Policy Play

Problem statement: 9

(Policy Play: Turn Dense Policies Into Interactive Learning & AI-Guided Drafting)

Policy play is a gamified policy learning platform that converts policy documents into structured data using Groq AI, stores them in MongoDB, and delivers interactive games to improve policy understanding for employees. This document provides an official overview of the system, architecture, features, setup, and future roadmap.

1. Introduction

PolicyPlay addresses the challenge of low engagement with traditional policy documents by transforming them into interactive, game-based learning experiences. The platform is designed for enterprise environments, with a focus on ease of deployment, user engagement, and extensibility for future integrations such as Azure and Microsoft 365.

Check the application click on this Link (running live) : http://3.111.52.146:3000/

Frontend : http://3.111.52.146:3000/
Backend : http://3.111.52.146:8000/docs

GitHUb Link : https://github.com/Ranjithkumar7760/PolicyPlay-hackathon.git

2. System Overview

PolicyPlay ingests policy documents in PDF or DOCX format, processes them through Groq AI to generate structured JSON, and persists the extracted content in MongoDB for use across multiple game modules. The platform exposes RESTful APIs via a FastAPI backend and delivers the user experience through a Next.js 14 frontend with modern React-based UI patterns.

Key capabilities include:

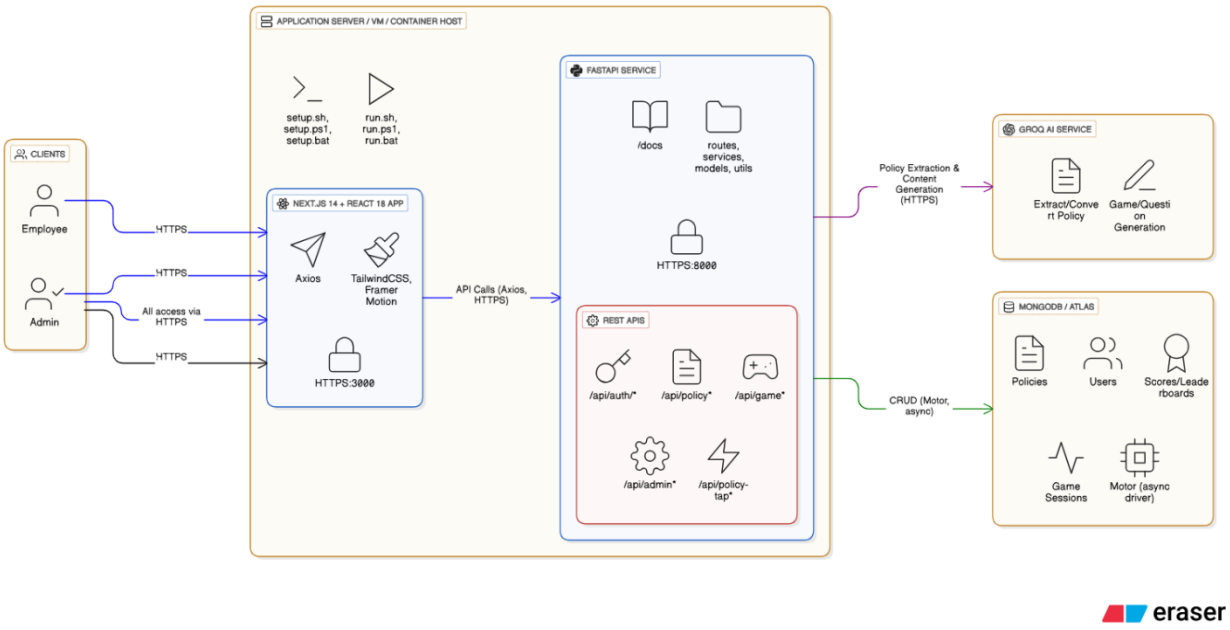
- Policy upload, parsing, and structured storage
- Multiple game types for policy learning and assessment
- User authentication, scoring, and leaderboards

- Admin capabilities for policy and user management

3. Architecture

3.1 High-Level Architecture

The system follows a modular, service-oriented architecture with clear separation between frontend, backend, and data layers.



3.2 Technology Stack

Backend:

- FastAPI for high-performance, Python-based API services with automatic OpenAPI documentation.
- MongoDB / MongoDB Atlas as the NoSQL data store for flexible policy and game document structures.
- Motor as the asynchronous MongoDB driver.
- Pydantic for schema validation and data modeling.
- Groq AI for OpenAI-compatible LLM-powered policy parsing and game generation.

Frontend:

- Next.js 14 with App Router for routing, server components, and modern React patterns.
- React 18 for UI components.
- TailwindCSS for utility-first styling.
- Framer Motion for animations.
- Axios for HTTP communication.

3.3 Directory Structure

The project is organized as follows:

- backend/: FastAPI application (routes, services, models, utilities, main entry point, and dependencies)
- frontend/: Next.js application (routes, components, dependencies)
- Cross-platform setup and run
  scripts: setup.sh, setup.ps1, setup.bat, run.sh, run.ps1, run.bat
- Documentation: README.md, QUICKSTART.md, SETUP_INSTRUCTIONS.md

4. Features

4.1 Policy Upload and Processing

- Upload of policy documents in PDF and DOCX formats through the admin interface.
- Invocation of Groq AI to extract key sections, rules, and entities from unstructured policy text into structured JSON.
- Storage of processed policy objects in MongoDB for reuse across game modules and analytics.

