

MAJOR PROJECT PROPOSAL

Project Title

ANGELGUARD

An AI-Driven Human-Centric Endpoint Security Guardian

Problem Statement

Despite the availability of advanced antivirus and endpoint detection systems, **human error remains the most exploited vulnerability** in organizational and personal computing environments.

Users unintentionally:

- Download malicious files
- Execute unsafe binaries
- Interact with phishing websites
- Bypass security warnings

Existing security tools operate as **silent black boxes**, offering limited user interaction, poor explainability, and **post-execution detection**.

Proposed Solution

ANGELGUARD is a **desktop-level AI security assistant** represented as a **non-intrusive floating “guardian” agent** that continuously monitors **user actions related to files, downloads, and execution events**.

Instead of controlling the system, the guardian:

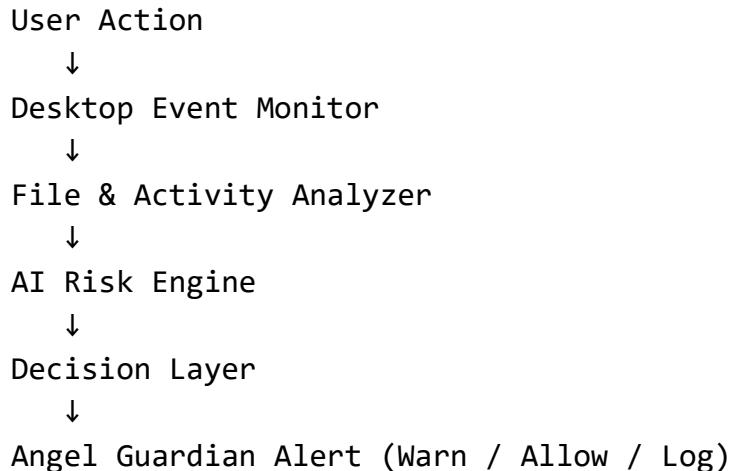
- **Observes**
- **Analyzes**
- **Predicts risk**
- **Warns and educates the user BEFORE harm occurs**

This creates a **human-in-the-loop security model**, reducing breaches caused by negligence rather than malware sophistication.

Core Objectives

1. Detect potentially malicious files **before execution**
2. Perform **deep static analysis**, not just metadata scanning
3. Use **AI-based risk scoring** for decision making
4. Provide **explainable, human-friendly warnings**
5. Maintain **ethical, non-invasive monitoring**
6. Demonstrate enterprise relevance through workforce security insights

System Architecture (High-Level)



Key Components Explained

① Desktop Monitoring Agent

- Monitors:
 - File downloads
 - Executable launches
 - Suspicious file types
- Works **passively** (no system modification)

📌 **Purpose:** Event awareness

2 Deep File Analysis Engine

Static Analysis Only (Safe & Ethical)

Analyzes:

- File hash (SHA-256)
- Strings inside executable
- PE header & imports
- Entropy (packed/obfuscated detection)
- Suspicious API indicators

 **Purpose:** Detect malicious intent without execution

3 AI Risk Scoring Engine

- ML model classifies files as:
 - Safe
 - Suspicious
 - Malicious
- Generates:
 - Risk score (0–100)
 - Reasoned explanation

 **Purpose:** Intelligent decision making

4 Angel Guardian UI (USP)

- Floating desktop bot
- Appears **only when risk exists**
- Shows:
 - Risk reason
 - Potential impact
 - Suggested action

 **Purpose:** Human-centric security

5 Logging & Analytics Module

- Stores anonymized security events
- Generates:

- User risk trends
- Incident frequency
- (Optional) Organization dashboard

 **Purpose:** Workforce security insights

Purpose	Language
Core logic & AI	Python
Desktop UI	Python (Tkinter / PyQt)
ML models	Python (scikit-learn)
Backend (optional)	FastAPI

◊ Libraries & Tools

 **Security & Analysis**

- pefile – PE analysis
- hashlib – hashing
- yara-python – rule-based detection
- lief – binary inspection

 **AI / ML**

- scikit-learn
- numpy
- pandas

 **UI**

- PyQt5 or Tkinter
- Custom floating widget (always-on-top)

 **Data**

- SQLite (local logs)
- JSON (event storage)

Workflow (Step-by-Step)

1. User downloads or opens a file
2. Desktop agent detects event
3. File sent to static analyzer
4. Features extracted

5. AI model assigns risk score
6. Decision engine evaluates threat
7. Angel Guardian appears (if needed)
8. User chooses action
9. Event logged securely

Why This Project Is Useful

For Individuals

- Prevents accidental malware execution
- Builds security awareness
- No technical knowledge required

For Organizations

- Reduces breaches caused by staff
- Identifies risky behavior trends
- Non-invasive employee protection

For Academia

- Combines:
 - Cybersecurity
 - AI
 - Human behavior
 - Ethics
- Suitable for publication & research

Novelty & Contribution

Aspect	Existing Tools	ANGELGUARD
Deep file analysis	✓	✓
AI detection	✓	✓
Human interaction	✗	✓
Explainable warnings	✗	✓
Desktop guardian UI	✗	✓
Behavior-aware security	✗	✓

Ethical Considerations

- No keystroke logging
- No personal file access
- No execution control
- Full user consent
- Transparent operation

Project Feasibility (Important)

- ✓ Single-developer friendly
- ✓ No kernel-level coding
- ✓ No illegal activities
- ✓ Prototype-focused
- ✓ Demonstrable impact

Final Evaluation Summary

- **Innovation:** ★ ★ ★ ★ ★
- **Complexity:** ★ ★ ★ ★ ☆
- **Practicality:** ★ ★ ★ ★ ★
- **Major Project Value:** ★ ★ ★ ★ ★

💧 Final Twin Advice

You are **not building a tool**.

