



박세연

Server Developer

Birth : 1999-06-05

Phone : 010-3607-9681

Email : 751psy@gmail.com

Address : 경기도 성남시 수정구

About me

#성실 #끈기 #화합 #책임 #논리
#Java #SpringBoot #JPA
#RESTfulAPI
#MSA #JWT #Flask #MySQL

학력

- 2024.03 ~ 현재 | 한국폴리텍대학 성남캠퍼스 인공지능소프트웨어과 | 졸업 예정

보유 자격증

- 2025.06 | 정보처리산업기사 | 한국산업인력공단
- 2024.09 | SQL 개발자 | 한국데이터산업진흥원
- 2024.09 | 데이터분석 준전문가 | 한국데이터산업진흥원

수상 내역

- 2024 | 아이디어 경진대회 부문 - 동상 | 한국실천공학교육학회
- 2024 | 제 8회 벤처창업 아이템 경진대회 - 장려상 | 한국폴리텍대학

기술 스택

- Languages** : Java, Python, JavaScript, C, R
- Markup / Styling** : HTML, CSS
- Frameworks / Libraries** : Spring, Spring Boot, JPA, Flask, React, React Native, JSP, Servlet, MyBatis, jQuery
- Data / AI** : Selenium, BeautifulSoup, Machine Learning, Deep Learning, NLP, BERT, OpenCV
- Database / Infra** : MySQL, OracleDB, SQL, AWS RDS, AWS, Docker
- Architecture / Protocol** : RESTful API, MSA, WebSocket
- Develop / Test Tools** : Git, GitHub, JUnit, Postman, Swagger
- Collaboration Tools** : Slack, Notion

주요 프로젝트 이력

- 2025.06 ~ 2025.07 | MSA 기반 안면인증 & 오픈 बैं킹 시뮬레이션 가상 금융 서비스 웹 플랫폼 - “Finday” | 개인
- 2025.04 ~ 2025.07 | 머신러닝 기반 가격, 위치, 15가지 취향 기반 음식점 추천 앱 서비스 - “내돈내픽” | 팀 프로젝트 | 리더
- 2025.06 | kcElectra BERT 기반 감성분석을 활용한 성남시 음식점 분석 시각화 웹 | 개인
- 2025.06 | InceptionResNetV2 전이학습을 통한 25개 이미지 클래스 분류 | 개인
- 2025.04 | 한국어 자연어 처리(NLP) 기반 키워드 빈도수 분석 및 시각화 웹 | 개인
- 2025.01 | Java를 활용한 블록, 트랜잭션, 해시, 전자서명, PoW 기반 블록체인 구조 시뮬레이션 실습 | 개인
- 2024.11 ~ 2024.12 | IT 지식 문제 등록·공유 및 통합, 주제별 모의 시험 응시 웹 서비스 - “QuizBank” | 개인
- 2024.10 | Java, Multi Thread, Socket 기반 채팅 프로그램 | 개인

자기 소개

새로운 기술에 대한 갈증이 있지만, 기술 구조에 있어 효율적인 기술인지 끊임없이 고민하고 실전에 책임을 다하는 “**바코드형, 성장형 개발자**” 박세연 입니다.

저는 단순한 기능 구현을 넘어, 전체적인 관점에서 서비스의 요구사항과 기능 명세가 일치하는지를 확인하고, 정확한 프로세스를 거쳐 책임감 있는 개발을 중요하게 여겨왔습니다. 정제되지 않은 데이터를 구조화하고 각 기능에 맞게 가공하여 주고받는 상호작용의 흐름에서 개발의 매력을 느꼈고, 이러한 과정을 사용자 중심의 UI/UX와 연결하는 서비스 전체 흐름에 깊은 흥미를 가지고 있습니다.

대학 입학 전부터 Python을 독학하고 코딩 테스트 문제를 풀며 선행학습을 시작하였고, 입학 후에는 Java, Spring, SQL, R, 데이터 분석, CS 이론, Linux, 인공지능 이론 등 소프트웨어 전반에 대한 학습과 실습을 이어왔습니다. 교과시간 외에도 새로운 기술을 배우고 싶은 마음에 React, React Native, Spring Boot, MSA 아키텍처, AWS, Flask 기반 AI 연동, REST API 설계 등을 자율적으로 강의를 듣고 적용해보며 기술 스펙트럼을 넓혀왔습니다.

