Grading Dino

Research Document

Overview

Backend	Django 5.0
Database	PostgreSQL 16
Password Hash	Argon2
Frontend	Pico.css
Container	Docker + Docker Compose
Language	Python 3.11

Django

Built-in Features:

- **Admin Panel** - Complete grade management UI for professors without building custom interface
- **Authentication** - User roles and permissions out-of-the-box
- **Forms + Validation** - Automatic form generation and server-side validation
- **Security** - CSRF, XSS, SQL injection protection by default
- **ORM** - Type-safe database operations

Django Philosophy:

- "Batteries included" approach
- Convention over configuration
- Security by default
- Rapid development focus
- Perfect match for grade management system requirements.

Good match for grade management system requirements. A little bit too advanced for the smaller project but perfect for fast development.

PostgreSQL

Selected for:

- Docker-native integration (avoids file permission issues)
- Concurrent write support (multiple professors can grade simultaneously)
- Production-ready deployment path
- Full ACID compliance for data integrity

Argon2 Password Hashing

Selection Rationale:

- **Security Standard** - Winner of Password Hashing Competition (2015)
- **Luxembourg Connection** - Developed at University of Luxembourg - provides strong local academic credibility
- **Django Integration** - Native support via django.contrib.auth.hashers.Argon2PasswordHasher
- **Modern Algorithm** - Memory-hard design, resistant to GPU/ASIC brute-force attacks (superior to bcrypt/PBKDF2)

The Luxembourg academic origin strengthens our technical justification significantly.

Pico.css + Server-Side Rendering

Pico.css Selection:

- Minimalist single-file CSS framework (~10KB)
- Semantic HTML - automatically styles standard HTML5 elements without custom classes
- Zero learning curve - just write HTML
- Professional, modern appearance by default
- Perfect for Django templates

Server-Side Rendering Decision:

- Django's native and recommended approach
- Security: All validation and logic server-side
- Simplicity: Standard HTTP POST → Process → Render cycle
- No JavaScript complexity
- Django CSRF protection works seamlessly with forms

Docker + Docker Compose

Project Requirements:

- System must start with single docker compose up command
- PostgreSQL + Django in isolated containers
- Environment consistency across team members
- Production-like architecture from start

Decision Summary

Django delivers a complete, secure grade management system with minimal custom development required.