AI Learning Tracker

Система інтелектуального планування та рефлексивного контролю навчальних цілей на основі генеративних мовних моделей

Державний університет інформаційно-комунікаційних технологій
Кафедра Комп'ютерних наук
Спеціальність F3 Комп'ютерні науки



Проблематика

Проблематика персоналізованого планування у системах самоосвіти

Актуальністю роботи є загострення проблеми когнітивної дезорієнтації та швидкого вигорання студентів під час неформального самостійного навчання.

Попри доступність великих навчальних ресурсів, більшість студентів стикається з так званою “ілюзією планування” – встановлення занадто оптимістичних дедлайнів без розрахунку власних фізичних сил і часу.

Наявні програмні інструменти фіксують лише формальні позначки виконання задач, проте не здатні оцінити реалістичність графіка або проаналізувати змістовні нотатки рефлексії про труднощі, з якими стикається людина.

Мета

В першу чергу метою дослідження стає підвищення систематичності та результативності індивідуальної самоосвіти шляхом проєктування, розробки й дослідження програмного комплексу “AI Learning Tracker”, який поєднує в собі як інтелектуальну декомпозицію навчальних планів моделлю Google Gemini із алгоритмічною верифікацією здійсненності графіку так і адаптивним аналізом рефлексивних журналів та інтерактивним Single Page Application веб-інтерфейсом.

Існуючі рішення та обмеження LLM

Порівняльний аналіз існуючих рішень та обмеження LLM

Класичні таск-менеджери (Trello, Todoist, Notion, Asana) базуються на списках завдань, канбан-дошках та таблицях. Їхні переваги полягають у надійному зберіганні даних, швидкій синхронізації та системі нагадувань.
Проте ці інструменти залишаються пасивними реєстраторами дій. Вони не здатні декомпонувати мету, вибудувати дидактичну послідовність або спрогнозувати час на засвоєння тем.

Діалогові мовні моделі (OpenAI GPT-4o, Anthropic Claude, Google Gemini Web) дозволяють синтезувати навчальні плани за текстовим запитом завдяки широкому контексту та гнучкій адаптації матеріалу.
Разом із тим їх пряме застосування має критичні обмеження:

- стохастичність виводу змішує структуру даних із розмовним текстом;
- відсутність внутрішнього обліку фізичного часу спричиняє галюцинації щодо календарних термінів;
- робота в режимі без збереження стану (Stateless) не дає змоги вести довготривалий облік завдань.

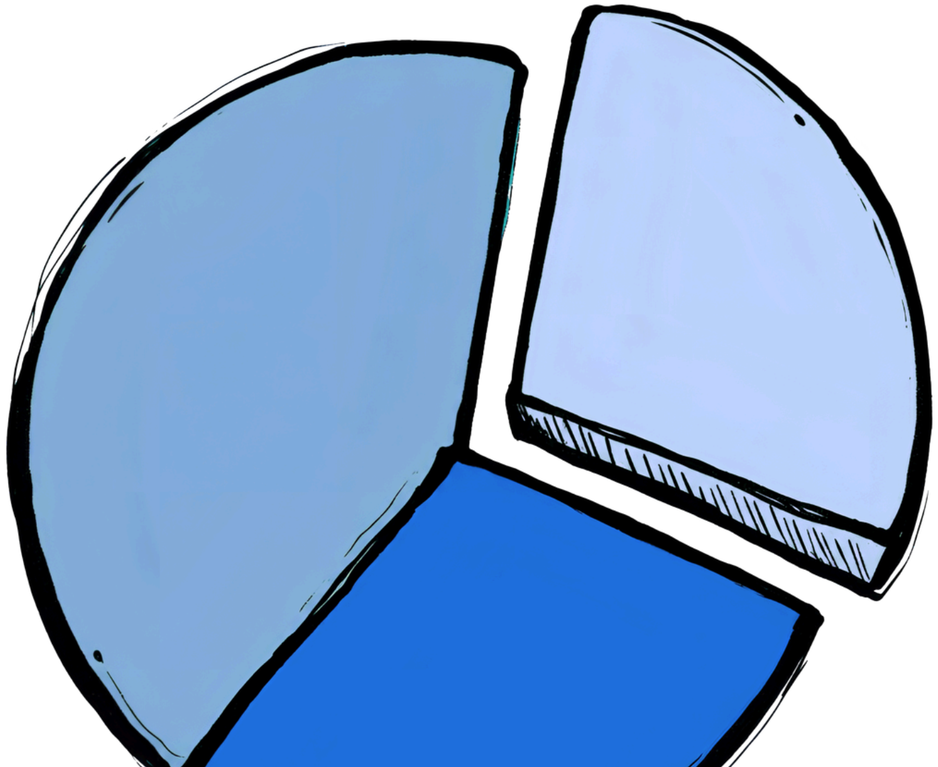
Двоетапний метод

Двоетапний метод контекстної декомпозиції завдань

Для подолання цих явищ запропоновано та реалізовано двоетапний метод контекстної декомпозиції цілей.

