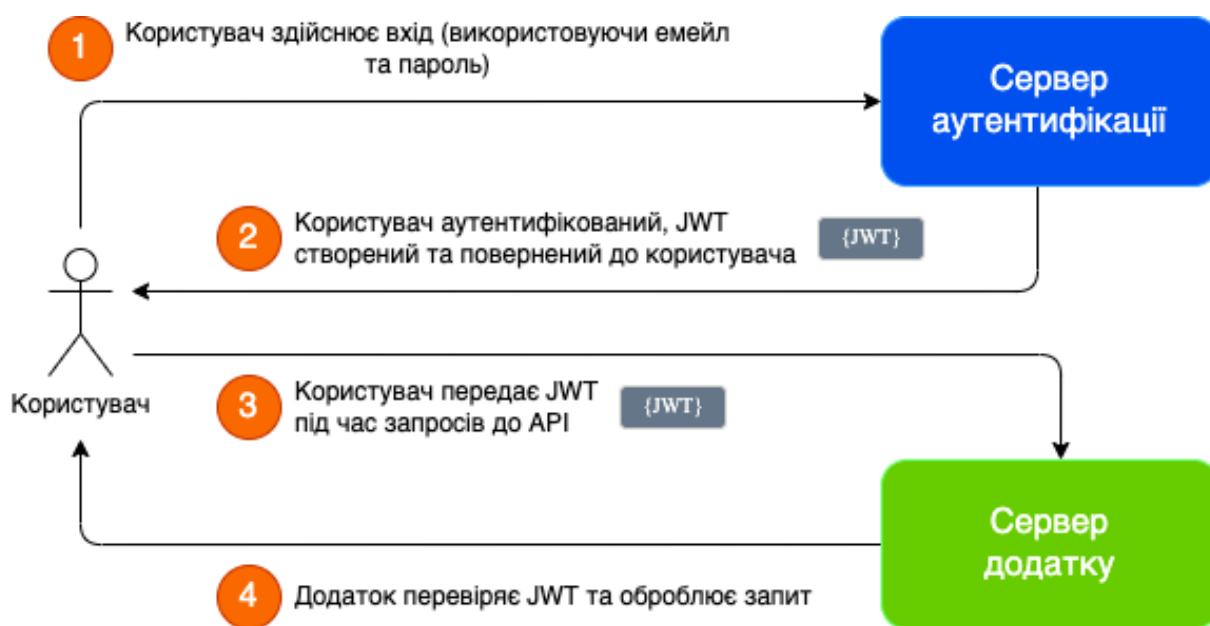


Рисунок 3.5 – Схематичне зображення процесу аутентифікації

У нашому додатку реалізована така схема використання JSON-токенів:

1. Клієнт проходить аутентифікацію в додатку (з використанням емейлу та паролю);
2. У разі успішної аутентифікації клієнту будуть надіслані access- і refresh-токени;

3. При подальших зверненнях до сервера клієнт має передати access-токен. Сервер перевіряє токен та надає клієнту доступ до інформації;
4. У випадку, якщо спливає час життя access-токену, він стає невалідним, і клієнт має відправити свій refresh-токен, у відповідь на який сервер надасть два оновлені токени;
5. Якщо ж сплив час життя refresh-токену, клієнт повинен знову пройти процес аутентифікації (п. 1).

### **3.1.2 Управління доступом на основі ролей**

Після розпізнання користувача в системі ми потребуємо також зрозуміти, чи має він доступ до запрошуваної інформації. Для цього ми маємо реалізувати у нашій системі один з доступних підходів до управління доступом. Одним з найрозповсюдженіших з них являється управління доступом на основі ролей, цей підхід є складовою частиною багатьох сучасних комп'ютерних систем та вже встиг себе зарекомендувати. Рольовий підхід часто використовується в системах, для користувачів яких чітко визначено коло їх посадових повноважень та обов'язків, як у нашому випадку.

Управління доступом на основі ролей (англ. Role Based Access Control, RBAC) — розвиток політики вибіркового управління доступом, при цьому права доступу користувачів системи до функціоналу та даних групуються з урахуванням специфіки їх застосування, створюючи ролі [15].

Формування ролей покликане визначити чіткі та зрозумілі для користувачів системи правила розмежування доступу. Рольове розмежування доступу дозволяє реалізувати гнучкі, динамічно змінювані в процесі функціонування правила розмежування доступу.

