

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Курсова робота

з дисципліни «Програмування»

на тему: «Електронний журнал успішності студентів»

Виконав:

студент 1 курсу, групи ІА-33

Клімчук Володимир Ігорович

(підпис)

Керівник:

асистент кафедри ІСТ

Мягкий Михайло Юрійович

(підпис)

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

Студент

(підпис)

Зміст

ВСТУП	2
1 ВИМОГИ ДО СИСТЕМИ	4
1.1 Функціональні вимоги до системи	5
1.2 Нефункціональні вимоги до системи	5
2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ	6
2.1 Діаграма прецедентів	6
2.2 Опис сценаріїв використання системи	7
3 АРХІТЕКТУРА СИСТЕМИ	12
4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	14
4.1 Загальна структура проекту	15
4.2 Компоненти рівня доступу до даних	16
4.3 Компоненти рівня бізнес-логіки	16
4.4 Компоненти рівня інтерфейсу користувача	17
ВИСНОВОК	18
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	

ВСТУП

Це потужний інструмент сучасної освіти та дуже корисний електронний журнал (е-журнал) для студентів та викладачів. Він має багато переваг, серед яких зручність, ефективність та покращення комунікації. У цій статті ми розглянемо, чому створення електронного журналу є важливим і корисним для студентів та викладачів. Електронний журнал дає можливість викладачам та студентам мати постійний доступ до необхідної інформації. Вони можуть легко перевірити свої оцінки, завдання, розклад уроків та іншу важливу інформацію з будь-якого пристрою, підключеного до Інтернету. Це зменшує потребу в паперових записах, які можуть бути втрачені або пошкоджені. Мобільність є ключовим аспектом електронного журналу. Учні можуть вносити записи до журналу будь-де і будь-коли. Це особливо вигідно тим учням, які можуть бути відсутніми в класі з різних причин, але все одно хочуть бути в курсі своїх академічних справ. Це також дозволяє викладачам мати більше часу для викладання та взаємодії зі студентами, а не для виконання адміністративних обов'язків". Автоматизація також зменшує ймовірність людських помилок під час введення даних ". Організація та структурування даних в електронному журналі робить їх доступними та зручними для аналізу". Викладачі можуть швидко отримати необхідну інформацію про прогрес кожного студента, що допомагає вчасно виявити проблеми та надати необхідну підтримку. За допомогою електронного журналу студенти можуть швидко та зручно ставити запитання викладачам, отримувати відповіді або коментарі щодо своєї роботи. Це сприяє кращому засвоєнню матеріалу та дозволяє студентам швидше виправлятися. Використання електронних журналів викладачами допомагає їм краще розуміти сильні та слабкі сторони кожного студента, що дозволяє частіше застосовувати індивідуальний підхід у навчанні. Це особливо корисно для студентів з особливими потребами, які потребують особливого підходу та підтримки. Моніторинг прогресу студентів за допомогою електронного журналу дозволяє

тьюторам вчасно виявити слабкі місця та скоригувати підходи до викладання. Це передбачає оперативне реагування на зміни в академічній успішності, а також надання необхідної допомоги та рекомендацій. Адміністративне навантаження зменшується завдяки електронному журналу, що дозволяє приділяти більше часу основним обов'язкам, таким як репетиторство, а не виконанню рутинних адміністративних завдань, які не мають безпосередньої цінності для студентів, таких як викладання або підготовка їх до майбутніх випробувань, замість виконання адміністративної рутини. Електронні журнали забезпечують високий рівень захисту даних. Електронний журнал дозволяє контролювати доступ до інформації. Вчителі можуть визначати, хто має доступ до тих чи інших даних, що забезпечує конфіденційність та безпеку персональної інформації студентів. Це особливо важливо для захисту особистих даних та дотримання законодавства про захист даних. Інші освітні системи, такі як системи управління навчанням (LMS), платформи електронного навчання та інші інструменти, можуть бути інтегровані з електронним журналом. Це дає можливість створити єдину екосистему, де всі необхідні ресурси та інструменти доступні в одному місці. Електронний журнал є важливим інструментом для підтримки дистанційного навчання. Через пандемію COVID-19 багато шкіл та університетів перейшли на онлайн викладання та навчання. Журнал дозволяє забезпечити безперервний навчальний процес, який підтримує взаємодію між студентами та викладачами незалежно від їхнього географічного розташування. Створення електронного журналу для студентів має численні переваги, які значно покращують навчальний процес. Від зручності та доступності до підвищення ефективності, кращої комунікації, персоналізації навчального процесу, економії ресурсів та захисту даних - електронний журнал є незамінним інструментом у сучасній освіті. Він сприяє розвитку інновацій, а також підтримує дистанційне навчання, що робить його життєво важливим елементом майбутньої освітньої системи.