Його суть полягає у фізичному розмежуванні операцій концептуального макросинтезу та покрокової мікродекомпозиції на рівні послідовних синхронних звернень до моделі Google Gemini через інтерфейси фреймворку Spring AI.

Фаза 1. Макросегментація мети (Macro-Goal Synthesis).
Фаза 2. Мікродекомпозиція етапів (Subgoal Decomposition).



Макросегментація та мікродекомпозиція

Фаза 1. Макросегментація мети (Macro-Goal Synthesis)

Системний промпт забороняє генерацію довільного діалогу та вимагає провести семантичну нормалізацію тексту: очистити запит від суб'єктивних мовних конструкцій, виділити коротку канонічну назву мети (generalGoal) та класифікувати її за одним із трьох дискретних рівнів складності.
Рівень EASY призначається для локальних навичок із чіткими межами; MEDIUM – для прикладних інженерних технологій середньої трудомісткості; HARD – для комплексних концептуальних стеків та розподілених архітектур.

Фаза 2. Мікродекомпозиція етапів (Subgoal Decomposition)

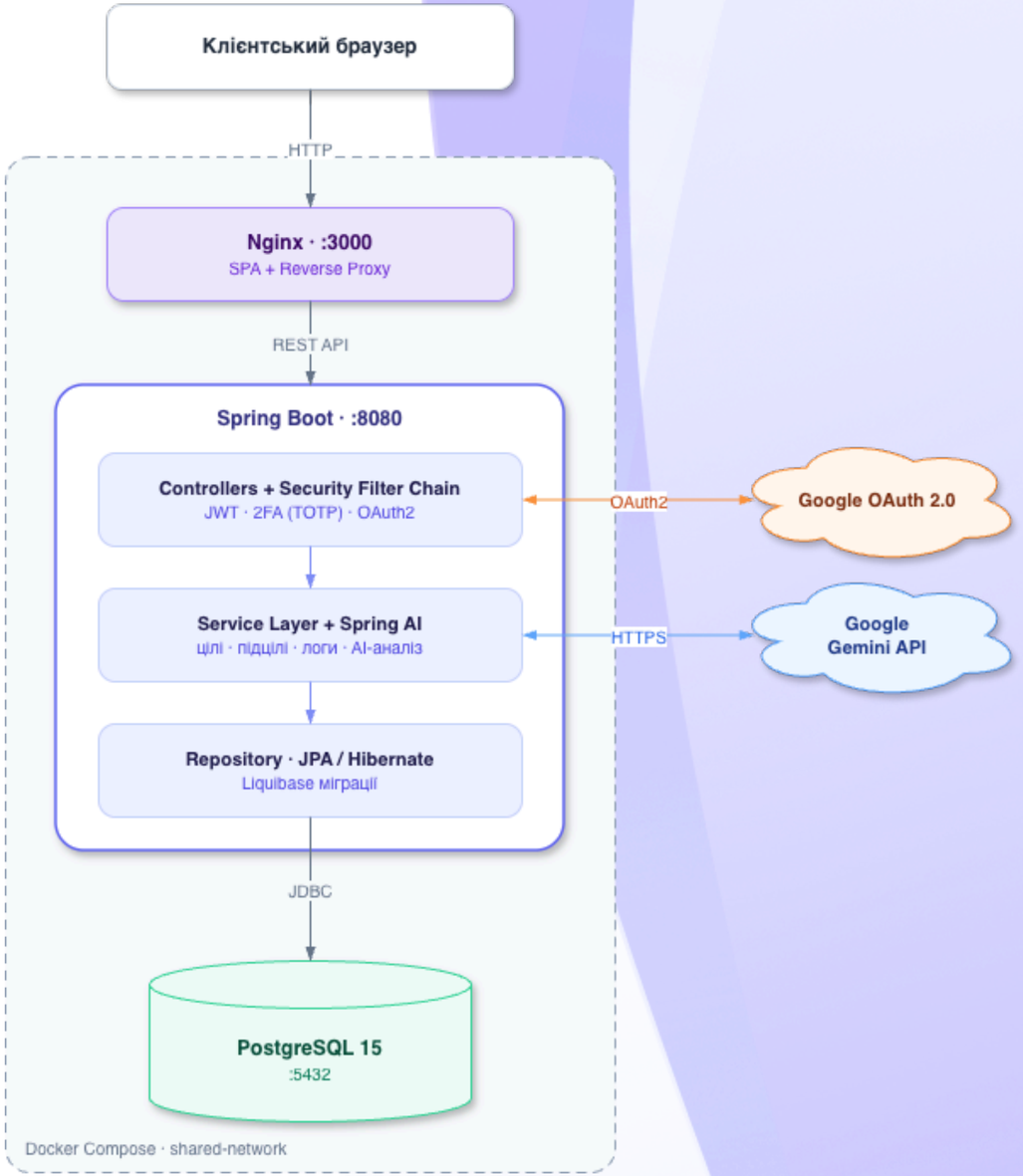
Конвеєр формує фіксований перелік із 3-5 послідовних підцілей, що відповідає оптимальному фокусу операційної уваги здобувача, деталізує методичний опис завдань для кожного етапу та призначає індивідуальну орієнтовну трудомісткість в академічних годинах (estimatedHours).

