

## [ 인적 사항 ]



성명 이준혁 (남)  
생년월일 1997년 12월 15일 (만 26세)  
병역사항 병장 만기 재대  
희망연봉 협의 후 결정  
입사가능일 채용 확정 후 2주 이내

## [ 직무 핵심 역량 ]

- 문제 해결 능력
- 팀워크와 커뮤니케이션 능력
- 꾸준한 열정과 동기

## [ 학력 사항 ]

2016.03 ~ 2022.02	나사렛대학교(천안)	정보통신학과	졸업
2013.03 ~ 2016.02	천안청수고등학교	이과계열	졸업

## [ 수상 내역 ]

- 나사렛대학교IT학부 학술제 금상 2022.02

## [ 경험/교육 ]

- 나사렛대학교 입학관리팀 인턴 (2023.11 ~ 2024.02)
  - 통합 관리 시스템을 사용하여 수시, 정시, 편입 및 추가 합격생들 등록 지원
- ESTsoft back-end 개발자 부트 캠프 (2023.06 ~ 2023.11)
  - 기초(GitHub 초급, HTML, CSS, JavaScript, Python, 알고리즘), Django, 심화(PostgreSQL, ERD, GitHub 중급)
  - 유튜브 클론 코딩, AI 글 자동 완성 개인 블로그, 당근 클론 코딩 개발
- 우송대학교 (2021.11 ~ 2022.04)
  - AI, Python 프로그래밍 기초, 크롤링, Python 데이터 분석 기초 수료

## [ 기타사항 ]

---

### 보유기술

OS: Windows

Language: Python, Java / 중급

Database: MySQL, MariaDB, PostgreSQL / CRUD 및 간단한 질의

Framework: Django, Streamlit, Spring / 웹앱 개발, 서버 연동

## [ 자기소개서 ]

---

### [ 주도적인 문제 해결 ]

AI 프로젝트를 진행할 때 처음 만난 팀원들과 역할을 분담해서 작업을 진행했습니다. 최종 마감일 하루 전 개발한 프로그램이 실행이 되지 않는 문제상황이 발생했습니다. 원인을 찾기 위해 팀 회의를 제안하였고 패키지과 라이브러리의 버전 충돌로 인한 의존성 문제를 확인하였습니다. .gitignore 파일을 작성하여 Git 저장소에 올라가는 파일을 관리하고 requirements.txt를 작성하여 Git 저장소 파일의 패키지와 라이브러리의 버전을 통일하여 문제 해결하였습니다. 이러한 경험으로 블로그에 개발 과정, 오류를 문서화시켜 문제 해결에 도움이 될 수 있도록 관리하고 있습니다.

### [ 지속적인 능력 향상 ]

프로젝트 이외 프로그래밍 언어와 알고리즘 공부를 하였습니다. Python, Java 언어로 알고리즘 문제 풀고 코드 가독을 하며 코드를 익혔습니다. 자료구조에 대한 이해가 향상하였고 문제 해결 능력과 효율적인 코드를 작성할 수 있었습니다. 좀 더 체계적인 교육을 통한 실무능력을 향상시키고자 백엔드 부트 캠프에서 Django, Python, RDBMS, 알고리즘 등을 학습하고 수료하였습니다. 저는 입사 시 꾸준한 자기 개발과 학습의 민첩성, 빠른 적응력의 장점을 살려 업무의 효율성을 높여 팀의 성과에 기여할 수 있을 것입니다.