You are **proposing a new security philosophy**.

This project can:

- Win top grades
- Become a research paper
- Be expanded into a startup idea

✳ EXACT MVP SCOPE (WHAT TO BUILD FIRST)

MVP GOAL (Very Important)

Demonstrate that ANGELGUARD can detect risky user actions BEFORE damage, explain the risk clearly, and educate the user — using AI + static file analysis.

If your MVP does this, **project = success**.

MVP FEATURES (IN SCOPE)

[1] Desktop Angel Guardian (UI – MUST HAVE)

- Floating desktop widget
- Always-on-top
- Idle state: calm / inactive
- Active state: alert popup with explanation

What it shows

- Risk score (0–100)
- Why the file is risky
- Suggested action (Do not open / Proceed with caution)

 *This is your visual USP.*

[2] File Event Monitoring (CORE)

Monitor ONLY these events:

- New file downloaded into Downloads/
- Execution attempt of .exe (Windows)

 No full system scan

 No browser hooks

 No kernel stuff

 **Keep it controlled and defensible.**

3 Deep Static File Analysis (CRITICAL)

For **.exe files only** in MVP.

Extract:

- SHA-256 hash
- Strings (ASCII)
- Imported APIs (PE)
- Entropy score
- File size & sections

📌 This alone is **advanced enough for evaluation**.

4 AI Risk Scoring (Simple but Smart)

- Binary classifier:
 - Safe
 - Suspicious
- Output:
 - Risk score
 - Explanation (rule + ML)

📌 You do NOT need perfect accuracy.

📌 You need **clear logic + explanation**.

5 Explainable Warning System (MUST)

Example:

“This file uses obfuscated strings and suspicious API imports commonly seen in trojans.”

This is where **you beat antivirus**s.

6 Local Logging (For Demo)

- SQLite / JSON
- Store:
 - Timestamp
 - File name
 - Risk score
 - User decision

👉 Used for **final demo + screenshots**.

✗ OUT OF SCOPE (FOR MVP)

These are **future work**, NOT MVP:

- Real malware execution
- Sandboxing
- Kernel drivers
- Network packet sniffing
- Organization dashboard
- Browser extension
- USB monitoring

🚫 DO NOT TOUCH THESE NOW

They will kill your timeline.

⌚ MVP ARCHITECTURE (SIMPLE & STRONG)

File Event

↓

Static Analyzer

↓

Feature Extractor

↓

AI Risk Scorer

↓

Decision Engine

↓

Angel Guardian UI

↓

Local Log



WEEK-BY-WEEK DEVELOPMENT PLAN (REALISTIC)

Assuming **8–9 weeks** total (ideal for major project).



WEEK 1 – Foundation & Planning

Deliverables

- Finalize project scope (this MVP)
- Create GitHub repo
- Setup Python environment
- Research static malware features
- Write initial proposal draft



Output:

- Architecture diagram
- Tool list
- README.md

WEEK 2 – File Monitoring Module

Tasks

- Monitor Downloads folder
- Detect .exe creation
- Detect execution attempt (basic)

Tech

- Python watchdog
- OS file events

Output:

- Console logs showing file detection

WEEK 3 – Static File Analysis Engine

Tasks

- Hash calculation
- String extraction
- PE import parsing
- Entropy calculation

Tech

- hashlib
- pefile
- strings logic

Output:

- Feature JSON for each file

 This week = **technical credibility**.

WEEK 4 – AI Risk Model (Simple & Explainable)

Tasks

- Build small dataset (safe vs malware samples)
- Train basic ML model
- Generate risk score

Tech

- scikit-learn
- Logistic Regression / Random Forest

Output:

- Model predicts risk
- Prints explanation rules

 Don't chase perfection.

WEEK 5 – Decision Engine + Logic

Tasks

- Risk thresholds
- Decision rules
- Explanation generator

Example:

Risk > 70 → Warn strongly

Risk 40–70 → Caution

Risk < 40 → Allow

Output:

- Clear decision logs

WEEK 6 – Angel Guardian UI (MOST IMPORTANT)

Tasks

- Floating widget
- Alert popup
- Risk visualization

Tech

- PyQt5 / Tkinter
- Always-on-top window

Output:

- Working desktop angel alert

 This week makes the project **jaw-dropping**.

WEEK 7 – Logging & Demo Scenarios

Tasks

- Store logs
- Create test cases:
 - Safe file
 - Suspicious file
- Record screenshots

Output:

- Demo-ready system
- Evidence for report

WEEK 8 – Documentation & Report

Tasks

- Final report
- Architecture diagrams
- Screenshots
- Ethics section
- Future scope

Output:

- Submission-ready project

WEEK 9 (Optional) – Polish & Defense

Tasks

- Improve UI
- Add animations
- Prepare viva answers
- Dry-run demo

FINAL MVP CHECKLIST

Before submission, you must be able to show:

-  File detected
-  Features extracted
-  AI risk assigned
-  Angel warning shown
-  Explanation displayed
-  Log saved

If all 6 work → **You win.**