У нашій системі буде сформовано управління доступом на підставі ролей, таких як «адміністратор», «студент» та «викладач». В залежності від ролі користувач отримуватиме доступ до інших функцій системи. Наприклад, адміністратори мають повний доступ до всіх функцій системи, викладачі можуть

споглядати інформацію щодо студентів своїх груп та заповнювати навчальні відомості, а студенти – тільки споглядати інформацію, яка до них відноситься.

Схему управління доступом реалізованого у нашій системі наведено на рисунку 3.6.

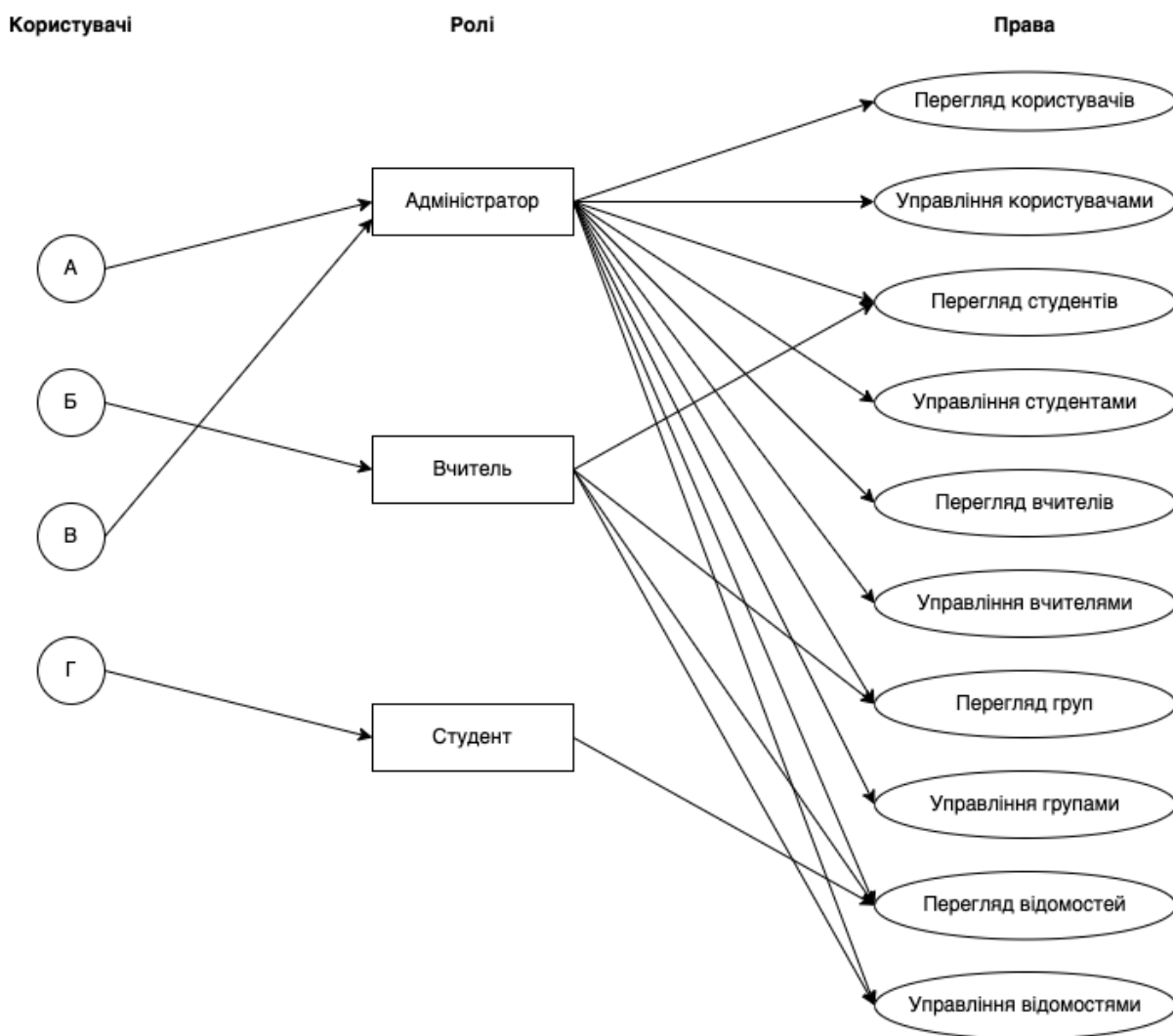


Рисунок 3.6 – Управління доступом на основі ролей

В подальшому можливе збільшення варіативності ролей, або розширення їх функціоналу, але управління даний підхід до управління доступом – дійсно найкращий вибір для нашої системи.