4.2 Game Modules

- Policy Tap: Real-time game where users tap falling option elements to select correct answers derived from policy content.
- Scenario-Based Games: Multiple-choice questions based on realistic situations referencing specific policy rules.

- Violation Detection: Tasks where users identify policy violations from provided scenarios to reinforce compliance behavior.

4.3 User Management

- User registration and authentication (credentials-based initially).
- Personal score and progress tracking across game sessions.
- Leaderboards to boost engagement and friendly competition.

4.4 Admin Dashboard

- Policy lifecycle management (upload, list, and manage available policies).
- User analytics, including engagement and performance indicators.
- Visibility into game performance and effectiveness of policy training.

6. API Overview

The backend exposes RESTful endpoints with interactive API documentation available at http://localhost:8000/docs via FastAPI's built-in OpenAPI UI.

Core endpoint groups:

- Authentication: /api/auth/signup, /api/auth/login
- Policies: /api/policy/upload, /api/policies
- Games: /api/game/start, /api/game/submit
- Policy Tap: /api/policy-tap/generate, /api/policy-tap/start, /api/policy-tap/submit
- Admin: /api/admin/users, /api/admin/scores

7. Installation and Prerequisites

7.1 Prerequisites

- Python 3.10 or higher
- Node.js 18 or higher
- MongoDB Atlas account or local MongoDB instance

- Groq API key (optional: the system can operate with fallback behavior)

7.2 Installation Process

- Run the platform-specific setup script to install backend and frontend dependencies.
- No manual pip or npm installation steps are required under normal circumstances.
- Default fallback values allow the system to run locally without immediate external configuration.

8. Configuration

The backend supports environment-based configuration via .env files.

Key configuration inputs:

- Groq API key for production-grade AI performance and reliability.
- MongoDB connection string, which can point to MongoDB Atlas or on-premise MongoDB.

Fallback values are provided for local development and demo usage. For production, values must be updated in backend/.env to ensure proper security and scalability.

9. Usage Guide

Standard usage workflow:

1. Execute setup script for the relevant OS.
2. Execute run script to start backend and frontend.
3. Access the frontend at http://localhost:3000.
4. Sign up or log in as a user.
5. Select a policy, choose a game mode, and begin playing to reinforce knowledge of policy content.

10. Testing and Validation

A baseline test checklist includes:

- Setup scripts executing successfully on Windows and Linux/macOS.
- Backend and frontend starting without errors.

- API endpoints responding correctly and visible in Swagger UI (/docs).
- Game modules functioning end-to-end (question generation, answering, scoring).
- User authentication, score tracking, and leaderboards working as intended.
- Admin dashboard pages loading and reflecting policy and user data.

11. Roadmap and Future Enhancements

11.1 Game Module Expansion

Planned enhancements:

- Additional game types such as quiz battles, matching games, timelines, compliance simulators, and interactive scenarios.
- Dynamic game generation that adapts to any uploaded policy document, leveraging Groq AI for automated content generation.
- Multiplayer modes for real-time competitive sessions and team-based learning.
- Achievement and badge systems to reward policy mastery and sustained engagement.

11.2 Enterprise Authentication and Access Control

Planned enterprise readiness features:

- Integration with Microsoft Entra ID (Azure AD) for authentication and SSO.
- Restriction of access to verified employees (e.g., specific tenant domains).
- Role-based access control (RBAC) for admins, managers, and standard users.
- Potential integration with Microsoft 365 for streamlined access and notifications.

11.3 Cloud Hosting and Infrastructure

Planned Azure-based deployment model:

- Backend hosted on Azure App Service for scalable, managed API hosting.
- Frontend hosted on Azure Static Web Apps or Azure App Service, depending on deployment requirements.
- Azure Cosmos DB with MongoDB API as a managed database backend.
- Azure Blob Storage for policy document storage and retrieval.

- Azure CDN for faster global content delivery.
- Azure Application Insights for monitoring, logging, and performance analytics.

11.4 Security and Compliance

Planned security enhancements:

- End-to-end encryption for sensitive data in transit (TLS) and at rest in storage.
- API rate limiting, throttling, and basic DDoS protection.
- Secure file upload pipelines including validation and malware scanning.
- Regular security assessments, dependency vulnerability scanning, and patch management.
- Implementation of secure session management and token refresh flows.
- Hardened CORS policies and security headers.
- Alignment with data protection and compliance standards such as GDPR and SOC 2, where applicable.

11.5 Analytics, Mobile, and Integrations

Planned functional enhancements:

- Advanced analytics dashboards for administrators, including policy comprehension and engagement metrics.
- Custom reporting and export features to support audits and HR reviews.
- Native or hybrid mobile applications (iOS and Android) with offline policy review and push notifications.
- Integration with HR systems, Slack/Microsoft Teams notifications, email reminders, and calendar-based reminders for policy deadlines

Thank you