



AI Software Engineer, 정예울입니다.

AI 기술을 활용하여 혁신적인 솔루션을 개발하고, 다양한 경험을 통해 지속적으로 성장하고 있습니다!

새로운 도전을 통해 배운 지식을 실생활의 아이디어와 결합하여 창의적인 가치를 창출합니다. 😊

contact	010-3595-9818
E-mail	papooo.dev@gmail.com
Github	github.com/papooo-dev
Blog	now : https://papooo-dev.github.io/ previous : [https://creamilk88.tistory.com/](

Skills

- **AI:** Langchain, Prompt Engineering
- **Languages:** Java, Python, JavaScript
- **Frameworks:** Spring Boot, FastAPI, Flask, Node.js, React
- **Tools & Platforms:** Docker, AWS, Git, GitLab
- **Databases:** PostgreSQL, Oracle, MySQL
- **Others:** CICD, LLM, Memory, RAG

Recent Interests and Studies

- **AI Trends:** 최신 AI 트렌드를 Medium으로 학습하고, 블로그에 정리해서 지식을 나누는 걸 좋아해요.
- **Generative AI Applications:** AI 어플리케이션 개발자로 성장하기 위해 AI 엔지니어 로드맵을 따라 체계적으로 공부하고 있어요.
- **System Design:** AI 시대에서 가장 중요한 역량이라고 생각해요. 시스템 설계와 다양한 설계 패턴에 대해 배우고 있어요.
- **Deployment:** AWS를 활용한 배포 및 관리 역량을 키우는 데 관심이 있어요.
- **Side Projects:** 제가 실생활에서 필요했던 어플리케이션을 사이드 프로젝트로 개발하면서 실무 경험을 쌓고 있어요.

Work Activities

Bankware Global

2021.03 ~ 현재

사내 AI 툴 개발, 기술문서 자동화, 코어뱅킹 시스템 개발, 테스트 자동화 프로그램 운영

Core Responsibilities & Achievements

- **Java Doc을 통한 자동 API 문서 프로세스 구축:** Java file의 주석을 이용해, 배포 시 Docusaurus API 명세서 사이트에 자동으로 반영되는 프로세스를 구축을 진행 중입니다.
- **AI 기반 어플리케이션 개발:** BDD 개발 방식에 자동화를 도입하는 툴과 대량의 테스트 데이터를 생성하는 프로그램을 AI를 활용하여 개발하였습니다.
- **LangChain 사내 강의 진행:** 10회 차로 구성된 강의를 통해 기본 이론부터 실습까지 진행하며, 사내 AI 제품 개발에 기여하였습니다.
- **BXI Simulator 버전 2 개발:** 대외 연계 시뮬레이터 솔루션을 개발하고, REST API를 통한 거래 테스트 기능을 추가하였습니다.
- **LINE Bank Japan 코어뱅킹 시스템 개발:** 고객업무를 담당하여 비대면 고객가입 및 고객 관리 시스템을 개발하고, 테스트 자동화 프로그램을 설계 및 운영하여 시스템 안정성을 확보하였습니다.

Impactful Experiences & Insights

AI 기반 BDD 도구 도입 실패와 극복 경험: 사용자 중심 개발의 중요성

- **실패 사례:** AI 기반 BDD 자동화 솔루션 OiBDD를 개발했으나, SRS 미사용 팀의 요구사항 정의 방식, 기존 도구(Postman)에 대한 익숙함, AI 도구 필요성 공감 부족으로 도입에 실패.
- **극복 방안:** Postman 대비 AI 도구의 효율성과 일관성을 교육하고, 체험 가능한 데모 사이트 제공. SRS 외 다양한 문서 형식(Excel, Word) 지원을 위해 데이터 파서를 추가 개발.
- **배운점 & 인사이트**
 - **기술의 유용성보다 채택 가능성이 중요하다:** 기술적으로 뛰어난 솔루션이라도 사용자 입장에서 필요성과 편리성이 느껴지지 않으면 효과적으로 도입될 수 없다는 점을 깨달았습니다.
 - **사용자 중심의 개발:** 초기 설계 단계부터 최종 사용자의 요구와 사용 패턴을 깊이 이해하는 것이 성공적인 도구 도입의 핵심입니다.
 - **커뮤니케이션과 교육의 중요성:** 새로운 도구의 도입 과정에서 지속적인 설득과 교육이 필수적임을 배웠습니다.