### 3.1.3 Створення профілів студентів та викладачів

Профілі студентів та викладачів необхідні для збереження інформації пов'язаної з студентами викладачами відповідно. Було прийнято рішення відділити ці профілі від облікових записів користувачів через те що деякі учасники навчального процесу можуть ще не мати власних аккаунтів у системі, але вже бути. Такий підхід також забезпечує набагато легше та прозоріше управління поступом – усі користувачі можуть редагувати свої власні облікові записи, але доступ до створення та редагування профілів студентів і викладачів мають лише адміністратори системи. Для цього процесу у системі відведено окремі сторінки (рисунок 3.7, 3.8 та 3.9).

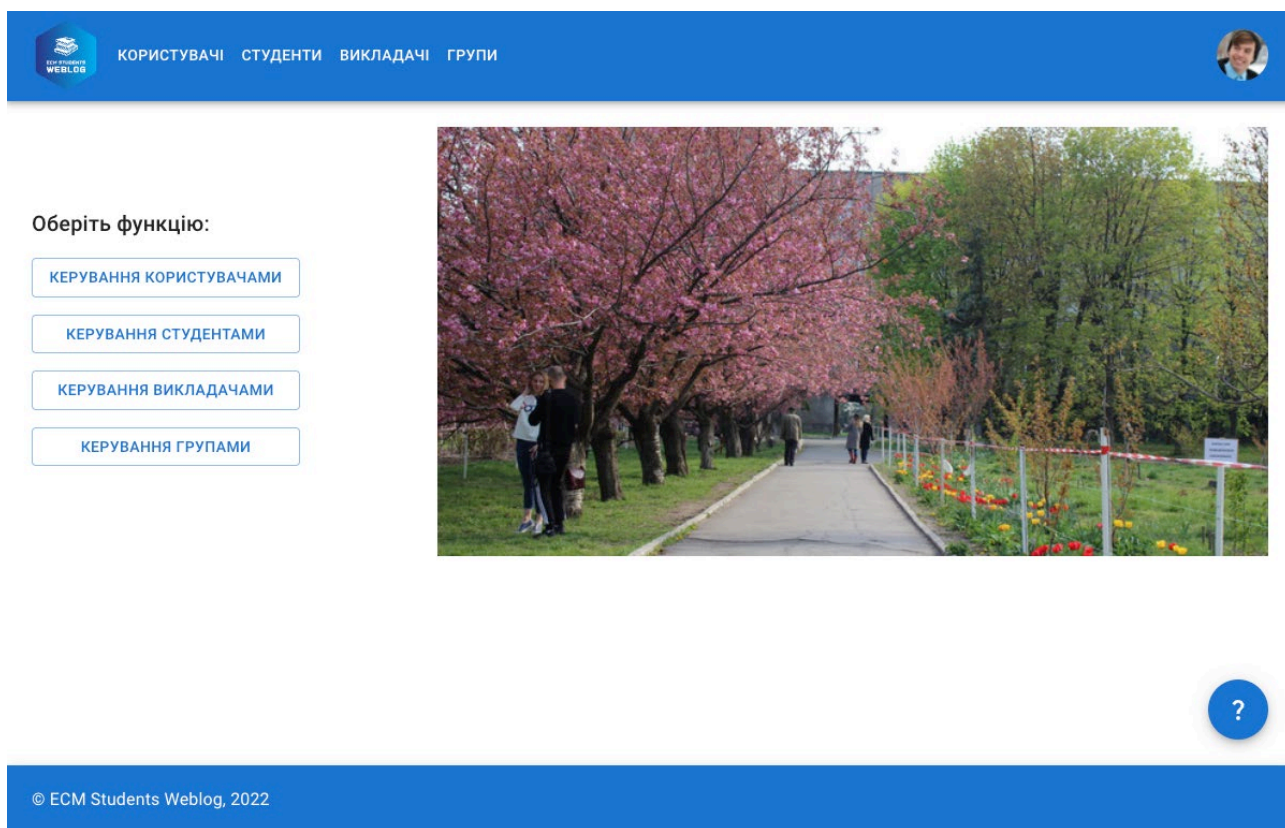
















Рисунок 3.7 – Головна сторінка додатку

КЕРУВАННЯ ВИБРОМ

КОРИСТУВАЧІ СТУДЕНТИ ВИКЛАДАЧІ ГРУПИ

Головна / Студенти

Ім'я	Номер індивідуального навчального плану	Емейл	Телефон	
Осипчук Йоган Семенович	120507	osup.yog@gmail.com	+380665120507	 
Сомко Ладислав Ярославович	120512	somko.lad@gmail.com	+380665120512	 
Гладун Устим Іванович	120514	gladun.ust@gmail.com	+380665120514	 
Крикунов Дмитро Валентинович	120516	dmytro.krykunov@gmail.com	+380665120516	 
Панчишин Радомир Олександрович	120517	panch.rad@gmail.com	+380665120517	 
Проненко Анастасій Олександрович	120519	pron.an@gmail.com	+380665120519	 
Винниченко Еммануїл Остапович	120521	emmanuel.v@gmail.com	+380665120521	 