### [ 업무를 수행하기 위한 노력 ]

저는 정보통신학부를 전공하고 CS 지식을 쌓았습니다. 개발자로 성장하고 실무 능력을 높이하고자 ESTsoft 백엔드 개발자 부트 캠프에 참여하여 Python, Django, PostgreSQL, 프론트 언어, AWS를 학습하고 개발하며 웹 개발 능력을 키웠습니다. AI 글 자동 완성 개인 블로그, 유튜브 클론 코딩 등 다수의 협업 프로젝트 경험도 쌓았습니다. 또한 API 관련 서적을 학습하고 SMTP와 Open API를 활용하여 AI 메일 자동완성 WebApp를 Streamlit 웹 프레임워크를 통해 개발하고 배포하였습니다. Java를 사용한 레거시 기술 기반의 웹 쇼핑몰을 개발하고 이후 Spring Boot를 학습하면서 웹 게시판으 만들고 웹 MVC, DB 접근 기술을 익혔습니다. DB는 주로 MySQL, MariaDB, PostgreSQL과 같은 오픈 데이터 베이스를 사용하였습니다. 최근에는 친구의 요청을 받아 Python으로 된 코딩을 WebApp으로 풀스택 개발하였고 사용자가 원하는 제한 조건이 충족된 로직을 가진 자판기 WebApp을 Django 웹 프레임 워크, MariaDB로 구현하여 서비스를 제공하였습니다. 비즈니스에 대한 이해와 개발 능력 향상할 수 있었던 좋은 학습의 기회였습니다. 저는 앞으로 주어질 개발 직무를 성공적으로 수행하기 위해 지속적인 개발을 하며 학습하고 있습니다.

## [ 프로젝트 경험 ]

---

### 1. [ Vending Machine WebApp 프로젝트 ]

제한 조건을 설정하고 이를 충족하는 자판기 WebApp 개발 프로젝트입니다.

Django의 MTV(Model-Template-View) 구조로 WebApp 개발을 하였습니다.

<제한 조건>

1. 현금 안될 경우 신용카드 통해 받을 수 있도록 조정
2. 천원, 오천원, 만원 가능, 오만원 불가
3. 거스름돈은 100원, 500원, 1000원 가능
4. 영수증 필요한 경우 금액 적힌 영수증 출력

- Tech Stack

Django 2.2.1, Python 3.10.9, MariaDB 10.1, Visual Studio Code 1.82.2, Figma, GitHub

- 배운 점

제한 조건을 충족하는 프로젝트를 개발하여 다양한 문제를 해결하는 솔루션을 경험하고, 이를 통해 문제 해결 능력을 향상시켰습니다.

views.py에 비즈니스 로직을 구현 하였고, Model에서는 데이터베이스의 테이블과 속성을 정의하여 필요한 데이터를 불러왔습니다.

Template은 개발자 도구를 사용하여 가시적으로 Template 구조를 보면서 화면을 구성하였습니다.

### 2. [ 국내 지역별 날씨 제공 AI WebApp ]

AI를 활용한 국내 지역별 날씨 제공 서비스 (Weather provision service by region in Korea using AI)입니다.

Weather API로 요청한 날씨에 대한 정보를 불러오고 OpenAI Chat Completions API 를 사용하여 사용자의 요청에 적절한 대답을 하도록 하였습니다. Streamlit으로 WebApp을 구현하고 사용자의 요청을 대화형으로 응답합니다.

- Tech Stack

Conda 23.7 .4, Jupyter notebook 6.5 .4, Python3.11.5, Streamlit 1.32.0, OpenAI Chat Completions API

model="gpt-3.5-turbo", Weather API

- 배운 점

두 가지의 API로 사용자의 요청에 대한 적절하고 신뢰성 높은 응답할 수 있었습니다.

Streamlit 프레임 워크 학습하고 활용해서 사용자가 이용 가능한 WebApp 서비스를 구현하였습니다.

사용자가 한글로 요청하면 지역에 대한 정보를 Weather API에 알맞게 보내기 위해 대한민국 지역의 데이터.

CSV 파일에 지역명을 한글과 영어의 키-값 쌍 데이터를 구성하였습니다.

서로 다른 API는 각각 고유한 기능과 데이터를 제공할 수 있고 두 API를 사용하면 사용자에게 더 많은 기능과 데이터를 제공할 수 있으며, 더 넓은 범위의 요구를 충족시킬 수 있다는 것을 해당 프로젝트로 직접 확인하였습니다.