LLM을 활용한 테스트 데이터 생성 최적화 경험

- **초기 문제점:** 필드별 LLM 호출로 인해 비용 과다와 속도 저하 발생.
- **해결 방안:** LLM으로 데이터 구조 분석 후 데이터 생성이 가능한 Python 코드를 생성, 초기 1회 LLM 호출 뒤 이후 LLM 호출 없이 데이터를 무한 확장 가능하도록 설계. faker 모듈을 활용해 실제와 유사한 고품질 데이터 생성.
- **성과:** 1,000,000건의 mock 데이터를 72.38초 만에 생성, 데이터 생성 비용과 속도를 획기적으로 최적화.

플랫폼 장애 해결 및 성능 최적화 경험

파일 변경 사항을 감지하고 자동으로 RAG를 위한 임베딩 처리하는 시스템에서 메모리 릭으로 AWS 서버가 반복적으로 다운되는 문제를 해결했습니다.

- **문제 상황:** 크로마DB 인스턴스가 비정상적으로 남아있어 임베딩 작업 시 메모리 사용량이 증가, 서버 다운 발생.
- **해결 방안:** Python 프로파일링으로 원인 분석 후, 비동기 락을 사용해 동시성을 제어하고, 청크 임베딩 작업마다 불필요한 참조 해제 로직 추가.
- **성과:** 메모리 릭 문제를 완전히 해결하고 안정적인 플랫폼 운영 환경 확보.

▶ LLM 기술에 대한 비전

코딩 자체는 LLM이 대체할 가능성이 높지만, 요구사항 분석과 효율적인 설계 능력은 여전히 인간의 역할로 중요할 것입니다. LLM의 잠재력을 최대한 활용하기 위해서는 정확한 설계를 기반으로 효과적인 프롬프트 작성 능력이 필수적이라고 생각합니다. 이를 위해 지속적으로 시스템 설계와 패턴에 대한 학습을 이어가고 있습니다.

또한, Agent 중심의 애플리케이션 개발이 주류가 될 것으로 보며, LLM의 독립적인 판단과 개선 능력을 활용한 Agent 기반 자동화가 미래의 핵심이 될 것이라 믿습니다. AI 기반 업무 자동화가 확산되면서 도메인 지식과 AI 활용 기술의 결합이 더욱 중요해질 것입니다. 이를 뒷받침하기 위해 커뮤니케이션 능력과 문서화 역량도 개발자의 핵심 역량으로 자리 잡을 것이라고 생각합니다.

궁극적으로, LLM 기술을 통해 효율적인 자동화와 정확한 설계를 실현하며, 다양한 비즈니스와 산업의 발전에 기여하는 것을 목표로 삼고 있습니다.

🎓 Education

경희대학교

국어국문학과 학사, 문화관광콘텐츠학과 복수전공

2013.03 ~ 2018.02

🏆 Personal Activities

AWS GameDay 참여

2024.12.20

- **개요:** AWS 클라우드 서비스를 활용한 문제 해결 실습 이벤트.
- **성과:**
 - Storage 부문 1등 달성.
 - AWS 서비스 활용 능력 향상 및 협업과 AI 활용 문제 해결 능력 강화.

🚀 Projects

OiBDD : AI 기반 BDD 자동화 솔루션

- **기간:** 2024/08 - 2024/12 (4M)
- **프로그램 개요:** BDD의 디스커버리 워크숍 과정을 AI로 자동화하여 지원하는 프로그램

- 주요 기능:
 - AI를 활용하여 SRS(사용자 요구사항 정의서)를 분석하고, 이를 기반으로 모든 경우의 테스트 케이스를 자동 생성합니다.
 - 생성된 테스트 케이스는 AI를 활용하여 cucumber 테스트 프로그램 코드(Feature, Step Definition)로 생성하여, 실제 테스트를 위해 바로 사용할 수 있습니다.
- 기술 스택: `LangChain`, `Python`, `Flask`, `FastAPI`, `PostgreSQL`, `React`, `CICD`, `Docker`, `AWS E2C`
- 주요 성과:
 - API 코어 서버와 클라이언트 서버를 분리 : 다양한 클라이언트(웹, 플러그인) 지원 가능한 확장성 증대, 코어 기능(프롬프트, 토큰)을 보호
 - AI 응답의 품질을 높이기 위해 Memory와 RAG 기술을 적용하여 보다 정확하고 일관된 결과를 제공
 - 버전 및 용도에 따른 프롬프트 관리 설계를 통해 UI에서의 유연한 프롬프트 수정 및 적용이 가능하도록 구현
 - FastAPI의 Background Task 기능을 활용하여 LLM의 응답 대기 중에도 실시간으로 진행 상태를 조회
 - 파일 변경 시 자동으로 임베딩 시스템을 업데이트하여 최신 상태를 유지
 - Docker 기반의 Gitlab CICD 배포 시스템을 구축하여 AWS E2C 인스턴스에 서버를 안정적으로 구성