Архітектура системи

Архітектурний патерн та організація клієнт-серверної взаємодії

Комплекс спроєктовано за клієнт-серверною архітектурою Controller-Service-Repository (CSR) з елементами Domain-Driven Design (DDD) у безстатусному режимі REST API.
Система структурована за п'ятьма функціональними рівнями:

- Клієнтське представлення (Client SPA)
- Веб-контролери та маршрутизація (Presentation Layer)
- Сервісна бізнес-логіка (Business Service Layer)
- Подійна підсистема (Event-Driven Domain Subsystem)
- Персистентність та збереження даних (Data Access Layer)



Контроль здійсненності графіку

Для запобігання «ілюзії планування» компонент AIServiceImpl реалізує алгоритмічну перевірку часових обмежень як обов'язковий транзакційний шлюз перед збереженням цілі в базу даних.

Алгоритм оперує трьома послідовними етапами валідації:

1. Евристична нормалізація трудомісткості.
2. Розрахунок доступного фонду часу.
3. Контроль системного інваріанту здійсненності.

Система зіставляє наявний фонд часу із сукупною плановою трудомісткістю.

У разі дефіциту часу транзакція блокується виключенням GoalValidationException, а обробник повертає клієнту точний розрахунок нестачі годин із пропозицією відтермінувати дедлайн або збільшити темп занять.

Захисний конвеєр LLM та комплексна підсистема безпеки

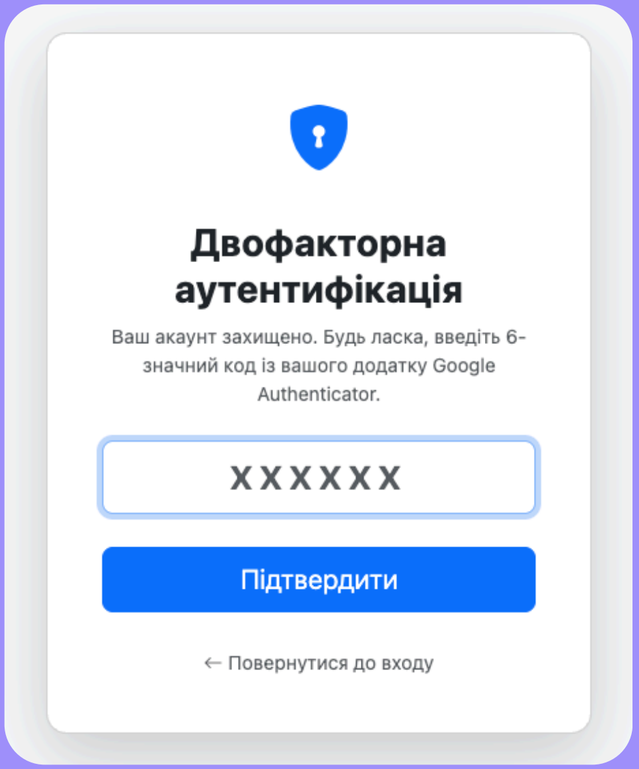
Захисний конвеєр обробки та типізованого парсингу виводу LLM

