Politechnika Śląska
Wydział Inżynierii Materiałowej I Cyfryzacji Przemysłu

Przedmiot:
Programowanie użytkowe

Temat:
Binary Game

autor	Sebastian Legierski, Jan Gryczon
prowadzący	dr inż. Mariusz Wnęk
rok akademicki	2025/2026
kierunek	Makrokierunek: Informatyka Przemysłowa
stopień	II stopnia magisterskie
rodzaj studiów	stacjonarne
semestr	1
sekcja	• Sebastian Legierski sekcja 1 • Jan Gryczon sekcja 2
termin oddania sprawozdania	2026-02-03

Założenia projektu

Stworzyć grę logiczną „Binary Game” – mniej popularną grę logiczną, podobną do sudoku. Gracz otrzymuje kwadratową siatkę o parzystych wymiarach, częściowo wypełnioną zerami i jedynkami. Zasady: nie można umieścić trzech identycznych cyfr (0 lub 1) obok siebie w pionie ani w poziomie, a każdy wiersz i każda kolumna muszą zawierać tyle samo zer co jedynek. Dodatkowo planowane jest użycie sztucznej inteligencji do rozwiązywania zagadek i udzielania graczowi podpowiedzi jako „koła ratunkowego”.

Projekt:

Backend: https://github.com/GryczonJ/BinaryGameBackEnd.git
Frontend: https://github.com/nimoCreator/BinaryGame-Frontend.git

Architektura techniczna:

Baza danych: Aembic’s ORM, Micorsoft SQL Server
Backend: FastAPI Python
Frontend: React
AI: Gemini

Dziennik prac nad projektem

Etap wstępny – analiza i planowanie (25–29 listopada 2025)

- analiza tematu projektu oraz wybór koncepcji aplikacji typu „Binary Game”
- określenie funkcjonalności systemu (zarządzanie użytkownikami, logowanie, zagadki/puzzle)
- zaplanowanie architektury aplikacji (backend + baza danych + uwierzytelnianie JWT)

23.12.2025 – rozpoczęcie implementacji

- utworzenie repozytorium projektu
- wykonanie pierwszego commita (initial commit)
- przygotowanie podstawowej struktury plików i katalogów
- konfiguracja środowiska backendowego

26.12.2025 – budowa fundamentów backendu i bazy danych

- zaprojektowanie oraz implementacja początkowego schematu bazy danych
- utworzenie modeli (users, puzzles itp.)
- implementacja rejestracji użytkownika i logowania

- dodanie haszowania haseł
- generowanie tokenów JWT do autoryzacji
- stworzenie schematów (schemas) do walidacji danych
- implementacja endpointów do obsługi użytkowników oraz zarządzania zagadkami
- dodanie kontroli uprawnień administratora
- aktualizacja plików projektu po kompilacji

27.12.2025 – porządkowanie kodu

- czyszczenie modułu security
- usunięcie zbędnych pustych linii
- poprawa czytelności i stylu kodu

12.01.2026 – refaktoryzacja backendu

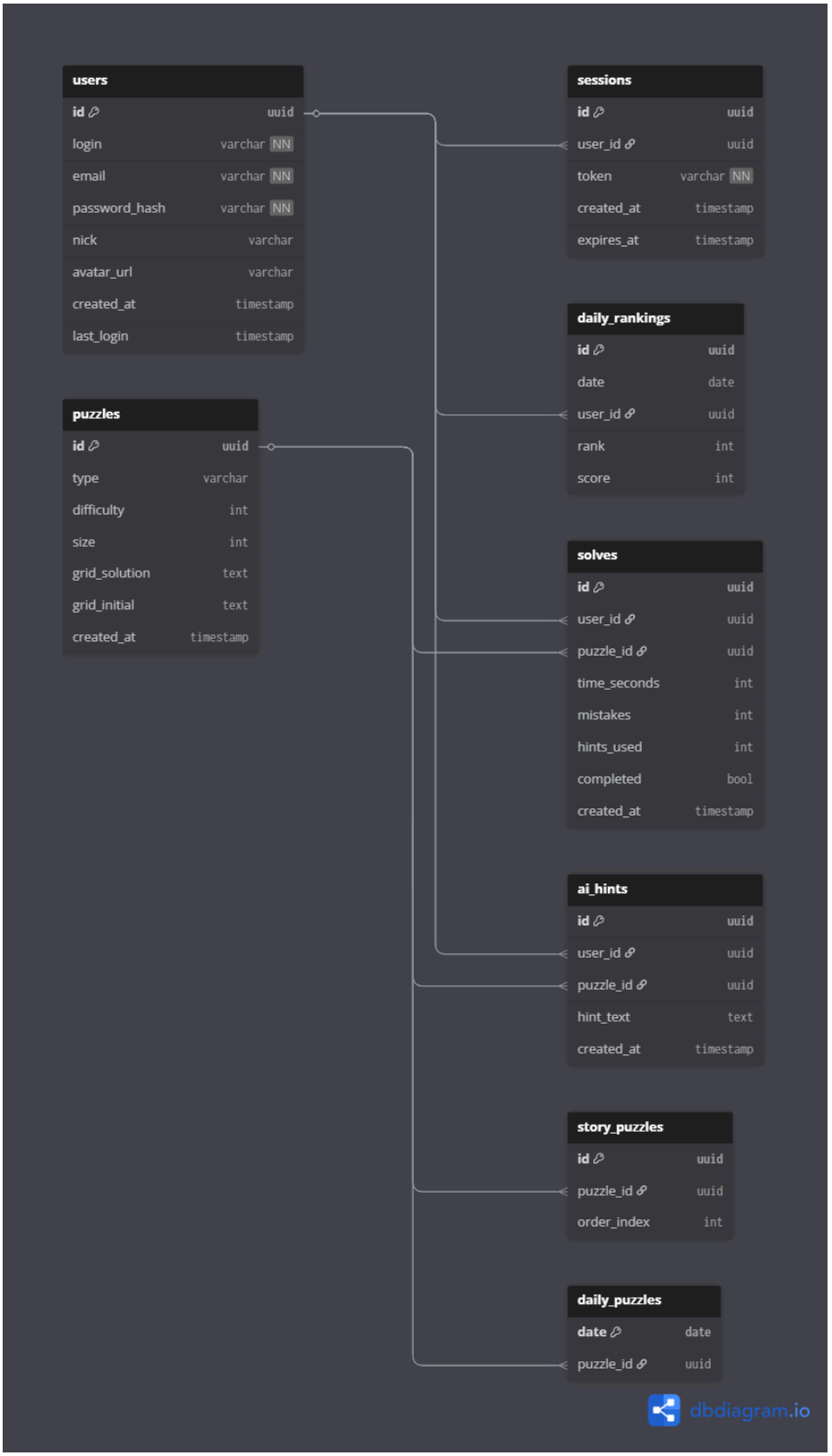
- refaktoryzacja modułu security
- uporządkowanie połączenia z bazą danych
- usunięcie redundantnego kodu
- uproszczenie oraz uporządkowanie importów