특히, 팀장으로 참여한 ‘내돈내픽’ 프로젝트에서는 WBS를 통한 일정 관리, 기획, 설계, 개발, 코드 통합까지 전 과정을 주도하며 실무적인 관점에서 서비스의 구조를 리딩한 경험이 있습니다. 또한 ‘Finday’ 프로젝트에서는 MSA 아키텍처로의 전환과 트랜잭션 처리 안정성 확보라는 도전적인 과제를 정의하고, 각 서버의 책임 분리와 통신 흐름, 트랜잭션 재처리 전략 등을 직접 설계·구현하며 문제 해결의 실전 경험을 축적했습니다.

저는 **문제를 정의하고 해결책을 스스로 찾아내는 개발자**로서, 자율과 책임을 중시하는 토스의 문화에 적합한 인재라고 확신합니다. 빠르게 시도하고 실패하며 개선하는 경험을 통해 성장했고, 토스의 DRI 문화 속에서 주도적으로 결정하고 실행하며 팀과 함께 몰입해 결과를 도출할 수 있는 개발자가 되고자 합니다.

주요 프로젝트 경험

- **MSA 기반 안면인증 & 오픈 banking 시뮬레이션 가상 금융 서비스 웹 플랫폼 - “Finday”** | 개인 | 2025-05 ~ 2025-07
 - 소개 : JWT, 안면인증 기반 Finday 금융 서비스 웹 플랫폼에 MSA 기반 오픈 banking 시뮬레이션을 적용하여 Finday 서버, 가상 금융결제원 중계 서버, 은행 서버를 분리하여 각 서버가 기대하는 API 명세를 지키며 계좌 조회, 카드 조회, 결제, 거래 내역 조회 등의 Finday 서비스 구축
 - 주요 기술 스택 : React, face-api, SpringBoot, JWT, JPA, Restful API, DeepFace, MySQL, AWS, Swagger UI
 - 문제 및 해결 과정
 - 문제 1 : 기존에 Finday 단일 서버와 단일 DB로 구성된 Monolithic 구조의 보안성과 확장성의 문제가 확인
 - 해결 1 : Finday 서버, 가상 금융결제원 중계 API 서버, 가상 은행 API 서버로 분리된 MSA 구조로 전환
 - 문제 2 : 이체 과정에서 가상 은행 API 서버 A가 출금에 실패하였을 때 B에서 입금이 되지 않도록 하는 로직 필요
 - 해결 2 : 가상 금융결제원 API 서버에서 수동 트랜잭션 로직을 구현하여 A에서 출금이 실패하면 A의 출금을 롤백
 - 문제 3 : 각 서버의 API에서 원하는 파라미터를 한눈에 확인하기 어려워 개발 과정에서 복잡성 증가
 - 해결 3 : Swagger UI 문서화를 통해, 요청/응답에 필요한 데이터 구조를 시각화
 - 문제 4 : 연동된 은행에 대한 계좌 조회 요청을 순차적(for문 기반)으로 처리할 경우, 전체 응답 속도가 느려짐 뿐만 아니라, 한 은행 응답이 지연되면 전체 조회 응답도 늦어짐

- 해결 4 : `parallelStream()`을 활용한 병렬 처리로 모든 은행에 대한 API 호출을 동시에 수행하여 응답 속도를 향상시키고, 예외 발생 시 빈 리스트로 대체하여 시스템의 안정성과 확장성을 강화함.

- 문제 5 : 금융 서비스의 특성상 로그인 + JWT만으로는 본인 확인에 한계가 있어 보안 강화가 필요함

- 해결 5 : 얼굴 인식 기반의 2단계 인증(Flask + DeepFace)을 도입하여 사용자 인증 신뢰도와 보안성 향상

○ [GitHub Repository](#)

● 머신러닝 기반 가격, 위치, 15가지 취향 기반 음식점 추천 앱 서비스 - “내돈내픽” | 팀장 | 2025-04 ~ 2025-07

- 소개 : 머신러닝 기반 가격, 위치, 15가지 취향에 기반하여 음식점을 추천하며 다양한 서비스를 이용할 수 있는 앱 서비스
- 기여 : 팀원 역량에 맞춘 역할 분배 및 기획과 설계에 필요한 문서 작성, 채팅 서비스 개발, 머신러닝 모델 학습 및 Flask 서버에 배포, 팀원 전체 개발 과정 참여, 깃허브 코드 통합
- 주요 기술 스택 : React Native, SpringBoot, JPA, Flask, RESTful API, Machine Learning, MySQL, AWS RDS
- 문제 및 해결 과정

