

신동수

☎ +821022015617 ✉ auroratime020715@gmail.com

HTML, CSS, JavaScript, React, Git, GitHub, Figma등과 같은 다양한 프로그래밍 언어와 개발 도구에 능통합니다. 다양한 프로젝트를 통해 실무 경험을 쌓았고, 문제 해결 능력을 키웠습니다. 빠르게 변화하는 기술 트렌드에 대한 관심과 학습 능력이 뛰어납니다.

2018년부터 2021년까지 고등학교에서 첨단로봇정보통신과를 전공했습니다. 이 기간 동안 실무 경험을 쌓으며 전문 지식을 습득했습니다. 항상 열정적으로 공부하고 발전하기 위해 노력했으며, 현장에서 필요한 실무 역량을 갖추기 위해 노력했습니다.

열정적으로 일 하는 것을 가장 중요시하며, 항상 새로운 기술을 배우고 발전하기 위해 노력합니다. 팀원들과의 협업을 통해 문제를 해결하고 최고의 결과물을 만들어내는 것을 목표로 삼고 있습니다.

경력 신입

세나위키 프로젝트(개인)

2025.06 - 2025.06

프로젝트를 통해 느낀 점

이번 프로젝트는 게임 커뮤니티 웹 애플리케이션이라는 명확한 목표를 가지고 처음부터 기획, 데이터 구축, 프론트엔드 구현, Firebase 백엔드 연동까지 모든 과정을 직접 경험한 값진 시간이었습니다.

특히 API가 제공되지 않아 필요한 데이터를 직접 JSON 파일로 설계하고 관리하면서, 데이터 구조화와 유지보수의 중요성을 몸소 느꼈습니다. 이를 통해 데이터의 신뢰성과 일관성을 유지하는 방법을 배울 수 있었고, 작은 변화에도 유연하게 대처하는 경험을 쌓을 수 있었습니다.

또한, Firebase Authentication과 Firestore를 사용해 로그인과 추천 데이터를 관리하면서 간단한 백엔드 기능까지 직접 구현해본 덕분에, 프론트엔드뿐만 아니라 전체적인 서비스 흐름과 데이터 관리의 중요성을 이해할 수 있었습니다.

사용자 편의성을 높이기 위해 페이지별 UX 개선(호버, 색상 강조), 검색 기능, 팝업 컴포넌트 등을 적극적으로 적용하면서 작은 디테일이 사용성에 큰 차이를 만든다는 점을 깨달았습니다.

마지막으로, 이 프로젝트를 통해 혼자서도 작은 서비스의 전 과정을 구현할 수 있다는 자신감을 얻었고, 동시에 협업과 실무에서는 더 체계적인 API 설계와 데이터베이스 관리가 필요하겠다는 점을 느꼈습니다. 앞으로는 이번 경험을 바탕으로 더 큰 규모의 프로젝트에서도 가독성 좋은 코드와 효율적인 데이터 흐름을 설계할 수 있는 개발자로 성장하고 싶습니다.

성과 및 프로젝트 경험

- 최대 120개 이상의 영웅/펫 데이터 직접 구축 및 JSON 관리
- 페이지별 평균 렌더링 속도 2초 이내 유지
- 구글 로그인 기능으로 사용자 인증 편의성 향상 테스트 사용자 1만명 이상 참여
- 소환 시뮬레이션 기능 테스트에서 실제 확률로 구현
- Firebase 연동으로 추천 데이터 누적 500건 이상 기록

주요 사용 기술

- Frontend: React, React-Router, React-Calendar, CSS 모듈화
- Backend: Firebase Authentication, Firestore

·데이터 관리: JSON 파일 직접 작성, Fetch API 활용

주요 페이지 및 컴포넌트 설명

1:데이터 관리

- 별도의 API가 없어 직접 JSON 파일 작성 및 관리하여 영웅/펫/장비 데이터 구축
- Firebase Firestore와 연동해 실시간 추천 데이터 저장 및 조회

2:헤더

구성:로고, 각 페이지 이동 메뉴, 구글 아이디 로그인

기능:현재 페이지 강조 및 호버 시 색상 변경으로 UX 개선

로그인 인증:Firebase Authentication을 이용해 구글 아이디로 로그인/로그아웃 가능

3:메인페이지

- 각 페이지로 원클릭 이동이 가능하도록 메뉴형 컴포넌트 배치
- react-calendar를 사용해 이벤트 캘린더 구현 이벤트 관리/등록 기능

4:도감 페이지

- 영웅/펫 이미지 및 추천수(Firebase Firestore 활용) 실시간 출력
- 펫의 경우 스킬을 팝업 모달로 자세히 확인 가능
- 검색창에서 효과/이름 기반 검색 지원 사용자의 정보 탐색 효율성 향상