1 ВИМОГИ ДО СИСТЕМИ

1.1 Функціональні вимоги до системи

Система має відповідати наступним функціональним вимогам:

- незареєстрований користувач не може бути допущений до веб-інтерфесу
- зареєстрований студент повинен мати можливість переглядати журнал
- зареєстрований викладач може ставити відмітки зареєстрованим студентам
- адміністратор сайту має необмежений функціонал

1.2 Нефункціональні вимоги до системи

Система має відповідати наступним нефункціональним вимогам:

- система повинна мати відкриту архітектуру;
- система повинна мати веб-інтерфейс;
- інтерфейс користувача має бути зручним та інтуїтивно-зрозумілим;
- система повинна бути крос-платформною.

2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Діаграма прецедентів зображена на рисунку 2.1. Актором є користувач, який одночасно може бути як клієнтом, так і хазяїном. Нижче, в пункті 2.2, будуть наведені всі використовувані сценарії.

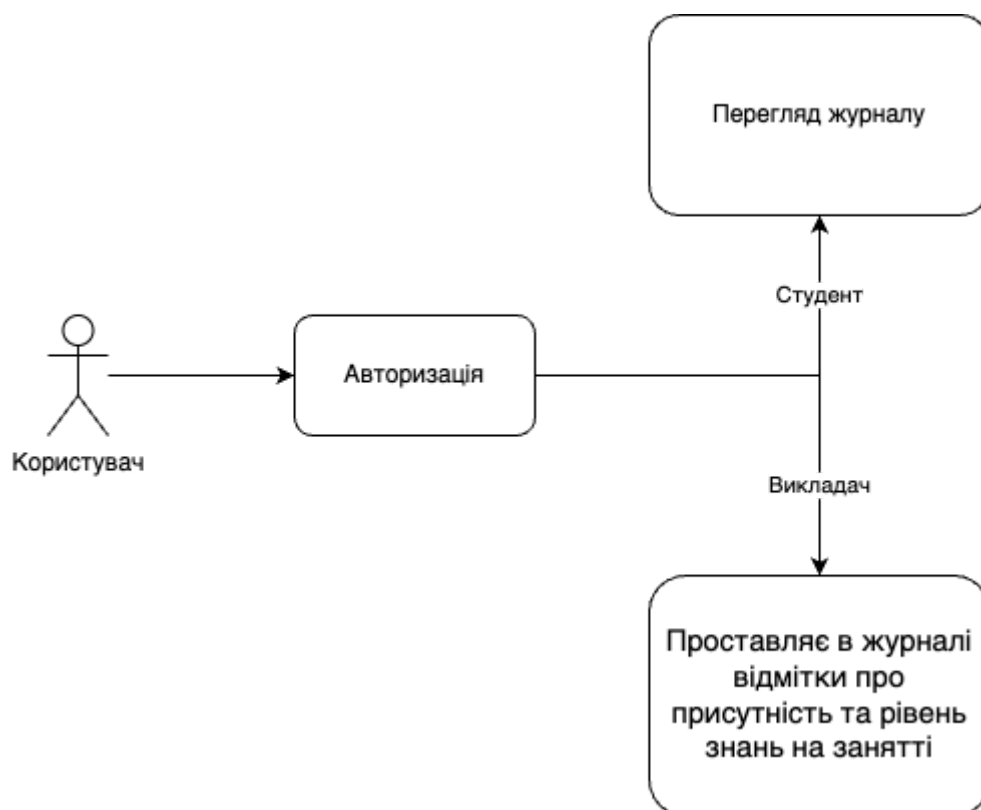


Рисунок 2.1 – Діаграма прецедентів

2.2 Опис сценаріїв використання системи

Детальні описи сценаріїв використання наведено у таблицях 2.1 – 2.5.

