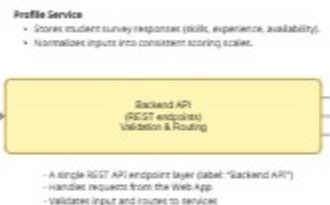


- Users → Web App
- Web App → Backend API
- Backend API → Profile Service → Database
- Backend API → Optimization Engine → Database
- Backend API → Scoring & Analytics → Database
- Backend Services → Logs/Monitoring

Students submit skill + availability survey data.
Professors configure constraints and trigger team generation, then review/export results.

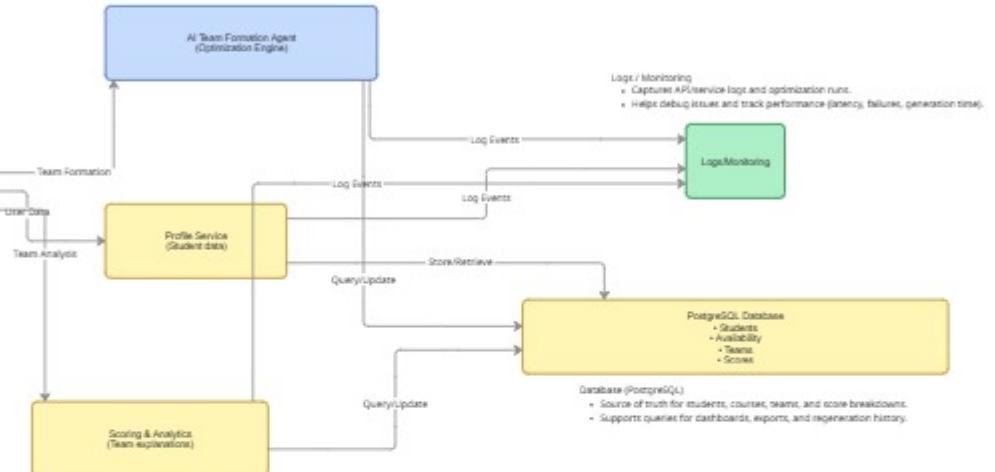


Optimization Engine (AI Team Formation Agent)

- Builds candidate team configurations under constraints (team size, balance, overlap).
- Scores candidates and selects the best overall team assignment.

Scoring & Analytics (Team Explanations)

- Validates team configurations under constraints (team size, balance, overlap).
- Score candidates and select the best overall team assignment.



TeamMatch System Architecture - Technology Breakdown

Frontend - React / Next.js

We are using **React with Next.js** to build the web application interface.

Why:

- Components-based architecture for scalable UI
- Fast rendering and clear state management
- Easy integration with REST APIs
- Production-ready build output

Purposes in our system:

- Students complete survey
- Professors configure constraints
- Displays optimized team results and scoring transparency

API Layer - REST backend (Node.js / Express)

We use a **Node.js backend with Express** to handle HTTP requests.

Why:

- Lightweight and fast
- Works well with JSON-based APIs
- Easy integration with PostgreSQL
- Simple routing and middleware support

Purposes in our system:

- Receive frontend requests
- Validate and sanitize input
- Routes requests to internal services
- Act as the secure entry point to the system

Backend Service - Microservice Logic

Inside the backend, we logically separate responsibilities:

Profile Service

Handle:

- Scoring survey data
- Normalizing skill values
- Managing availability inputs

Optimization Engine (Core AI Agent)

Implement:

- Constraint satisfaction logic
- Heuristic optimization
- Scoring function evaluation
- Selection of best team configuration

Technology:

- Implemented in Node.js (or Python if using optimization libraries)
- Uses combinatorial search or greedy heuristics

This is the "intelligence" of the system.

Scoring & Analytics Service

Responsible for:

- Calculating team balance metrics
- Generating score breakdowns
- Providing transparency explanations

This ensures our AI is explainable, not a black box.