### 3. [ 메일 자동완성 AI WebApp ]

사용자 요청에 따라 메일을 자동으로 완성하여 보낼 수 있는 WebApp입니다. SMTP와 OpenAI Chat Completions API를 사용하여 사용자의 요청에 따라 수신자 및 메일의 내용을 작성하여 전달합니다. WebApp 상에서 address\_book.json(주소록) CRUD를 제공하여 수신자의 주소 정보 관리가 편리하도록 하였습니다.

- Tech Stack

Python3.11.5, IMAP/SMTP, Conda 23.7.4, Jupyter notebook 6.5.4, Streamlit 1.32.0, OpenAI Chat Completions API model="gpt-3.5-turbo"

- 배운 점

간이 전자 우편 전송 프로토콜인 SMTP와 Open Chat Completions API에 대해서 학습하였고 Streamlit 프레임워크로 WebApp을 생성하여 해당 기능을 사용자가 사용할 수 있도록 만들었습니다.

address\_book.json(주소록) CRUD를 제공하여 사용자가 수신자의 주소 정보를 관리할 수 있도록 해보았습니다.

사용자에게 받은 요청을 Open Chat Completions API와 IMAP/SMTP로 처리하는 로직을 각 함수로 작성하여 구현하였습니다. ex) smtp\_setting\_login, send\_email, run\_conversation\_for\_email, load\_address\_book, save\_address\_book, main

IMAP/SMTP 조합의 메일 송수신에 대한 흐름을 익힐 수 있었습니다.

[ 발신자 --SMTP--> 서버 --SMTP--> 서버(송신자) --IMAP--> 수신자 ]

### 4. [ 이미지 분류 인공지능 WebApp ]

TensorFlow/Keras에서 사전 훈련된 ResNet50 모델을 사용하여 이미지를 업로드하면 해당 이미지를 여러 범주로 분류하고 해당 범주의 신뢰도 점수를 매겨 최적의 이미지 분류하는 WebApp입니다.

Streamlit으로 WebApp 생성하여 사용자와 상호작용하여 해당 기능을 사용할 수 있도록 하였습니다.

- Tech Stack

Conda 23.7.4, Jupyter notebook 6.5.4, Streamlit 1.32.0, Ngrok, PIL, TensorFlow, Numpy

- 배운 점

Streamlit의 'file\_uploader' 함수를 사용하여 이미지 파일('.jpg' 또는 '.png')업로드 할 수 있도록 하였습니다.

업로드한 이미지는 ResNet50모델이 필요로 하는 입력 차원에 맞게 크기를 조정하였고(224x224 픽셀) 크기가 조정된 이미지는 NumPy 배열로 변환됩니다.

'decode\_predictions' 함수로 사용자가 이해할 수 있는 라벨과 신뢰도 점수를 구하였습니다.

Streamlit 프레임워크로 사용자가 사용 가능하도록 구현하였고 Ngrok을 이용하여 배포하였습니다.

## 5. [ 나이 및 성별 예측 모델(Age and prediction models) ]

OPENCV와 CNN을 이용하여 이미지, 영상 등에서 얼굴을 인식하고, 나이와 성별을 예측하도록 합니다.

kaggle의 UTK FACE DATA SET을 이용해 CNN 모델을 구축하였습니다.

- Tech Stack

Conda 23.7.4, Jupyter Notebook 6.5.4, Python3.11.5, Colab, Kaggle, Pandas, Numpy, Tensorflow

- 배운 점

데이터 파일을 pandas 라이브러리를 이용하여 데이터 프레임으로 변환하였습니다.

학습을 위한 데이터 셋을 만들기 위해서 데이터를 np.array 형태로 저장하였습니다.

나이 학습, 성별 학습을 위해 age\_model, gender\_model 두 모델을 만들어 학습하였습니다.

val loss 기준으로 다음 5회 val loss가 더 이상 줄어들지 않으면 학습을 종료하도록 설정하였습니다.