Test Data Generator : AI를 활용하여 고품질의 테스트 데이터를 신속하게 생성하는 프로그램

- 기간: 2024/07 - 2024/08 (1.5M)
- 프로그램 개요: AI를 활용하여 다양한 형식의 테스트 데이터를 자동으로 생성하는 솔루션입니다. 데이터 구조와 관계를 인식하여 사용자가 정의한 자연어 규칙에 따라 mock 데이터를 생성하며, 다양한 출력 형식으로 제공하여 테스트 데이터 생성의 효율성을 극대화합니다.
- 주요 기능
 - 데이터 구조 인식: SQL DDL 및 XML 파일을 통해 데이터 구조와 타입, 관계 정보를 자동으로 인식합니다.
 - 자연어 기반 데이터 생성 규칙: 사용자가 자연어로 작성한 규칙을 통해 mock 데이터를 생성하며, 다양한 출력 형식(csv, json, sql, DB에 직접 insert)으로 데이터를 제공합니다.
- 기술 스택: `LangChain`, `Prompt Engineering`, `Python`, `Flask`, `jquery`
- 주요 성과
 - 빠르고 저렴한 테스트 데이터 생성: 데이터 생성 규칙을 기반으로 무한의 데이터를 생성할 수 있는 Python 코드를 AI를 통해 자동 생성하여, AI 호출 비용을 최소화하고 빠른 속도로 대량의 데이터를 생성합니다. LLM 사용 비용이 최초 1번만 소요되고, AI보다 프로그램 수행 속도가 훨씬 빠르기 때문에 매우 빠르게 대량의 데이터를 생성할 수 있습니다.
 - 코드 생성 템플릿 최적화: LLM을 활용한 Python 코드 생성 시, 사전에 준비된 모듈과 테스트 데이터 규칙을 적용하여 코드 템플릿을 최적화함으로써 LLM의 할루시네이션으로 인한 오류 발생을 최소화하고 코드의 안정성을 강화하였습니다.
 - 고성능 데이터 처리: 테이블 1건(컬럼 수:22개)에 대해 1,000,000건의 mock data를 72.38초만에 생성할 수 있는 성능을 제공합니다.

사내 LangChain 강의 진행

- 기간: 2024/06 - 2024/07 (1.5M, 10회)

- 강의 내용: LangChain 기초 및 응용 스터디 자료를 준비하고 강의를 진행하여, 사내 AI 활용 역량을 강화
- 주요 성과
 - 직원들이 AI를 회사 사업에 창의적으로 적용할 수 있는 아이디어를 발굴
 - 강의 내용을 바탕으로 실제 개발 프로젝트(상품 팩토리 AI, OiBDD)로 이어짐

CS Portal : MA(유지보수 계약) 고객사를 위한 이슈 트래킹 웹 사이트

- 기간: 2024/01 - 2024/04 (4M)
- 프로그램 개요:
CS Portal은 MA(유지보수 계약) 고객사를 위한 이슈 트래킹 웹 애플리케이션으로, 고객사가 자신의 이슈를 효율적으로 관리하고 신속히 해결받을 수 있도록 지원합니다.
- 주요 기능
 - GitLab과의 연동을 통해, CS Portal 사이트에서 이슈 조회, 등록, 코멘트 추가 등 다양한 기능 제공
 - 프로젝트 별로 이슈 히스토리를 중앙화하여 문제 해결 과정 추적에 용이
- 기술 스택: Java17, Spring boot3, Spring Security, Spring WebFlux, React(Material-UI), GitLab Service Desk API, Multi Module Architecture, Reactive Programming
- 주요 성과:
 - **효율적인 이슈 관리 제공:** 고객사가 CS Portal을 통해 GitLab에 직접 접근하지 않아도 손쉽게 이슈를 관리할 수 있도록 UI/UX를 개선.
 - **GitLab Service Desk 연동 구현:** 이메일을 통해 접수된 이슈를 CS Portal에서 확인 및 관리 가능하도록 API 통합 개발.
 - **프로젝트 관리 효율화:** CS Portal에서 프로젝트별 이슈를 중앙 집중식으로 관리 가능하게 하여 팀 협업 효율성 증대.
 - **멀티 모듈 설계:** 백엔드 모듈을 분리하여 유지보수성과 확장성을 강화.
 - **리액티브 프로그래밍 도입:** Spring WebFlux를 활용하여, 데이터 흐름 관리 및 비동기 작업 최적화를 통해 성능 개선.