[ДОДАТИ СТУДЕНТА](#)

Рисунок 3.8 – Сторінка управління студентами


КЕРУВАННЯ ВИБРОМ

КОРИСТУВАЧІ СТУДЕНТИ ВИКЛАДАЧІ ГРУПИ

Головна / Викладачі

Ім'я	Вчене звання	Емейл	Телефон	
Загальський Киян Глібович	Професор	zagul@gmail.com	+380665187291	 
Каспрук Гатусил Денисович	Доцент	kaspruk.gat@gmail.com	+380665187292	 
Піскун Милодар Герасимович	Професор	piskun.mil@gmail.com	+380665187293	 
Богаченко Никодим Захарович	Доцент	bogachenko.nikon@gmail.com	+380665187294	 
Балі Чеслав Устимович	Асистент	bali.cheslav@gmail.com	+380665187295	 
Володимир Володимирович Дзюба	Старший викладач	dv2000@gmail.com	+380665187296	 

[ДОДАТИ ВИКЛАДАЧА](#)



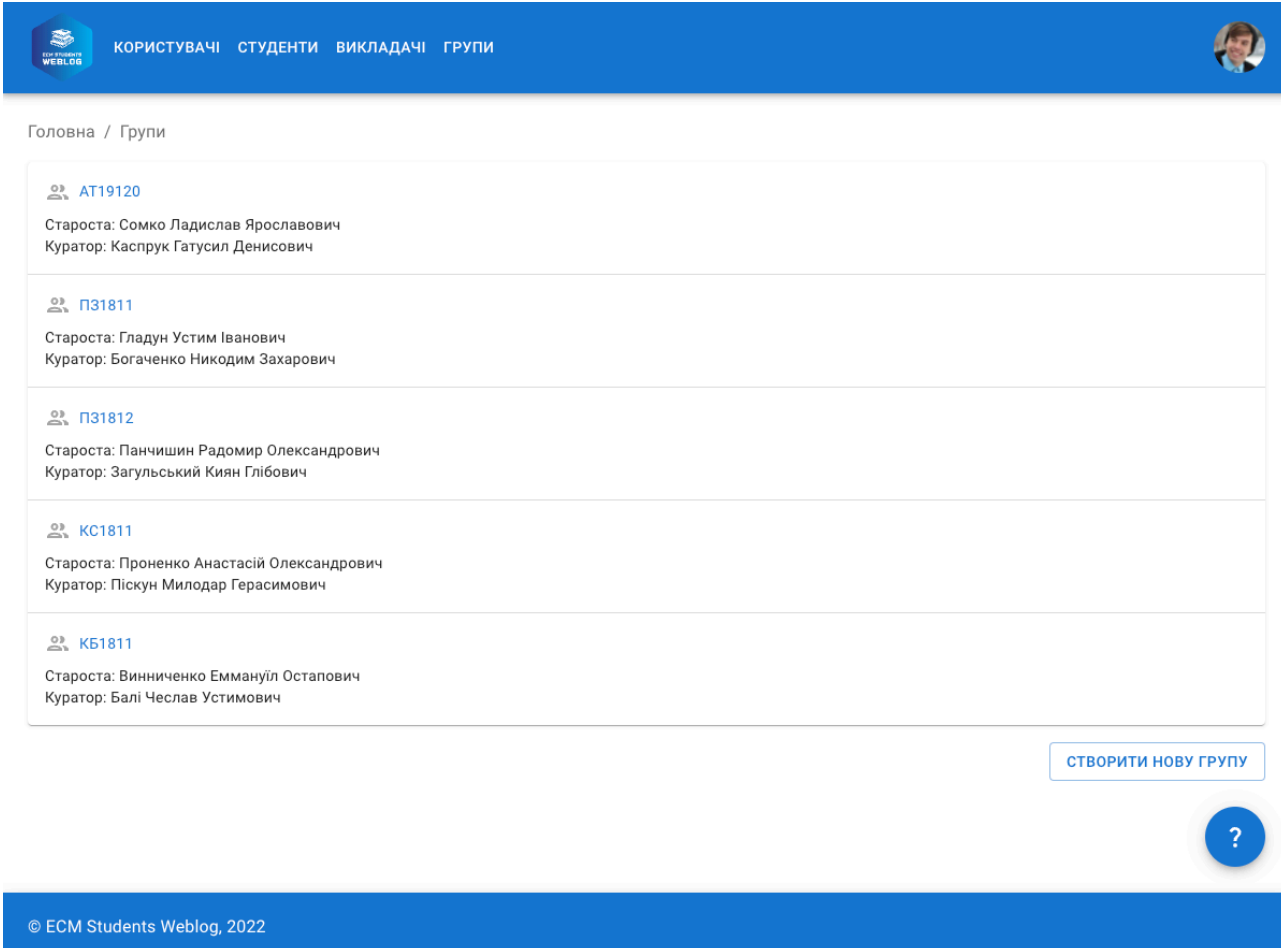
© ECM Students Weblog, 2022

Рисунок 3.9 – Сторінка управління викладачами

Адміністратори можуть «приєднати» профіль студента або викладача до облікового запису будь-якого користувача, що надасть йому права до перегляду своїх оцінок, або керування групами відповідно до типу профілю.






### 3.1.4 Створення навчальних груп та предметів

Після створення необхідних профілів ми можемо приступити до формування навчальних груп. Для управління передбачена окрема сторінка додатку (рисунок 3.10), на якій є можливість додати до групи студентів, призначити її старосту та куратора, а також визначити дати старту та закінчення поточного навчального року для окремої групи.



Користувачі | Студенти | Викладачі | Групи

Головна / Групи

 AT19120 Староста: Сомко Ладислав Ярославович Куратор: Каспрук Гатусил Денисович
 ПЗ1811 Староста: Гладун Устим Іванович Куратор: Богаченко Никодим Захарович
 ПЗ1812 Староста: Панчишин Радомир Олександрович Куратор: Загальський Киян Глібович
 КС1811 Староста: Проненко Анастасій Олександрович Куратор: Піскун Милодар Герасимович
 КБ1811 Староста: Винниченко Еммануїл Остапович Куратор: Балі Чеслав Устимович

[СТВОРИТИ НОВУ ГРУПУ](#)

?

