GAZİ UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CENG 473 SECURE CODING - CENG367 SCRIPTING LANGUAGES

PROJECT PROPOSAL FORM

GROUP 6

Fundanur Öztürk – 21118080056

Reyyan Temel – 21118080015

Hatice Sevde Kaplan – 21118080090

Osman Sefa Coşar – 21118080013

Baran Türkmen - 21118080012

NOVEMBER 2025

Digital Ethics Monitor – A Secure and Transparent AI Framework

Introduction

The movie Tron (1982) depicts a world where human consciousness and computer systems merge inside a digital universe. Within that universe, programs behave like autonomous entities, governed by rigid rules and central authorities. This concept inspires the Perfect System Project, which aims to explore the relationship between human ethics and digital control in modern computing systems.

In today's world, artificial intelligence systems make autonomous decisions in areas such as recruitment, healthcare, and finance. However, these systems often lack transparency, ethical awareness, and security. The Digital Ethics Monitor (DEM) is proposed as a "perfect system" that ensures AI decisions remain transparent, explainable, and aligned with ethical standards — all built upon secure coding principles.

Project Objective

The primary objective of this project is to design and develop a secure and ethical monitoring system for AI-driven applications. The system aims to detect unethical or biased decisions produced by artificial intelligence models and ensure transparency by securely logging all critical actions and decisions. By integrating secure coding principles, the framework will protect data integrity, confidentiality, and user privacy. Furthermore, the system will establish a trustworthy interface between human users and machine intelligence, allowing ethical oversight and accountability within AI decision-making processes.

Problem Definition

Although AI systems are highly efficient and capable of automating complex decision-making processes, they can also produce harmful or biased outcomes when not properly monitored. Such issues often arise from poor data quality or inherently biased datasets, which can skew model predictions and lead to unfair results. Additionally, the lack of explainability—often referred to as the black-box problem—makes it difficult for users to understand how AI models reach their conclusions. Compounding these risks, insecure data storage and communication channels can expose sensitive information to unauthorized access or manipulation. These challenges underscore the necessity of a Digital Ethics Monitor (DEM) — a system designed to uphold human values and ethical integrity within digital environments, much like how Tron's protagonist strives for justice within the digital grid.

Methodology

The Digital Ethics Monitor (DEM) is built on a multi-layered architecture to ensure security, transparency, and ethical accountability. The Input Validation Layer secures incoming data against corruption and attacks. The Ethical Evaluation Engine analyzes AI outputs for bias and fairness. The Secure Logging Module records all AI decisions in encrypted form with restricted access. Lastly, the Transparency Dashboard provides users with clear reports on system actions, promoting trust between humans and AI.

System Description

The Digital Ethics Monitor (DEM) will be developed using Python with a modular architecture to ensure scalability and security. The back-end will be implemented with FastAPI, while data will be stored in an encrypted PostgreSQL database using AES-256 encryption. All communication between modules will be protected through TLS 1.3.

AI bias detection will be performed using scikit-learn and Fairlearn, applying fairness metrics such as demographic parity and equalized odds. System activities and AI decisions will be securely recorded using hash-based logging (SHA-256) and Role-Based Access Control (RBAC) for data access.

The front-end dashboard, built with HTML5, Tailwind CSS, and Chart.js, will visualize ethical reports for authorized users. Authentication will be managed via JWT tokens. Static code analysis tools like Bandit will be used to ensure compliance with secure coding standards throughout development.

Security Considerations

The system is designed with security-by-design principles. All user inputs are sanitized to prevent injection attacks. Sensitive information is encrypted both in transit and at rest. Authentication is handled through JWT tokens, and unauthorized access is restricted via RBAC policies. Error handling mechanisms prevent data leakage, while static code analysis tools (Bandit) detect insecure code patterns before deployment.

PROJECT TEAM:

Fundanur Öztürk – 21118080056

Reyyan Temel - 21118080015

Hatice Sevde Kaplan – 21118080090

Osman Sefa Coşar – 21118080013

Baran Türkmen – 21118080012