- 문제 1 : 사용자 현재 위치(좌표)에 기반하여, 음식점과의 거리를 계산 방법 고민
(Open API를 사용하여 거리 결과를 받아오는 것은 사용자 응답시간을 늦춘다고 판단)
- 해결 1 : 기존 DB에 저장되어 있는 음식점 좌표를 이용하여 Service layer에서 Haversine 공식 적용
- 문제 2 : 개발 과정에서, 팀원들의 각기 다른 개발 환경에서 어떻게 효과적으로 개발할 수 있는지 고민
- 해결 2 : 깃허브 브랜치 전략으로 코드 통합 및 AWS RDS를 이용하여 DB Migration 없이 팀원 모두가 접근 가능
- 문제 3 : 음식점 검색 시, 검색 조건에 일치하는 모든 음식점을 불러오는 과정에서 응답 시간 지연 발생
- 해결 3 : Pagenation 방법을 통해, 사용자가 스크롤을 내릴 시에 추가적으로 음식점을 불러올 수 있게 조치
- 문제 4 : 음식점 추천 모델 개발에 필요한 데이터를 만족하는 공공데이터를 구하기 어려움
- 해결 4 : 기존 존재하는 네이버 리뷰를 크롤링하고 정제하여 모델 학습에 필요한 데이터를 스스로 구축
- 문제 5 : 사용자간 채팅 서비스에 AI를 어떻게 접목시킬까에 대한 고민
- 해결 5 : 감성분석(부정, 중립, 긍정 분류) 모델을 활용하여 사용자가 /음식점명을 검색한 채팅에 대해서만 감성분석을 진행하고 이달의 음식점 랭킹 시각화

○ [GitHub Repository](#)

● kcElectra BERT 기반 감성분석을 활용한 성남시 음식점 분석 시각화 웹 | 개인 | 2025.06

- 소개 : 네이버 지도에서 수집한 성남시 음식점 리뷰를 기반으로 딥러닝 인공지능 기술을 이용한 감성 분석을 수행하고 시각적으로 분석 결과를 확인할 수 있는 Streamlit 기반 웹 애플리케이션으로, GPT-4 API를 통해 리뷰 기반 장점/개선점 자동 요약 제공하여 지역 사용자와 식당 운영자 모두에게 인사이트 제공 가능
- 주요 기술 스택 : Python, Streamlit, Selenium, BERT, OpenAI API, PieChart, WordCloud
- 문제 및 해결 과정
 - 문제 1 : Streamlit 배포 시 검색한 음식점에 해당 되는 리뷰를 크롤링을 하는 별도의 서버가 필요함

- 해결 1 : 사전에 모든 음식점의 리뷰를 크롤링을 한 후 csv로 저장하여 사용자가 음식점 입력 시, 해당 음식점의 리뷰들을 활용하여 감성 분석 진행
- 문제 2 : 기존 LSTM 기반 감성 분석 모델은 긍정 클래스에서 높은 정확도를 보였으나, 중립 및 부정 클래스에 대한 재현율이 낮아 전체적인 균형 성능이 떨어지는 한계
- 해결 2 : 한국어 댓글에 특화된 사전학습 언어모델인 beomi/KcELECTRA-base-v2022를 파인튜닝하여 감성 분류 모델을 개선
 - LSTM 모델 대비 부정 클래스의 F1-score가 0.56 → 0.78로 상승, 중립 클래스의 F1-score가 0.28 → 0.60으로 2배 이상 향상, 긍정 클래스는 기존의 높은 성능을 유지하면서도 균형 개선
 - macro average 기준 F1-score가 0.59 → 0.70으로 향상 → 모든 감성 클래스에 대해 안정적인 성능을 확보

○ [GitHub Repository](#)

● InceptionResNetV2 전이학습을 통한 25개 이미지 클래스 분류 | 개인 | 2025.06

- 소개 : InceptionResNetV2 전이학습을 활용하여, 직접 수집한 총 25개의 일상 사물 및 자동차 이미지를 분류하는 이미지 분류 모델 구축, 검증 정확도 97% 이상 달성
- 주요 기술 스택 : Python, Steamlit, InceptionResNetV2, Tensorflow, Numpy, Pandas, Scikit-learn, Seaborn, Matplotlib
- 문제 및 해결 과정
 - 문제 1 : 사전학습 모델을 그대로 사용할 경우 클래스 수(25개) 미일치 및 과적합 발생 가능
 - 해결 1 : include_top=False 설정으로 기존 출력층 제거 후 GlobalAveragePooling → Dropout → Dense(1024) → Dropout → Dense(25, softmax) 로 사용자 정의 출력층 구성
 - 문제 2 : 모델 구조 및 하이퍼파라미터 설정에 따라 성능이 크게 달라졌고, 학습 정확도는 높지만 검증 정확도는 낮은 과적합 상황이 반복
 - 해결 2 : 다양한 하이퍼파라미터 조합 실험을 위해 약 10개의 Jupyter 노트북을 병렬 구성하여 Dropout 비율, learning rate, Dense 레이어 구조를 반복적으로 조정하고, val_accuracy 및 loss 곡선 비교를 통해 최적 조합을 도출