Таблиця 2.1 – Сценарій використання «Переглянути журнал оцінок»

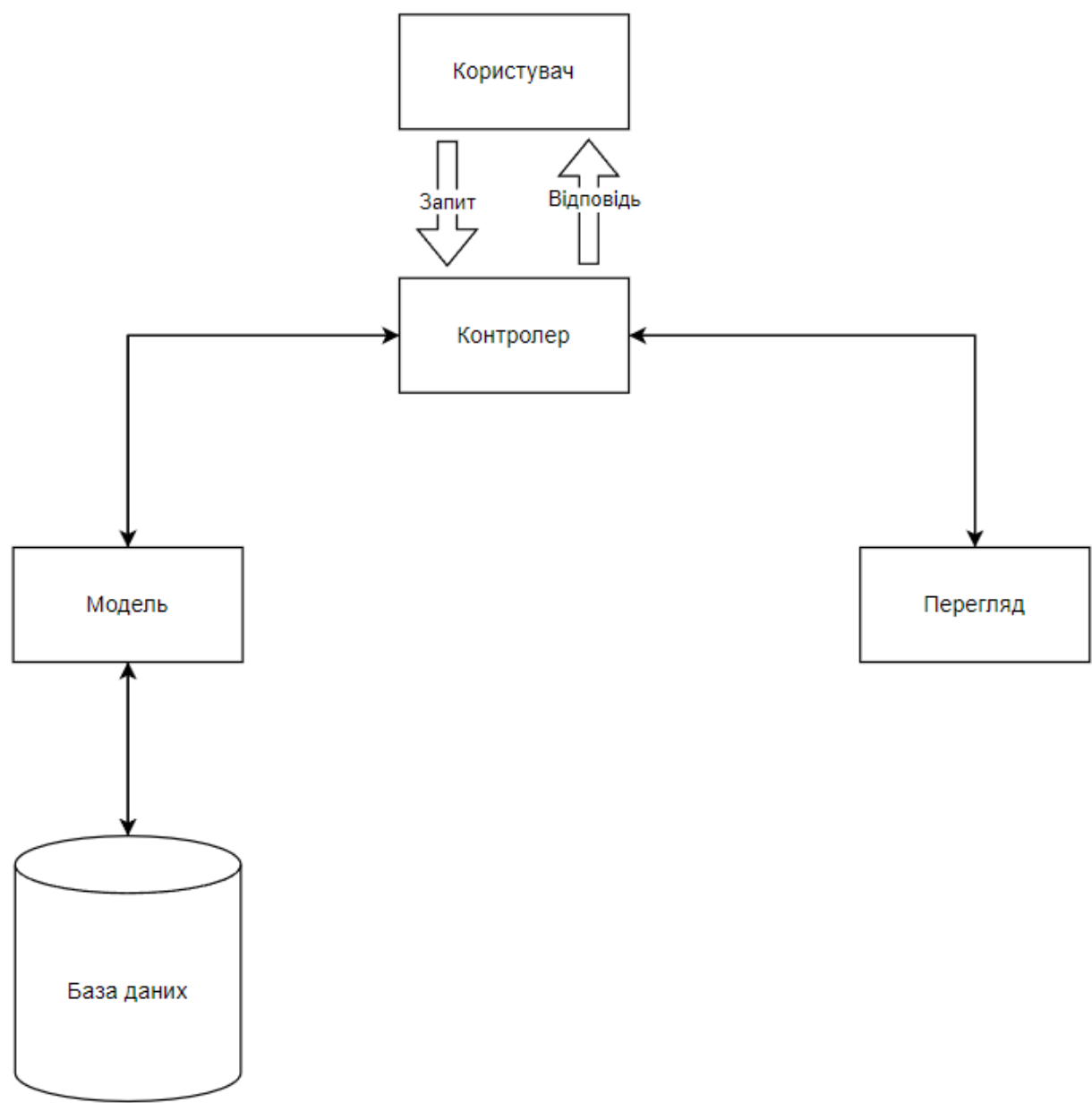
Назва	Переглянути журнал оцінок
ID	1
Опис	Студент натискає на кнопку “Журнал”
Актори	Студент
Вигоди компанії	Без можливості переглядати журнал, проект втрачає свій сенс
Частота користування	Постійно
Тригери	Студент переходить на відповідну сторінку
Передумови	Посилання на сторінку зі списком всіх оцінок
Постумови	Студент потрапляє на сторінку зі списком своїх оцінок
Основний розвиток	Студент натискає на посилання на сторінку зі всіма оцінками, відбувається перехід на сторінку з квартирами, де студент може бачити список своїх оцінок
Альтернативні розвитку	–
Виняткові ситуації	–