Стохастичні відповіді великих мовних моделей часто містять супровідні артефакти: розмітку Markdown (```json), пробіли чи коментарі, передача яких у стандартний парсер викликає збої JsonParseException. Для захисту бізнес-контуру в методі AIServiceImpl.getString() розроблено конвеєр санітизації: алгоритм визначає індекси першої дужки { та останньої }, ізолює корисне навантаження й трансформує підрядок через Jackson ObjectMapper у строго типізовані Java DTO.

Комплексна підсистема безпеки

Підсистема безпеки забезпечує захист веб-сесій, криптографічне підписування токенів та багатофакторний доступ:

- Двотокенова модель JWT (HMAC-SHA256): короткоживучий Access-токен (2 години) валідується фільтром AccessTokenAuthenticationFilter на кожному HTTP-запиті, а довгоживучий Refresh-токен (10 годин) забезпечує безпечну пролонгацію сесії.
- Двофакторна автентифікація (RFC 6238 TOTP): реалізовано інтеграцію з Google Authenticator.
- Гібридна інтеграція Google OAuth2: підтримує безпечний вхід із автоматичною ініціалізацією облікового запису в БД.



Експериментальні дослідження

Тип операції / Кінцева точка	Вхідні параметри	Середній час (мс)	Мін. / Макс. час (мс)	Статус обробки
Двоетапна декомпозиція мети /goal/create	3 підцілі	**1140**	890 / 1420	Успішно
Двоетапна декомпозиція мети /goal/create	5 підцілей	**1380**	1020 / 1790	Успішно
Синтез рекомендацій рефлексії /aiAnalysis/...	Вибірка 10 логів	**920**	740 / 1210	Успішно
Автентифікація та валідація TOTP /auth/2fa/verify	6-значний код	**18**	12 / 26	Успішно