○ [GitHub Repository](#)

● 한국어 자연어 처리(NLP) 기반 키워드 빈도수 분석 및 시각화 웹 | 개인 | 2025.04

- 소개 : 한국어 뉴스 기사 데이터(csv)를 기반으로 핵심 키워드를 추출하고 시각화하여 Insight 발굴에 도움을 주는 Streamlit 기반 웹 애플리케이션
- 주요 기술 스택 : Python, Streamlit, konlpy, Matplotlib, WordCloud, Pandas, Counter
- 문제 및 해결 과정
 - 문제 1 : 뉴스 기사 데이터를 토큰화시, 모든 품사에 대해서 토큰화가 이루어지므로 핵심 키워드를 파악하기 어려움
 - 해결 1 : 품사 태깅으로 'Noun' 으로 지정된 명사의 키워드만 추출할 수 있도록 함

- 문제 2 : 명사 품사만 추출하였지만 “등” 이나 “수” 등 불필요하지만 빈도수가 명사가 존재하여 시각화로 인한 분석이 어려워지는 문제
- 해결 2 : 불용어 사전을 제작하여 시각화 결과에 영향을 줄 수 있는 단어를 제거

○ [GitHub Repository](#)

● Java를 활용한 블록, 트랜잭션, 해시, 전자서명, PoW 기반 블록체인 구조 시뮬레이션 실습 | 개인 | 2025.01

- 소개 : Java를 활용하여 블록, 트랜잭션, 해시, 작업증명(PoW) 등의 블록체인 구조를 직접 구현한 시뮬레이션 프로젝트로, 암호화폐의 핵심 기술인 블록체인의 원리를 실습을 통해 체득하기 위한 목적
- 주요 기술 스택 : Java, SHA-256, Java Security API, ArrayList, PoW 알고리즘
- 문제 및 해결 과정
 - 문제 1 : 단순 블록 객체 구현을 넘어 실제 블록체인 구조처럼 블록 간 연결과 해시를 설계해야 함
 - 해결 1 : Block 클래스에서 previousBlockHash를 참조하여 체인 구조를 구현하고, 블록의 해시는 이전 블록과 트랜잭션 기반으로 생성
 - 문제 2 : Java에서 SHA-256 해시를 구현하기 위한 암호화 알고리즘 사용법이 어려웠음
 - 해결 2 : MessageDigest API를 활용하여 입력 문자열에 대해 SHA-256 해시를 계산하고, 블록 해시 생성에 활용
 - 문제 3 : PoW 방식의 채굴 조건(예: 해시 앞자리 0000)을 만족시키는 로직이 필요함
 - 해결 3 : nonce 값을 1씩 증가시키며 반복적으로 해시 계산을 수행하는 mine() 함수를 구현하여 채굴 성공 시점 감지
 - 문제 4 : 트랜잭션이 블록에 포함되었을 때 블록의 해시가 적절히 반영되도록 설계 필요
 - 해결 4 : Transaction 객체 리스트를 블록에 포함시키고, 거래 정보들을 문자열로 합쳐 해시 생성에 반영하여 변경 감지 가능하도록 구성

○ [GitHub Repository](#)

● IT 지식 문제 등록·공유 및 통합, 주제별 모의 시험 응시 웹 서비스 - “QuizBank” | 개인 | 2024.11 ~ 2024.12

- 소개 : 콘솔 기반으로 제작했던 QuizBank v1을 웹으로 확장하여, 사용자들이 객관식·주관식 문제를 등록·공유하고, 과목별 또는 전체 랜덤 시험을 응시할 수 있는 문제은행 기반 웹 서비스. 시험 결과 확인, 사용자 참여 이벤트, 크레딧 제공 기능을 포함함. 시험 기간 중 학습과 실습을 병행하기 위해 설계됨.
- 주요 기술 스택 : Java, JSP, JSTL, Servlet, HTML/CSS, JavaScript, JQuery, MyBatis, OracleDB
- 문제 및 해결 과정
 - 문제 1 : 댓글 등록/수정/삭제 시 전체 페이지가 새로고침되어 사용자 경험이 저하됨
 - 해결 1 : AJAX 기반 비동기 처리 도입으로 댓글 섹션만 갱신되도록 구현
 - 문제 2 : 객관식 문제에서 사용자가 정답을 외워 대비하는 비효율성 존재
 - 해결 2 : Service 레이어에서 정답 옵션의 순서를 무작위로 섞어 전달, 시험 응답의 예측 가능성 완화