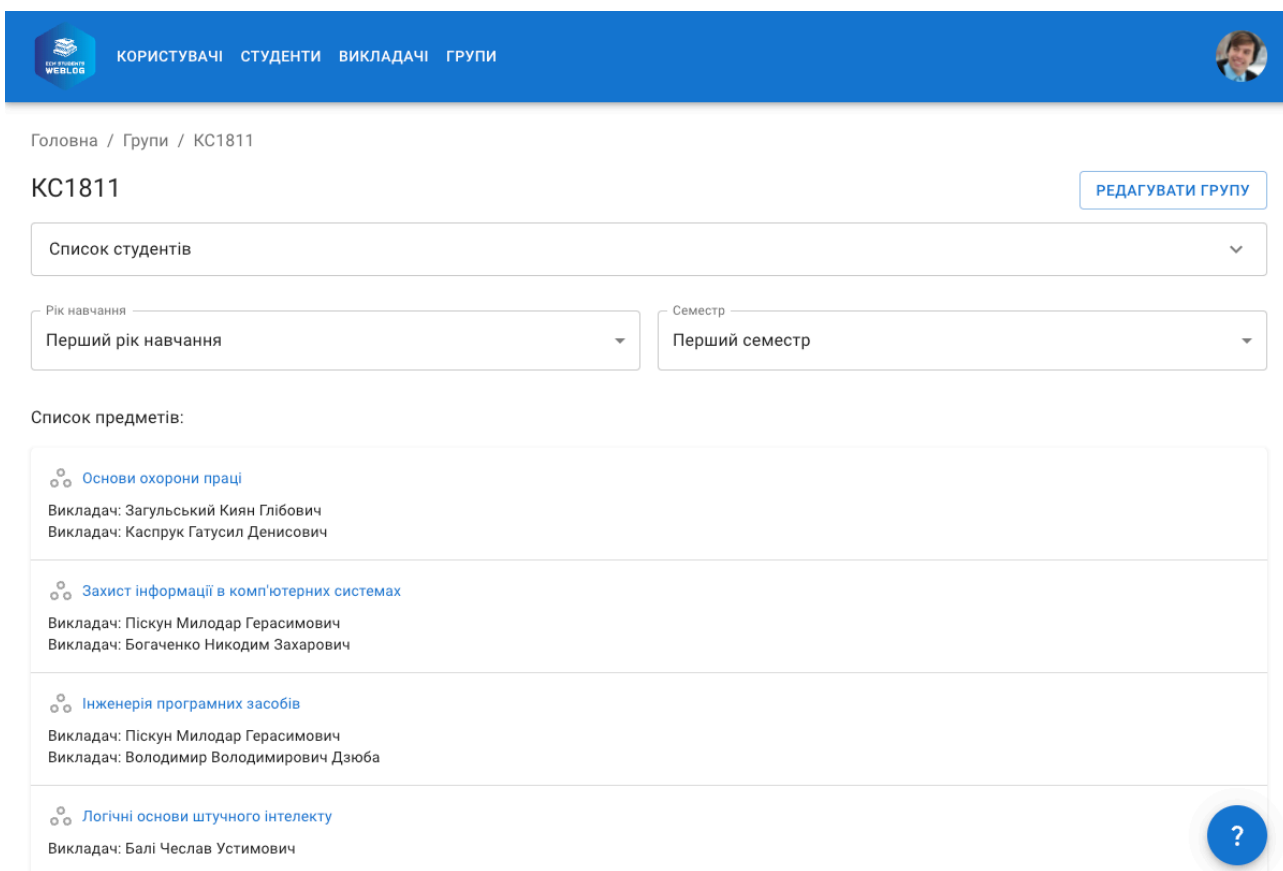
© ECM Students Weblog, 2022

Рисунок 3.10 – Сторінка управління групами

Для того щоб додати по групи предмети потрібно спочатку обрати рік навчання і потрібний семестр, адже предмети зазвичай при настанні нового року або семестру майже всі предмети змінюються. Такий підхід також допомагає застерегти викладачів від помилок, адже назви предметів можуть повторюватись впродовж декількох семестрів або навіть років. Запропоноване розділення не лише усуває плутанину, а ще й дозволить нам збирати статистику по роках навчання і семестрах, як для окремої групі так загальну для всієї системи. У випадку якщо потрібного року або семестру немає – у нас передбачений функціонал для їх створення.

Створення нового року навчання або семестру потребує зазначення їх назви (наприклад, «перший рік навчання» або «перший семестр») і дати початку та закінчення. Після його ми можемо перейти до наступного етапу.

Для управління предметами ми також маємо спеціально призначену сторінку (рисунок 3.11).



Користувачі | Студенти | Викладачі | Групи

Головна / Групи / КС1811

КС1811 РЕДАГУВАТИ ГРУПУ

Список студентів

Рік навчання: Перший рік навчання

Семестр: Перший семестр

Список предметів:

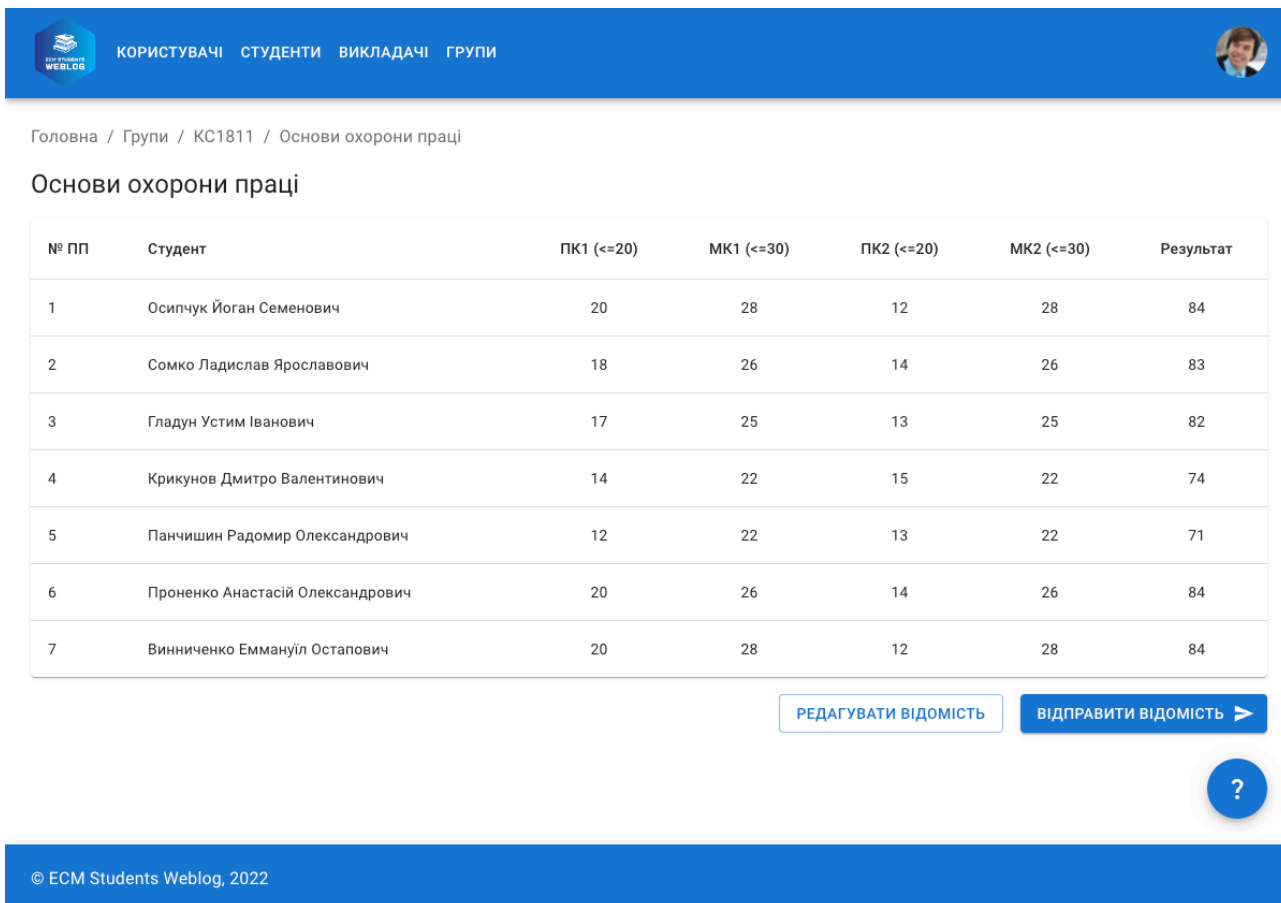
- Основи охорони праці  
Викладач: Загальський Киян Глібович  
Викладач: Каспрук Гатусил Денисович
- Захист інформації в комп'ютерних системах  
Викладач: Пісун Милодар Герасимович  
Викладач: Богаченко Никодим Захарович
- Інженерія програмних засобів  
Викладач: Пісун Милодар Герасимович  
Викладач: Володимир Володимирович Дзюба
- Логічні основи штучного інтелекту  
Викладач: Балі Чеслав Устимович

Рисунок 3.11 – Сторінка управління групою

На цій сторінці ми можемо редагувати або додавати до групи нові предмети. Кожен предмет має свої власні екзамени, наприклад «ПК1», «МК1», «ПК2» та «МК2». Кожен екзамен повинен мати максимальну оцінку і крайній строк здачі. При створенні предмету одним з обов'язкових полів також являється перелік викладачів, які ведуть даний предмет і мають доступ до заповнення відомостей.

### 3.1.5 Заповнення й відправка відомостей на навчальну кафедру

На відміну від усіх попередніх пунктів, доступ до яких мають лише адміністратори системи, заповнювати відомості можуть також і викладачі. На сторінці управління відомістю (рисунок 3.12) викладач може заповнити відомість, виставивши отримані оцінки усім членам групи.



Користувачі: СТУДЕНТИ ВИКЛАДАЧІ ГРУПИ

Головна / Групи / КС1811 / Основи охорони праці