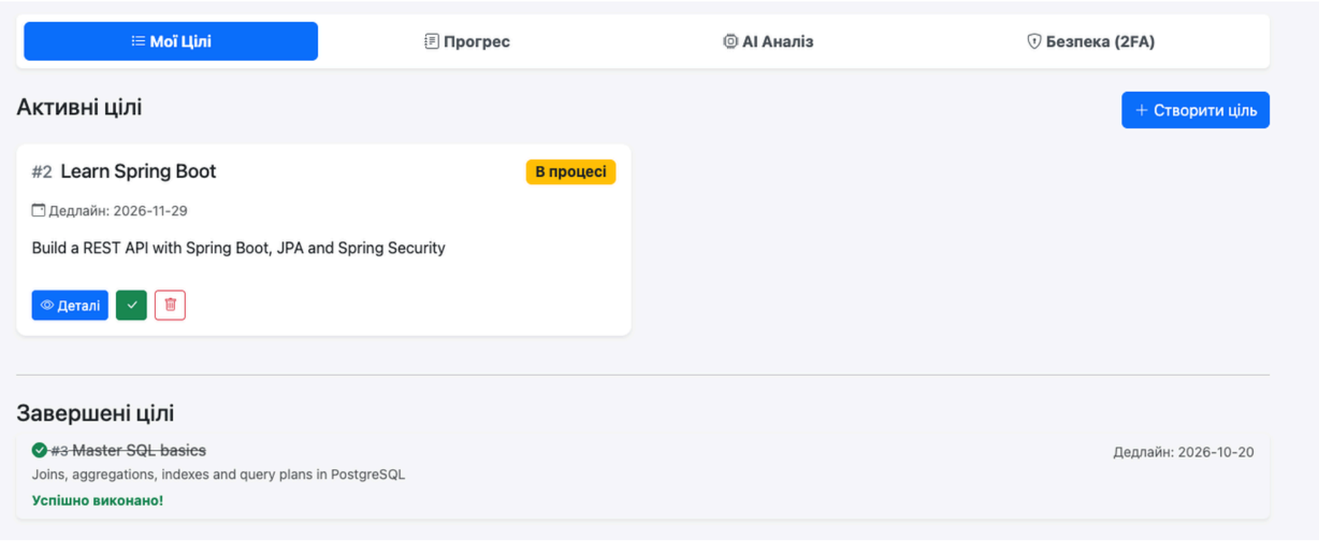
Результати тестування

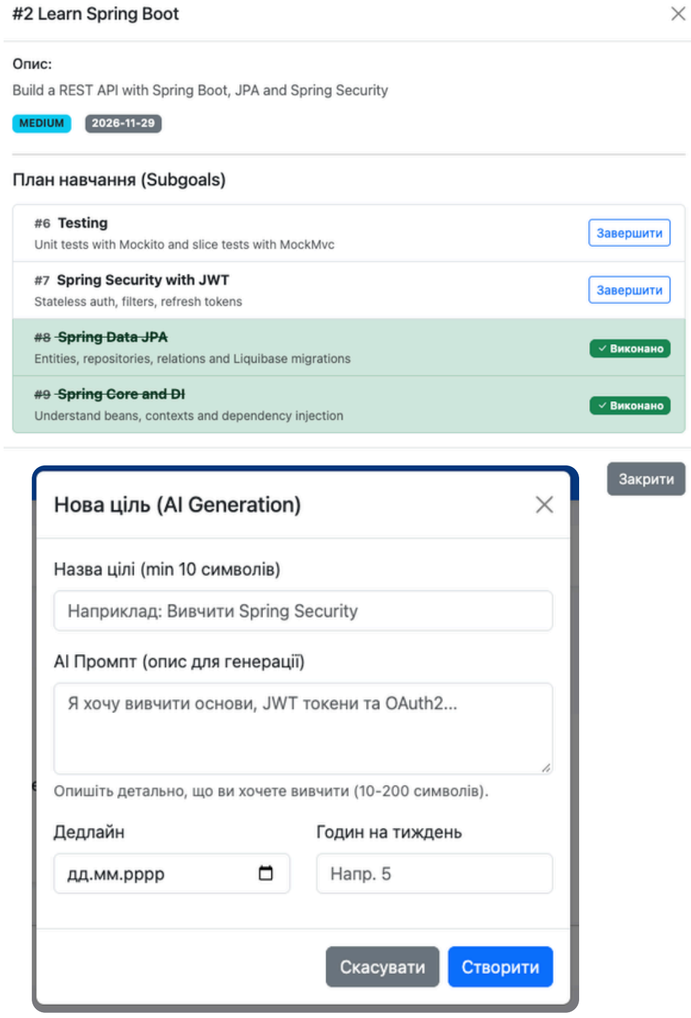
Дослідження стійкості алгоритму верифікації дедлайнів

Перевірку алгоритму здійсненності проведено на вибірці зі 150 контрольних сценаріїв із варіюванням складності курсів, тижневого бюджету (2–20 год/тиждень) та інтервалу дедлайну (від 3 днів до 2 місяців).

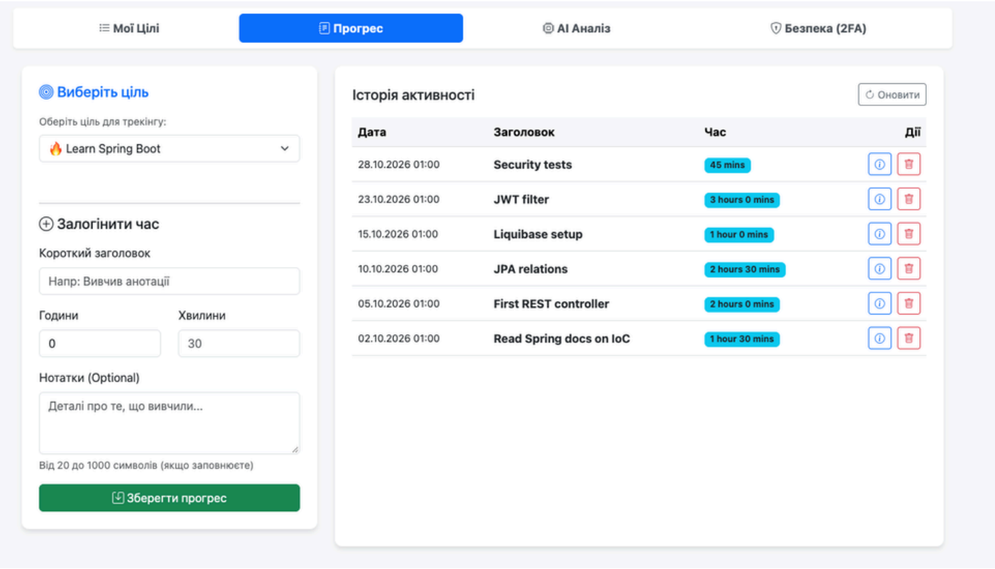
Алгоритм продемонстрував 100% точність відсікання нереалістичних планів: у 78 випадках дефіциту часу створення було заблоковане з генерацією GoalValidationException та обчисленням точного браку годин, а у 72 сценаріях плани успішно збережено в базі даних без хибних відхилень.