13.01.2026 – dalsza optymalizacja struktury projektu

- refaktoryzacja importów w modułach security i main
- usprawnienie użycia modelu użytkownika
- dalsze uproszczenie struktury kodu
- poprawa czytelności i utrzymywalności projektu

Przygotowanie prezentacji zaliczeniowej projektu, 26–30 stycznia 2026 roku.

Baza Danych:



Baza danych została zaprojektowana w modelu relacyjnym (Microsoft SQL Server) i obsługuje użytkowników, zagadki logiczne, rozgrywkę, rankingi oraz system podpowiedzi AI.

Struktura opiera się na centralnych encjach users oraz puzzles, do których przypisane są wyniki rozgrywek i statystyki.

Główne encje:

- **users**
  Przechowuje dane kont użytkowników: login, email, hasło (hash), profil oraz metadane logowania.
- **sessions**
  Obsługuje autoryzację poprzez tokeny sesji (JWT/refresh tokeny). Każda sesja jest powiązana z użytkownikiem i posiada datę wygaśnięcia.
- **puzzles**
  Bazowa tabela zagadek. Zawiera typ (story/daily/random), poziom trudności, rozmiar planszy oraz stan początkowy i rozwiązanie.
- **story_puzzles**
  Definiuje kolejność zagadek w trybie fabularnym (kampanii).
- **daily_puzzles**
  Mapuje konkretną datę na jedną dzienną zagadkę.
- **solves**
  Rejestruje próby rozwiązania zagadek przez użytkowników: czas, błędy, użyte podpowiedzi oraz status ukończenia.
- **daily_rankings**
  Ranking dzienny generowany na podstawie wyników użytkowników. Unikalność zapewniona dla pary (data, użytkownik).
- **ai_hints**
  Historia podpowiedzi generowanych przez AI dla użytkownika podczas rozgrywki.

Opis endpointów API

API zostało zaimplementowane w architekturze REST przy użyciu FastAPI.
Komunikacja odbywa się w formacie JSON, z autoryzacją tokenową.

AUTH & USERS

Endpointy odpowiedzialne za uwierzytelnianie i zarządzanie sesją.

- **POST /auth/register** – rejestracja nowego użytkownika
- **POST /auth/login** – logowanie i zwrot tokenów
- **POST /auth/logout** – unieważnienie sesji
- **POST /auth/refresh-token** – odświeżenie tokena dostępu
- **GET /auth/me** – dane aktualnie zalogowanego użytkownika

USER PROFILE

Zarządzanie profilem.

- **GET /users/me** – pobranie własnego profilu
- **PATCH /users/me** – aktualizacja danych profilu
- **GET /users/:id** – publiczny profil wskazanego użytkownika

PUZZLES (core game)

Pobieranie zagadek do rozgrywki.

- **GET /puzzles/story** – lista zagadek fabularnych
- **GET /puzzles/story/:id** – konkretna zagadka fabularna
  GET /puzzles/random – losowa zagadka
- **GET /puzzles/daily/today** – dzisiejsza zagadka
- **GET /puzzles/daily/:date** – zagadka z wybranego dnia

GAMEPLAY / SOLVES

Obsługa wyników gry.

- **POST /solves** – zapis wyniku rozwiązania
- **GET /solves/me** – historia własnych rozwiązań
- **GET /solves/puzzle/:puzzleId** – wyniki dla danej zagadki

RANKINGS & STATS

Statystyki i rankingi.

- **GET /rankings/daily/:date** – ranking dzienny
- **GET /rankings/daily/:date/top** – najlepsze wyniki dnia
- **GET /rankings/user/:userId** – statystyki użytkownika

CALENDAR

- **GET /calendar/daily** – kalendarz dostępności zagadek dziennych

AI HELP / HINTS

- **POST /ai/hint** – generowanie podpowiedzi AI dla aktualnej zagadki