Функціональні вимоги

1. Створення та редагування проєктів - користувач може створити новий проєкт, вказавши назву, опис та дедлайн, а також редагувати інформацію про нього до завершення роботи.
2. Формування команд та розподіл ролей - лідер команди має можливість додавати учасників до проєкту та призначати їм ролі, а також змінювати ролі в процесі роботи.
3. Створення та управління завданнями - система дозволяє створювати завдання в межах проєкту, призначати відповідальних за їх виконання та змінювати статус завдань.
4. Відстеження прогресу проєкту - система автоматично відображає прогрес виконання завдань у проєкті, наприклад у вигляді списку або графічної діаграми.
5. Оцінювання проєкту викладачем - викладач може переглядати хід виконання проєкту та внесок кожного учасника, додавати коментарі та виставляти підсумкову оцінку.

Нефункціональні вимоги

1. Безпека даних - доступ до проєкту та внутрішніх даних має бути обмежений авторизованим користувачам. Паролі повинні зберігатися у зашифрованому вигляді, передача даних через HTTPS.
2. Зручність та інтерфейс - інтерфейс має бути інтуїтивно зрозумілим і адаптивним, забезпечувати комфортне використання на комп’ютерах і мобільних пристроях.
3. Надійність та збереження даних - дані про проєкти, завдання та оцінки повинні зберігатися у базі даних і залишатися доступними навіть після перезапуску системи або оновлення серверної частини.

Обґрунтування вибору моделі життєвого циклу

Ітераційну модель розробки доцільно обрати тому, що система має функціонал, який зручно впроваджувати поступово: спочатку створення та редагування проєктів, потім розподіл ролей, управління завданнями та, на завершальному етапі, оцінювання викладачем. Вимоги можуть змінюватися або уточнюватися в процесі роботи, тому важливо мати можливість повертатися до вже реалізованих частин та вдосконалювати їх. Ітераційна модель дозволяє швидко отримати проміжну робочу версію системи, продемонструвати її, отримати зворотний зв’язок та поліпшити функціонал. Вона забезпечує гнучкість і адаптивність, що є важливими для навчальних проектів та систем, які розвиваються за потребами користувачів.

Проєктування архітектури

Модулі

1. Користувачі (User Management)

* Реєстрація та автентифікація користувачів
* Зберігання профілів
* Авторизація доступу до функцій системи

1. Проєкти (Project Management)

* Створення, редагування та видалення проєктів
* Додавання учасників і призначення лідера
* Управління термінами виконання

1. Ролі та команди (Team & Role Management)

* Призначення ролей
* Зміна ролей у процесі роботи
* Перевірка прав доступу до окремих функцій

1. Завдання (Task Management)

* Створення завдань у межах проєкту
* Призначення відповідальних осіб
* Зміна статусів завдань
* Автоматичне обчислення прогресу

1. Оцінювання (Evaluation Management)

* Викладач переглядає стан проєкту
* Додає коментарі, виставляє оцінку
* Формує підсумковий звіт по команді

1. Моніторинг та історія (History & Logging)

* Зберігання історії змін у проєкті
* Відстеження дій користувачів
* Формування журналів активності для викладача

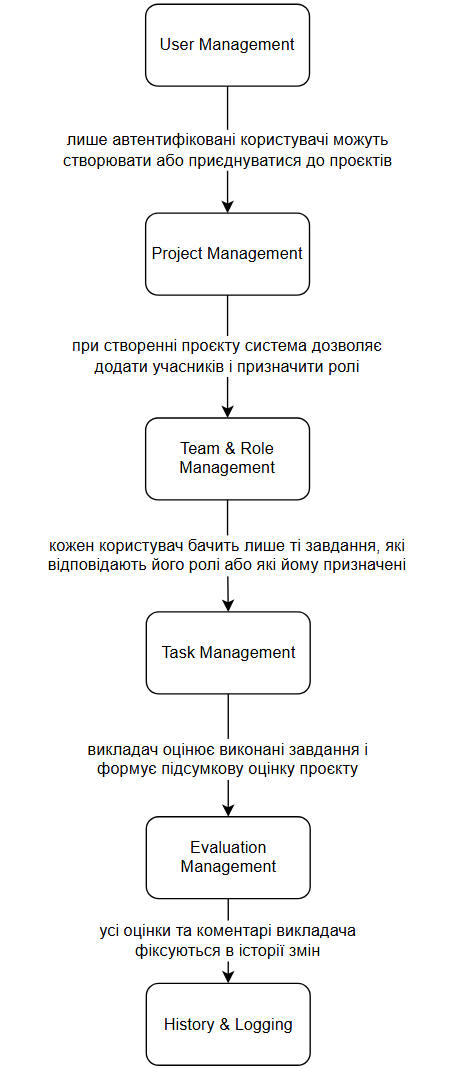
1. Безпека (Security & Privacy)

* Захист даних користувачів
* Авторизація доступу за токенами
* Шифрування паролів і з’єднань

Інтерфейси між модулями

1. User Management → Project Management - лише автентифіковані користувачі можуть створювати або приєднуватися до проєктів.
2. Project Management ↔ Team & Role Management - при створенні проєкту система дозволяє додати учасників і призначити ролі.
3. Team & Role Management → Task Management - кожен користувач бачить лише ті завдання, які відповідають його ролі або які йому призначені.
4. Task Management ↔ Evaluation Management - викладач оцінює виконані завдання і формує підсумкову оцінку проєкту.
5. Evaluation Management ↔ History & Logging - усі оцінки та коментарі викладача фіксуються в історії змін.

Схема



Вибір архітектурного підходу

Для даної системи доцільно використати монолітну архітектуру, оскільки всі її основні модулі тісно пов’язані між собою та активно взаємодіють один з одним. Такий підхід робить розробку простішою і швидшою, оскільки вся логіка розміщується в одному застосунку. Це спрощує процес створення системи і її тестування, розгортання та подальше обслуговування. Система призначена для використання в рамках навчального закладу і не потребує високої масштабованості, яку забезпечують мікросервіси.