○ [GitHub Repository](#)

● Java, Multi Thread, Socket 기반 채팅 프로그램 | 개인 | 2024.10

- 소개 : 자바 기반의 소켓 통신과 멀티스레드 구조를 활용하여 구현한 실시간 채팅 프로그램으로, 닉네임 변경, 멘션, 이미지 전송, 사용자 목록 확인, 대화 로그 저장 등 다양한 채팅 기능을 명령어 기반으로 제공. 저수준 네트워크 통신과 예외 처리, 사용자 인터페이스 설계 등을 직접 구현
- 주요 기술 스택 : Java, Socket, Multi-threading, Stream IO, Hashtable
- 문제 및 해결 과정
 - 문제 1 : 닉네임 중복 체크를 위해 2차원 리스트를 사용했으나, 사용자 수가 늘어날수록 성능 저하 발생
 - 해결 1 : Hashtable<String, ClientHandler> 구조를 활용하여 사용자 닉네임을 key로 관리함으로써 중복 확인을 상수 시간으로 처리
 - 문제 2 : 사용자가 /rename, /to, /option 등 다양한 명령어를 입력했을 때, 이를 처리하는 서비스 로직이 길고 복잡해져 코드 가독성이 떨어짐
 - 해결 2 : 각 명령어를 구분할 수 있는 시작 문자열을 기준으로 분기하고 기능별로 별도 처리 함수(닉네임 변경, 귓속말, 명령어 가이드 확인 등)로 분리하여 명확하고 유지보수하기 쉬운 구조로 개선
- [GitHub Repository](#)

개발자로서의 성장 노력

저는 IT 직무에서 가장 중요한 자질은 끊임없는 학습에 대한 갈증과 스스로 성장해 나가는 힘이라고 믿습니다.

기술은 빠르게 변화하고, 그 속도에 발맞추기 위해서는 개발자 스스로 능동적으로 배우고 실행하는 태도는 필수라고 생각합니다.

2년제 대학이라는 제한된 교육 환경 속에서도 한계를 극복하고자, 저는 학과 커리큘럼 외의 시간에도 자율적으로 학습하고 새로운 기술을 실전에 적용해 왔습니다.

React, React Native, Spring Framework, AWS, MySQL, Flask 등 핵심 기술들을 선행 학습하고, 이를 실제 프로젝트에 적극 반영하여 실무 중심의 감각을 키웠습니다. 또한 프레임워크나 문법에만 의존하지 않고, 네트워크·운영체제·데이터베이스 등 컴퓨터 공학의 기초 CS 지식도 병행하며 개발자로서의 필수 역량을 굳혀 나아갔습니다.

그러나 저는 새로운 기술을 무조건적으로 도입하지 않습니다.

기술은 수단일 뿐, 문제 해결이 본질이라는 점을 잊지 않기 위해, 신기술을 접할 때마다 “이 기술이 우리 서비스에 정말 적합할까?”, “기존 구조보다 분명한 개선 효과가 있을까?”를 스스로 계속 생각하는 것 같습니다. 이러한 균형 잡힌 기술 판단력은 개발자의 본질적 역량이라고 생각하며 꾸준히 훈련하고 있습니다.

이와 함께, 저는 개발자로서 필요한 소프트 스킬 역시 중요하게 생각합니다.

WBS 기반 일정 관리, 요구사항 명세서, 유스케이스 및 시퀀스 다이어그램, ERD 및 아키텍처 설계 문서 등 실무에서 요구되는 문서들을 직접 작성해 왔으며, 이를 통해 서비스의 흐름과 구조를 전체적으로 이해하고 설계하는 역량을 길렀습니다.

또한 여러 프로젝트에서 팀장을 맡아 팀원 간의 역할 분담, 소통, 코드 통합, 품질 유지 등 프로젝트 전반을 리딩하며 협업과 책임을 경험해 왔습니다.

앞으로도 저는 단순한 기술 습득을 넘어, 서비스의 본질과 구조를 이해하고 문제를 정의하며, 명확한 기준으로 판단하는 백엔드 개발자로 성장하고자 합니다. 감사합니다.