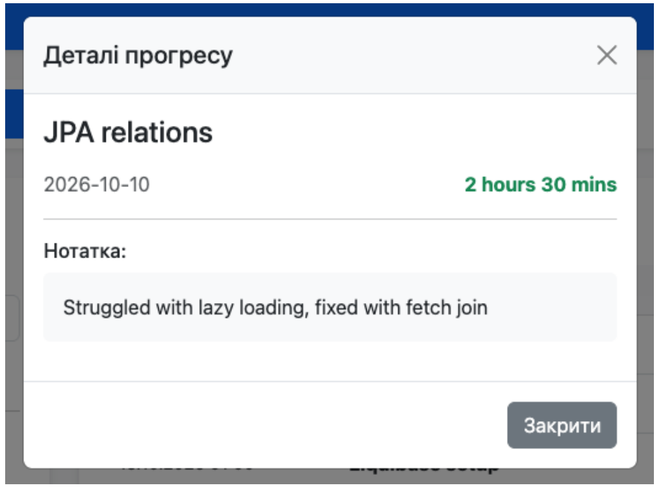
Загальна робота застосунку



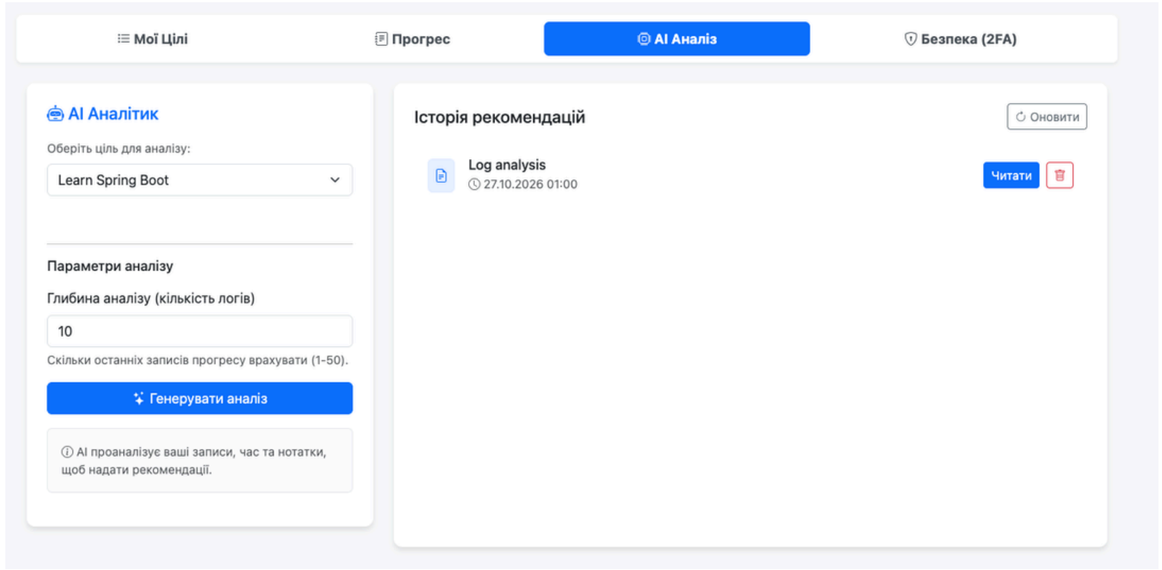


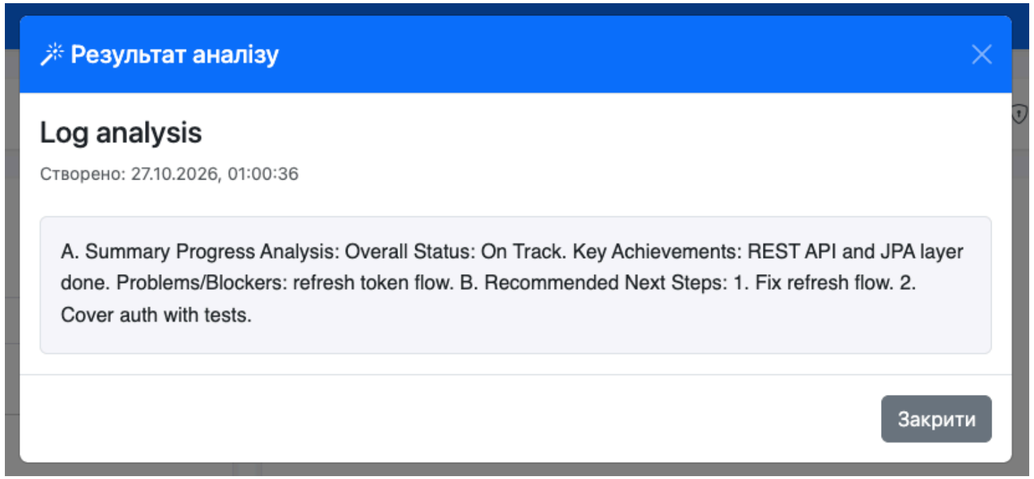
Прогрес


Мої Цілі
Прогрес
AI Аналіз
Безпека (2FA)
Виберіть ціль
Оберіть ціль для трекінгу:
Learn Spring Boot
Залогінити час
Короткий заголовок
Напр: Вивчив анотації
Години
Хвилини
0
30
Нотатки (Optional)
Деталі про те, що вивчили...
Від 20 до 1000 символів (якщо заповнюєте)
Зберегти прогрес
Історія активності
Оновити
Дата
Заголовок
Час
Дії
28.10.2026 01:00
Security tests
45 mins
23.10.2026 01:00
JWT filter
3 hours 0 mins
15.10.2026 01:00
Liquibase setup
1 hour 0 mins
10.10.2026 01:00
JPA relations
2 hours 30 mins
05.10.2026 01:00
First REST controller
2 hours 0 mins
02.10.2026 01:00
Read Spring docs on IoC
1 hour 30 mins


Деталі прогресу
JPA relations
2026-10-10
2 hours 30 mins
Нотатка:
Struggled with lazy loading, fixed with fetch join
Закрити

AI Аналіз


Мої Цілі
Прогрес
AI Аналіз
Безпека (2FA)
AI Аналітик
Оберіть ціль для аналізу:
Learn Spring Boot
Параметри аналізу
Глибина аналізу (кількість логів)
10
Скільки останніх записів прогресу врахувати (1-50).
Генерувати аналіз
AI проаналізує ваші записи, час та нотатки, щоб надати рекомендації.
Історія рекомендацій
Оновити
Log analysis
27.10.2026 01:00
Читати


Результат аналізу
Log analysis
Створено: 27.10.2026, 01:00:36