Таблиця 2.2 – Сценарій використання «Виставлення оцінки викладачем»

Назва	Виставлення оцінки викладачем
ID	2
Опис	Викладач вибирає студента з випадającego списку, вибирає предмет зі списку і вписує значення оцінки
Актори	Викладач
Вигоди компанії	Можливість виставляти оцінки пряма задача електронного журналу
Частота користування	Постійно
Тригери	Викладач виставляє оцінку
Передумови	На сторінці виставлення оцінки маємо два випадających списки в яких можна вибрати студента і предмет, після чого ввести значення оцінки і відправити в базу даних
Постумови	Викладач отримує повідомлення просто виставлення оцінки
Основний розвиток	Викладач натискає на посилання на сторінку вибирає студента з випадającego списку, вибирає предмет зі списку і вписує значення оцінки після чого дані відправляють в базу даних
Альтернативні розвитку	—
Виняткові ситуації	—

3 АРХІТЕКТУРА СИСТЕМИ

Загальна архітектура системи наведена на рис. 3.1



Таблиця 3.1 – Загальна архітектура системи

Система складається з наступних елементів:

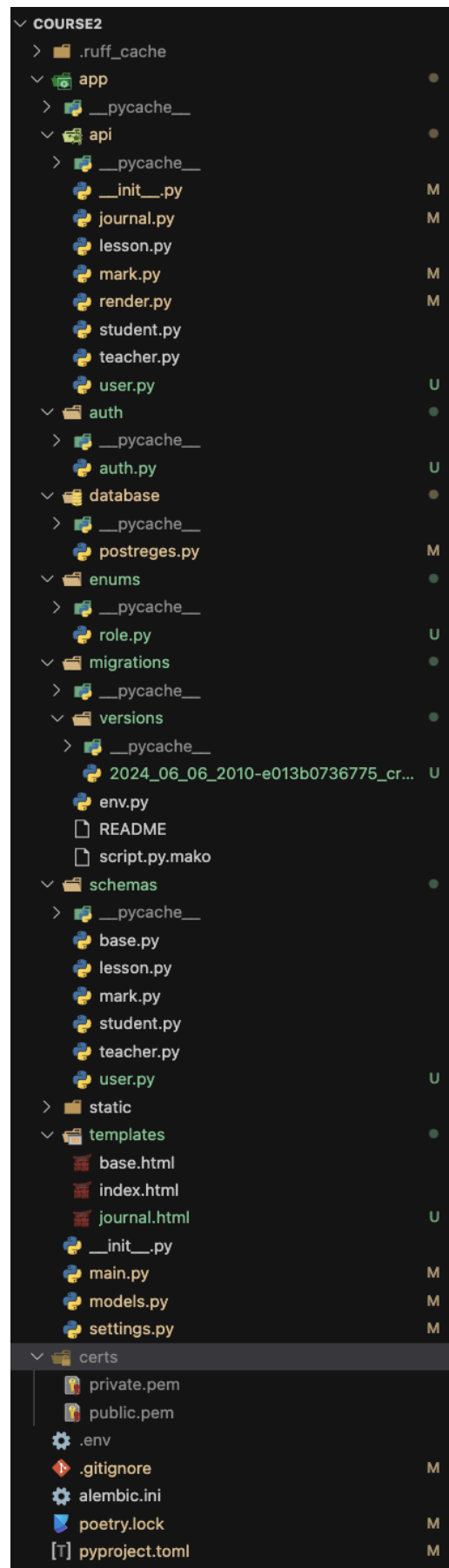
- графічний інтерфейс;
- серверна частина;
- база даних.

