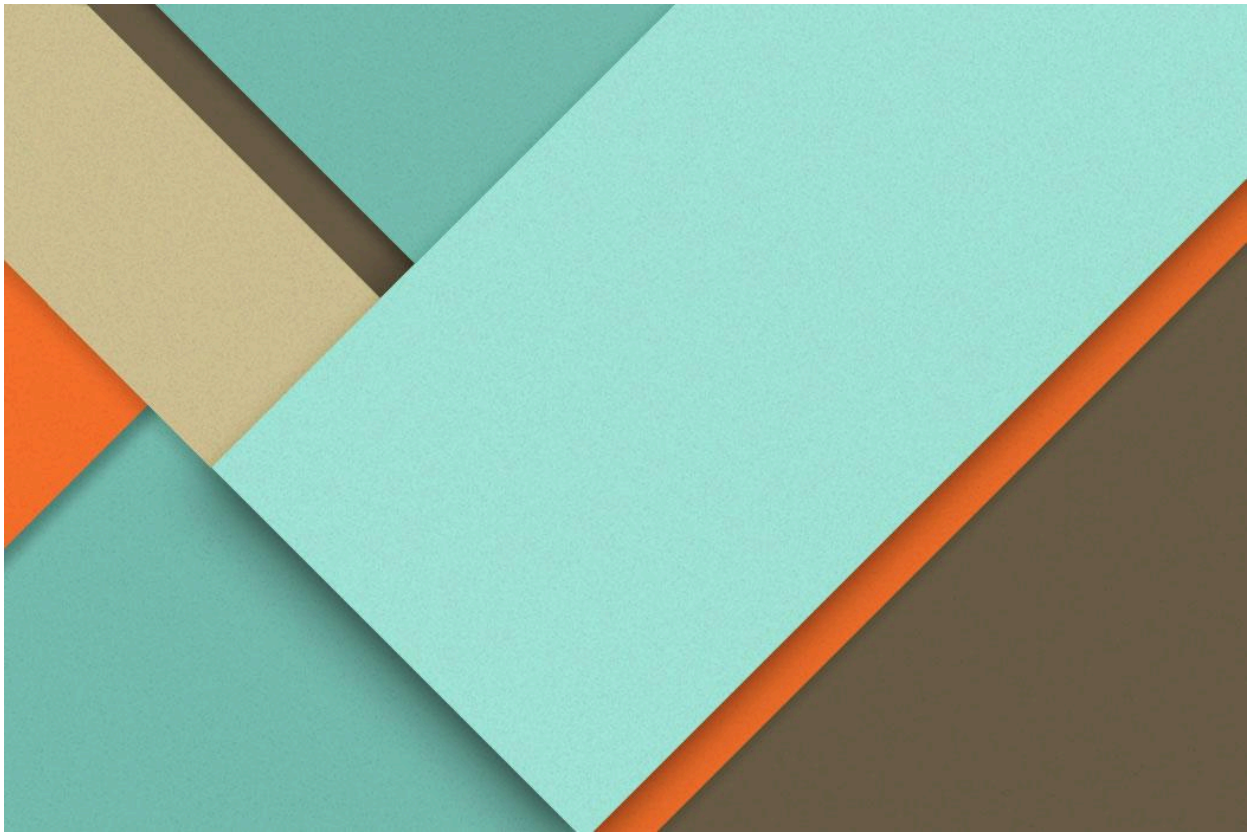

E.L.L.A

Enhanced - Locust - Logic - Architecture

11.04.2025

#Group Python

Overview

E.L.L.A (Enhanced Locust Logic Architecture) is a Python-based system designed for **high-speed, intelligent data recovery** from local databases and distributed servers. Inspired by the **collective intelligence and efficiency of locust swarms**, this architecture models nature's decentralization to provide fault-tolerant, parallel, and ultra-responsive data retrieval.

This project is ideal for scenarios requiring rapid access to large or fragmented datasets—such as search systems, logging infrastructures, or backup recovery solutions—built entirely with native Python modules (no external libraries).

Goals

1. Deliver a **lightweight yet powerful system** for request-driven data recovery
2. Use **nature-inspired algorithms** (like swarm routing and redundancy mapping)
3. Minimize data access latency with **threaded cache-first architecture**
4. Build an educational and scalable solution suitable for academic and enterprise use

Specifications

Systems Architect

Design ELLA's architecture (modules, workflow, cache recovery, fallback logic). Ensure alignment with locust-inspired logic.

Skills : Python OOP, design patterns, diagrams, system thinking

Data Handler

Build and maintain the database layer (`intel_db.py`), create mock data, and optimize queries. Ensure fallback compatibility

Skills : SQL, Python `sqlite3`, data modeling

Concurrency Engineer

Develop threaded systems (`router.py` and `fallback_recovery.py`). Optimize for speed and safety using Python's threading model.

Skills : Python threading, synchronization, queues

Caching Strategist

Create intelligent caching algorithms (`locust_cache.py`) based on memory-first access. Mimic swarm memory logic

Skills : Memory optimization, caching strategies, indexing

Documentation Commander

Own the `README.md`, internal comments, and final project report (PDF). Also creates walkthroughs and diagrams.

Skills : Markdown, technical writing, presentation tools

Please note all members are to join and contribute actively on Github

-Visit E.L.L.A Repo on Github (Click here)

Once made admins on the repo, each student has been assigned his/her personal folder to upload your code for merge and review. Thank You.