A. Summary Progress Analysis: Overall Status: On Track. Key Achievements: REST API and JPA layer done. Problems/Blockers: refresh token flow. B. Recommended Next Steps: 1. Fix refresh flow. 2. Cover auth with tests.
Закрити

Висновки

Запропоновано та реалізовано двоетапний метод контекстної декомпозиції навчальних завдань за допомогою Google Gemini API.

Розроблено та програмно реалізовано евристичний алгоритм перевірки інваріанту здійсненності графіку навчання на рівні сервісної логіки.

Алгоритм зіставляє сукупні нормалізовані трудовитрати підцілей із наявним календарним фондом годин студента. Експериментальні випробування на 150 контрольних сценаріях підтвердили 100% точність виявлення та блокування фізично нереалістичних планів.

Спроєктовано відмовостійку багатошарову клієнт-серверну архітектуру за патернами CSR та DDD, яка поєднує реактивний SPA-інтерфейс (Vanilla JS, Nginx), серверний REST API (Java 21, Spring Boot 3, Spring AI) та ізольовану реляційну СУБД (PostgreSQL 15 із версіонуванням міграцій Liquibase).

Впроваджений конвеєр санітизації синтаксису забезпечив 100% успішне вилучення JSON-структур із 300 аномальних відповідей мовної моделі

Дякую за увагу!

Ai Learning Tracker — система інтелектуального планування та рефлексивного контролю навчальних цілей на основі генеративних мовних моделей

ДУІКТ

Державний університет інформаційно-комунікаційних технологій