Основи охорони праці

№ ПП	Студент	ПК1 (<=20)	МК1 (<=30)	ПК2 (<=20)	МК2 (<=30)	Результат
1	Осипчук Йоган Семенович	20	28	12	28	84
2	Сомко Ладислав Ярославович	18	26	14	26	83
3	Гладун Устим Іванович	17	25	13	25	82
4	Крикунов Дмитро Валентинович	14	22	15	22	74
5	Панчишин Радомир Олександрович	12	22	13	22	71
6	Проненко Анастасій Олександрович	20	26	14	26	84
7	Винниченко Еммануїл Остапович	20	28	12	28	84

РЕДАГУВАТИ ВІДОМІСТЬ    ВІДПРАВИТИ ВІДОМІСТЬ ►

© ECM Students Weblog, 2022

Рисунок 2.12 – Сторінка заповнення відомості

При виставленні оцінок передбачена валідація для уникнення помилок, наприклад, якщо максимальною оцінкою являється 30 балів, то викладач не може занести до відомості оцінку більше ніж 30 або менше за 18, бо за правилами студент не може отримати оцінку нижче 60% від максимальної. Після заповнення відомості викладач має можливість відправити її на кафедру за допомогою спеціальної кнопки. Після відправки відомості працівник кафедри отримує сповіщення на свою електронну пошту та має при першій можливості занести результат до системи оцінювання використовуваної на кафедрі. Після внесення результату потрібно зафіксувати це у нашій системі, адже час відправки та занесення чітко фіксується, щоб уникнути будь-яких непорозумінь або розбіжностей.