Графічний інтерфейс необхідний для взаємодії з користувачем. HTTP запит надходить до серверної частини, де оброблюється і повертається відповідь. На серверній частині виконується основна логіка системи. Дані, отриманні з графічного інтерфейсу валідуються, конвертуються. Також, серверна частина формує запит до бази даних та оброблює відповідь і передає її до графічного інтерфейсу. База даних зберігає дані, які були сформовані на серверній частині та повертає їх у разі запиту.

До серверної частини належать наступні елементи:

- контролер;
- модель та вигляд;
- сервіс;
- репозиторій.

На контролер надходять дані з графічного інтерфейсу. З контролеру, дані формуються в сервісі для запиту в репозиторій. З репозиторію дані надсилаються до бази даних і зберігаються. Також в контролері формується вид, тобто об'єкт і його ім'я для відображення на графічному інтерфейсі.



4.1 Загальна структура проекту

Рисунок 4.1 – Загальна структура проекту

Проект складається з веб-ресурсів, бібліотек та вихідного коду, який в свою чергу можна поділити на компоненти рівня доступу до даних, компоненти бізнес-логіки та веб-компоненти

Веб-ресурси включають HTML файли, які забезпечують структуру веб-сторінок (base.html, index.html, journal.html), CSS файли, які забезпечують стилізацію веб-сторінок (styles.css)

Бібліотеки включають сторонні Python бібліотеки та фреймворки, необхідні для роботи проекту.

ВИСНОВОК

Ми оглянули багато ресурсів під час виконання цієї роботи, щоб зрозуміти що і навіщо є в інтернет-проектах для оренди квартир. Багато сучасних підходів до оренди квартир ми проаналізували.

Спочатку ми переглядали техзавдання і сформулювали функціональні та нефункціональні вимоги до системи. Потім ми обрали технології та мову програмування. На роботу сервера нами було обрано мову Python, точніше кажучи фреймворк FastAPI. Ми обрали цей фреймворк оскільки нам пропонуються можливості створювати необхідний для нас функціонал швидко, тестувати його, змінювати майже миттєво. Для самої структури графічного інтерфейсу і його зв'язок із сервером ми використовували HTML, CSS і JavaScript. У ролі бази даних використовували PostgreSQL у зв'язку зі зручністю та можливістю інтуїтивного користування користувацьким інтерфейсом. Спочатку планувалося використання бази даних Redis, однак ця база даних має численні обмеження, як-от: відсутність структури, обмеженість масштабованості та необхідність грошей для великих обсягів даних. Ось тому ми вибрали PostgreSQL, оскільки у нього немає подібних мінусів.

Після всього перерахованого вище ми описали сценарії використання нашої системи. Саме завдяки цьому сценарію ми успішно і безпомилково реалізували потрібний нам продукт.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація FastAPI: <https://fastapi.tiangolo.com/>
2. Документація Python: <https://docs.python.org/3/>
3. Документація Python українською:
<https://github.com/python/python-docs-uk>
4. Документація Jinja2Templates:
<https://fastapi.tiangolo.com/advanced/templates/>

ДОДАТОК А Лістинг програми

<https://github.com/kelwi/course>