성능 저하를 보이는 과적합발생으로 학습 곡선을 확인해가며 모델을 구축하는 것이 중요함을 확인하였습니다.

모델 학습과 OPENCV를 이용하여 이미지, 영상, 실시간 캠으로 결과 값을 확인해 보니 같은 데이터로 학습을 하여도 달라지는 성능을 보았습니다.

데이터와 데이터 셋이 모델 학습에 중요한 요소이고 성능에 중요한 핵심 요소라는 것을 직접 확인하였습니다.

## 6. [ AI 프로젝트 - AI 글 자동 완성 블로그(Python) ]

Python Django를 사용하여 블로그를 개발한 팀 프로젝트입니다. 회원 비밀번호 해싱과 CSRF 방지를 적용하여 보안을 강화했습니다. 글은 조회 순서대로 배치되며 사용자의 키워드에 기반한 맞춤형 추천 글을 제공합니다.

소셜 로그인과 글 작성 시 AI를 활용한 자동 완성 기능을 API를 통해 구현했습니다.

- Tech Stack

PostgreSQL 15, Pgadmin4 7.5, Visual Studio Code 1.82.2, Django 23.2.1

- 배운 점

Django의 MTV 구조 및 보안 메커니즘(CSRF, XSS 등)을 활용한 코드를 적용했습니다.

모듈화된 애플리케이션 구조와 간편한 문법으로 코드를 구조화하여 유지 보수성을 높였습니다.

객체 관계형 데이터베이스 시스템인 PostgreSQL을 사용해 보며 RDMS와는 다른 특징을 이해했습니다.

AWS 서비스를 활용하여 프로젝트를 배포하고, requirements.txt를 통한 의존성 관리를 습득했습니다.

## 7. [ Front-end 프로젝트 - 유튜브 웹 클론 코딩 ]

HTML5, CSS3, JavaScript를 사용하여 팀 프로젝트로 유튜브 Web을 클론 코딩했습니다.

<div>태그를 활용하여 웹 페이지의 구조를 정의하였습니다.

아이콘과 이미지를 .SVG로 변환하여 용량을 최적화하고, 비디오 목록을 비동기/병렬 처리하여 전체 작업 속도를 향상시켰습니다.

- Tech Stack

HTML5, CSS3, JavaScript, Figma, Visual Studio Code 1.8, GitHub

- 배운 점

HTML5, CSS3, JavaScript를 사용하여 웹 페이지의 구조와 동작을 구현하는 경험을 쌓았습니다.

Figma와 같은 디자인 도구를 활용하여 팀원 간의 시각적 소통과 작업 효율성을 향상시키는 방법을 배웠습니다.

SVG를 활용하여 이미지를 최적화하고, 비동기/병렬 처리를 통해 성능을 개선하는 방법을 경험했습니다.

## 8. [ Back-end 프로젝트 - 과자 쇼핑몰 웹앱 (Java) ]

Java를 사용한 개인 프로젝트 과자 쇼핑몰 Webapp입니다.

반응형 웹 디자인을 적용하여 다양한 디바이스에서 편리한 이용을 가능케 했습니다.

DB를 User table과 Product table로 구성하였으며, WebServer와 DB를 연결하여 CRUD가 가능하도록 하였습니다.

- Tech Stack

Eclipse IDE 2022-12, Java, mariadb 10.11, Apache Tomcat 9.0

- 배운 점

미디어 쿼리(Media Query)를 사용하여 반응형 웹 디자인을 구현하는 방법을 배웠습니다.

ERD를 활용하여 데이터 모델의 구조를 명확하게 이해하고 관리하는 데 도움이 되었습니다.

Maria DB를 활용하여 User table과 Product table을 구성하였으며, CRUD 작업을 통해 데이터 다루는 방법을 익혔습니다.

회원 및 상품 테이블을 생성하고 CRUD 작업을 통해 문자열 데이터와 이미지 파일 저장에 대해서 배웠습니다.