

EGE YESILYURT

Richmond Hill, ON | 647-804-5834 | egeyesilyurtca@gmail.com

github.com/egeyesss | linkedin.com/in/egeyesss

Education

York University — North York, ON

Expected May 2027

Bachelor of Engineering, Major in Software Engineering

Relevant Coursework: Object-Oriented Programming, Operating Systems, Data Structures and Algorithms, Software Design, Design and Analysis of Algorithms, E-Commerce Systems

Projects

z9bra — Daily Logic-Puzzle Game | [Website](#) | [Source Code](#)

Live, June 2026

- Built and shipped a **Wordle-style daily puzzle web app** that grew to **500+ unique players** across **15+ daily puzzles**, using deterministic date-seeded selection so every player worldwide gets the same grid on a fixed America/Toronto reset.
- Engineered a **commit-based solving engine** (no undo, overwrite-count scoring), **server-rendered share cards via Satori**, and optional account sync (auth + database) for cross-device streaks with a localStorage fallback.
- Deployed on **Vercel** with a custom domain and automated **CI/CD** (GitHub-triggered preview + production deploys), behind a typed data contract that validates versioned puzzle bundles from the generator below.

Technologies: Next.js 16, React 19, TypeScript, Tailwind CSS, Vercel, Satori, Vitest, ESLint/Prettier, Git

Puzzle Generator — Constraint-Solver-Based Logic-Grid Engine | [Source Code](#)

May 2026 • Open Source

- Built a **research-grade Python engine** that generates the high-difficulty logic-grid puzzles powering z9bra, using **Google OR-Tools CP-SAT** as a uniqueness oracle so every exported puzzle is provably **single-solution** across 4x4 and 5x5 grids.
- Wrote a custom **propagation tracer** measuring human deduction depth and branching factor to score and tune difficulty, plus a 7-type clue system **cross-checked** between the CP-SAT translation and an independent pure-Python semantics implementation.
- Developed under **strict TDD** with **Pydantic v2** models, a **YAML-defined theme system** (one engine, many puzzle flavors), and a Click CLI; property-based tests, 42-test suite, mypy-strict + ruff clean, public MIT repo.

Technologies: Python 3.12, OR-Tools CP-SAT, Pydantic v2, pytest, Hypothesis, mypy, Click

BundesPredict — Calibrated Match Predictor with an LLM Agent | [Website](#) | [Source Code](#)

Live, July 2026

- Built a Bundesliga match-outcome predictor on a **hand-written Dixon–Coles Poisson model** (weighted MLE via SciPy, time decay, promoted-team shrinkage); **backtested walk-forward over 1,800+ out-of-sample matches** to a calibrated **RPS of 0.205**, within a hundredth of the de-vigged market line.
- Designed an **LLM tool-calling agent** that maps plain-English match context into **bounded, server-clamped expected-goals adjustments** — the deterministic model owns every probability; the LLM only proposes typed, audited nudges grounded in scraped squad data.
- Architected a Dockerized **FastAPI + Next.js + Postgres** monorepo with a **170+ test suite** (parameter-recovery, calibration, invariants), **GitHub Actions CI/CD**, and a weekly refit job; deployed on Vercel + Railway + Neon.

Technologies: Python, FastAPI, NumPy/SciPy, scikit-learn, Anthropic SDK, Next.js, TypeScript, PostgreSQL, Docker, GitHub Actions

Spocity — Spotify-Powered 3D Voxel City | [Website](#) | [Source Code](#)

Live, July 2026

- Built and shipped a web app that turns a user's **Spotify listening history** into an explorable **3D voxel city** rendered in-browser with **React Three Fiber** — each artist is a building whose height maps to a decay-weighted play score, auto-zoned into 10 genre districts.
- Engineered an **exponential-decay scoring engine** (~90-day half-life) with a nightly **Celery + Redis** recompute that re-tiers buildings and emits growth/decay events, so the skyline evolves as listening habits shift.
- Integrated **Spotify OAuth 2.0** (Auth Code + PKCE) into Django session auth and built a client-side **WebGL postcard generator** for shareable snapshots; deployed frontend on **Vercel**, Django backend + Postgres on **Railway**.

Technologies: Next.js, React, TypeScript, React Three Fiber, Django REST Framework, Celery, Redis, PostgreSQL, Spotify API, OAuth 2.0, Docker

Fitiva — Workout Training Planner Web App | [Source Code](#)

April 2026

- Led a 6-person team as **project lead**, running sprint planning and task delegation through Jira to hit every deadline.
- Architected a **full-stack planner**: user/trainer authentication, fitness profiles, program scheduling, an exercise template library, and workout recommendations.
- Designed a **repository-pattern data layer** with a stub/real DB swap for fast isolated tests; wrote **40+ Django unit tests** and containerized the full stack with Docker.

Technologies: Next.js, TypeScript, React, Python, Django REST Framework, MySQL, Docker, Jira, REST APIs, Git

Skills

Technical: Python, Java, TypeScript, JavaScript, SQL, HTML/CSS, MATLAB, Django REST Framework, FastAPI, Next.js, React, React Three Fiber, Celery, Docker, Kubernetes, PostgreSQL, MySQL, Redis, Git, REST APIs, OAuth 2.0, Jira

Languages: Fluent in English, Turkish; N3 Level Japanese