### **3.2 Висновки**

У цьому розділі було спроектовано та розроблено систему дистанційного обліку навчальної успішності, а саме її клієнтську та серверну частину.

## ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

В кваліфікаційній роботі виконано дослідження та розробку веб-системи дистанційного обліку навчальної успішності, а саме її клієнтську та серверну частини.

Ця система допоможе викладачам, які не можуть з'явитись на кафедрі у потрібний строк, вчасно передати інформацію щодо навчальної відомості.

У системі передбачені наступні функції: створення облікового запису та вхід до системи; реалізація управління доступом на основі ролей; створення профілів студентів та викладачів; створення навчальних груп та предметів; заповнення й відправка відомостей на навчальну кафедру.

В результаті виконання кваліфікаційної роботи було здійснено:

1. Проведено огляд сучасних систем обліку навчальної успішності;
2. Обрано технологічний стек для розробки;
3. Проведено опис функцій розроблюваної системи;
4. Розроблено програмне забезпечення клієнтської частини;
5. Розроблено програмне забезпечення серверної частини.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Огляд «Щоденник». [Електронний ресурс] // Державна наукова установа "Інститут освітньої аналітики". – 2020. – Режим доступу до ресурсу: <https://e-journal.iea.gov.ua/>.
2. Огляд «Additio App». [Електронний ресурс] // Additio. – 2014. – Режим доступу до ресурсу: <https://www.additioapp.com/>.
3. Огляд «TeacherKit». [Електронний ресурс] // TeacherKit. – 2022. – Режим доступу до ресурсу: <https://www.teacherkit.net/>.
4. Как правильно выбрать технологический стек для своего проекта [Електронний ресурс] // Oleg Melnic. – 2021. – Режим доступу до ресурсу: <https://tproger.ru/articles/kak-pravilno-vybrat-tehnologicheskij-stek-dlja-svoego-proekta/>.
5. Как правильно выбрать стек для своего проекта [Електронний ресурс] // Анна Яарова. – 2021. – Режим доступу до ресурсу: <https://vc.ru/u/957161-anna-yarova/316151-kak-pravilno-vybrat-stek-dlya-svoego-proekta>.
6. Документація React [Електронний ресурс] // Meta Platforms, Inc. – 2022. – Режим доступу до ресурсу: <https://reactjs.org/docs/getting-started.html>.
7. Документація React Query [Електронний ресурс] // Tanner Linsley. – 2020. – Режим доступу до ресурсу: <https://react-query.tanstack.com/overview>.
8. Документація Axios [Електронний ресурс] // Matt Zabriskie. – 2014-2022. – Режим доступу до ресурсу: <https://axios-http.com/docs/intro>.
9. Современный учебник JavaScript [Електронний ресурс] // Илья Кантор. – 2007—2022. – Режим доступу до ресурсу: <https://learn.javascript.ru/>.
10. Документація Node.js [Електронний ресурс] // OpenJS Foundation. – 2022. – Режим доступу до ресурсу: <https://nodejs.org/en/docs/>.
11. Документація Express.js [Електронний ресурс] // OpenJS Foundation. – 2022. – Режим доступу до ресурсу: <https://expressjs.com/>.

12. Документація MongoDB [Електронний ресурс] // MongoDB, Inc. – 2022. – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>.
13. What is the MERN Stack? [Електронний ресурс] // MongoDB, Inc. – 2022. – Режим доступу до ресурсу: <https://www.mongodb.com/mern-stack>.
14. JSON Web Token [Електронний ресурс] // Wikipedia. – 2022. – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/JSON\\_Web-Token](https://uk.wikipedia.org/wiki/JSON_Web-Token).
15. Керування доступом на основі ролей [Електронний ресурс] // Wikipedia. – 2022. – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/%D0%9A%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F\\_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC\\_%D0%BD%D0%B0\\_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%96\\_%D1%80%D0%BE%D0%BB%D0%B5%D0%B9](https://uk.wikipedia.org/wiki/%D0%9A%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC_%D0%BD%D0%B0_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%96_%D1%80%D0%BE%D0%BB%D0%B5%D0%B9).