Simulator Version 2 : 대외 연계 시뮬레이터 솔루션 버전 업 진행

- 기간: 2023/08 - 2023/12 (5M)
- 프로그램 개요: 대외 연계 인터페이스의 테스트와 시뮬레이션을 지원하는 솔루션의 추가 기능 개발
- 주요 기능:
 - 대외기관 시뮬레이션 응답의 다양화: 대외기관 시뮬레이터가 테스트 시 다양한 요청에 맞춰 다채로운 응답을 제공할 수 있도록 기능을 확장하여, 보다 현실적인 테스트 환경을 구현
 - REST 시뮬레이터: REST API를 사용하는 대상 기관의 인터페이스 관리 및 테스트 시뮬레이션 기능 제공
 - FEP 시뮬레이터: Core System에서 대외 인터페이스 시뮬레이션 거래를 직접 진행(전문 변환, HTTP 엔드포인트 제공)
- 기술 스택:
 - **Backend:** Java8, Spring Boot2
 - **Frontend:** React (Material-UI)
 - **Integration:** REST API, HTTP/HTTPS, TCP

- **Other:** 시스템 헤더 처리, 전문 변환 로직, 동기/비동기 호출 처리
- 주요 성과:
 1. **REST 시뮬레이터 기능 개발:** REST API 거래의 시뮬레이션 기능 추가, 대상 기관 및 인터페이스 정보의 생성, 변경, 삭제 기능 구현.
 2. **REST 서버 및 클라이언트 시뮬레이션:** 서버 가동 및 각 인터페이스 거래 실행 기능 개발.
 3. **FEP 시뮬레이터 확장:**
 - HTTP 엔드포인트 제공 및 통신 전문 포맷 정의.
 - 입력/출력 전문 변환 및 응답 생성 로직 구현.
 - 동기/비동기 응답 전송 기능 추가.
 4. **기존 Angular 기반 소스 코드를 React로 전환:** 기존의 Angular 기반 소스 코드를 React로 전환하여 유지보수성과 성능을 개선.
- 배운 점:
 1. 기존의 복잡하고 대규모의 소스 코드를 분석하며 코드 이해 및 유지보수 능력 향상.
 2. TCP 및 REST API 거래 처리에 대한 심층적인 이해와 개발 경험 축적.
 3. REST API HTTP 및 HTTPS 통신과 전문 변환에 대한 기술적 역량 강화.

LINE Bank Japan 코어뱅킹 프로젝트

- 기간: 2021/04 - 2022/04 (13M)
- 프로그램 개요:
 - LINE Bank Japan의 코어뱅킹 시스템 개발 프로젝트
 - 고객 업무계좌 조회, 비대면 법인고객 가입, 채널 연동 개인 고객 정보 검증 등 주요 고객 업무를 구현하고, 시스템 안정성을 보장하기 위해 자동화 통합 테스트 프로그램을 설계 및 운영하였습니다. 이를 통해 고객 서비스 품질과 개발 생산성을 동시에 향상시켰습니다.
- 기술 스택: `Java`, `Node.js`, `Oracle`, `Git`, `GitLab`, `AngularJS`, `BXM Framework`
- 주요 성과:
 1. **고객 업무 개발:**
 - 계좌 조회 서비스 구현
 - 비대면 법인고객 가입 프로세스 개발
 - 채널 연동을 통한 개인 고객 정보 검증 기능 개발
 2. **자동화 통합 테스트 프로그램 관리:**
 - 테스트 시나리오를 프로그램으로 구현하여 테스트 효율성 및 정확도 향상.
 - 매일 자동으로 통합 테스트를 진행하고 결과를 레포트로 출력, 수정된 프로그램의 영향도와 버그를 신속히 식별.
 - 테스트 프로그램 모듈화 및 공통화 작업을 통해 유지보수성 및 확장성 강화.
 3. **프로젝트 안정화 기여:**
 - 자동화 테스트 도입을 통해 주요 릴리즈 및 기능 추가 시 품질 보증을 강화.
 - 팀 간 협업 효율성 증가로 개발 속도 향상.



Core Strengths

- 끊임없는 배움을 통해 AI와 최신 기술을 실무에 적용.
- 창의적 문제 해결 능력 및 팀 협업.
- 기술적 깊이와 설계 역량을 활용하여 안정적이고 확장성 있는 시스템 구현.