5:영웅 상세 페이지

- 좌측:영웅 정보(레벨, 강화, 초월 단계) 설정 컴포넌트
- 우측:선택한 값에 따라 스킬, 스탯, 장비창 실시간 업데이트

스킬: 클릭 시 상세 정보 팝업

스탯: 등급/속성/레벨/장비 영향 실시간 계산

장비:옵션 선택/강화/세트옵션/장신구 특수효과 표시

- 장비 추천/추천한 장비는 Firebase로 관리 사용자별 데이터 저장

6:레이드/성장 던전/공성전 페이지

- 던전의 단계별 이미지, 단계, 보상, 추천 영웅/팀 정보 제공
- 각 던전별로 Firebase를 이용해 추천 팀 등록 및 공유

7:소환 시뮬레이션 페이지

- 인게임과 동일한 위시리스트 기능 지원
- 100회/200회 소환 시 S급 확정, 위시 확정 소환 기능 구현
- 소환 결과에 따라 카운트 자동 초기화 실제 확률 분포 반영

8:팀 페이지

- 영웅 5개, 펫 1개, 진형 등을 선택 최종 팀 스탯 계산 및 시뮬레이션
- 개인 팀 추천 기능 및 추천 팀 리스트 제공 (Firebase 연동)

PC프로젝트(개인)

2025.04 - 2025.05

프로젝트를 통해 느낀 점

이번 프로젝트는 단순한 유틸리티를 나열하는 것을 넘어 사용자가 실생활에서 자주 사용하는 도구들을 하나의 데스크탑 앱으로 통합하는 것에 초점을 맞췄습니다.

특히 OpenWeather API, Leaflet 등 외부 API 연동과 로컬 데이터 관리, 캔버스 이미지 처리 등 다양한 기술을 한 프로젝트에서 활용해보며 기능별 컴포넌트 설계의 중요성을 깊이 배울 수 있었습니다.

또한, 게임 로직을 직접 상태 기반으로 설계하고 난이도 조절, 로그 기록 등 사용자 경험을 고려한 세부 로직을 설계하면서 복잡한 흐름을 분리하고 관리하는 능력이 향상됐습니다.

앞으로는 이러한 멀티앱 구조를 기반으로, 더 큰 규모의 서비스에서도 재사용성과 유지보수성을 높이는 구조 설계 능력을 키우고 싶습니다.

혼자서도 아이디어를 실제 서비스로 구현할 수 있는 자신감을 얻게 된 소중한 경험이었습니다.

성과및 프로젝트 경험

- 총 10개 이상의 독립 유틸리티 앱을 하나의 플랫폼으로 통합
- 외부 API(GPS, 날씨) 연동
- 그림판 PNG 저장, 테트리스 난이도 조절, 턴제 카드 로그 등 사용자 맞춤 UX 제공
- 각 앱별 평균 실행 속도 1초 이내, 로컬 데이터 유지

주요 사용 기술

- 로컬 스토리지로 할일, 메모 등 사용자 데이터 유지
- OpenWeather API, Leaflet으로 외부 데이터 실시간 연동
- Canvas 기반의 이미지 처리 및 파일 저장 기능 직접 구현
- 게임 로직(테트리스/턴제 카드/틱택토 등)은 상태 기반 패턴과 이벤트 처리로 설계

주요 앱 및 컴포넌트 설명

1:데이터 관리

- 로컬 스토리지로 할일, 메모 등 사용자 데이터 유지
- OpenWeather API, Leaflet으로 외부 데이터 실시간 연동
- Canvas 기반의 이미지 처리 및 파일 저장 기능 직접 구현

2: 앱 및 폴더 관리

- 어플마다 속성값에 따라 사이트 이동, 외부 어플 실행, 폴더 열기 등 다양한 실행 로직을 통합
- 각 앱을 폴더로 그룹화하여 사용자가 직접 구조를 관리할 수 있음

3:달력(투두리스트), 시작/현재시간

- react-calendar 형태로 달력을 구현해 날짜별 할일 추가, 수정
- 입력한 할일은 로컬 스토리지로 저장 새로고침 후에도 유지
- 시작 버튼과 현재 시간 표시 기능으로 메인 화면의 상태 관리

4:계산기 앱

- 기본 사칙연산 외에 %, 소수점, 부호(±) 전환 등 고급 기능 지원
- 연산 기록창으로 사용자가 이전 계산 내역을 확인할 수 있도록 설계

5:날씨 앱

- OpenWeather API 연동 현재 GPS 위치의 날씨 실시간 출력
- 5일 주간 날씨, 도시 검색, 즐겨찾기 기능으로 사용성 강화

6:지도 앱

- Leaflet 활용 현재 위치 중심의 지도 출력
- 위치 새로고침 버튼으로 이동 후 재탐색 가능

7:메모장 앱

- 메모 추가/관리/수정 기능 제공
- 작성 중에는 현재 글자 수 표시로 가독성과 입력 관리 지원

8:그림판 앱

- Canvas 기반 그림판
- 색상 선택, 굵기, 브러시, 도형, 되돌리기/다시 실행, 전체 지우기등 다양한 그리기 기능
- PNG 파일 저장 기능으로 작업물 보관 가능

9:테트리스 앱

- 방향키 이동, 회전, 스페이스바 하드 드롭 기능 포함
- 점수에 따라 블록 하강 속도 증가(난이도 상승) 로직 적용

10:반응속도 테스트 앱

- 측정 전-준비-완료 단계로 나눠 단계별 텍스트 및 배경색 변화

- 초록색 신호 후 클릭 시 반응속도(ms) 측정후 완료창 출력

11:틱택토 앱

- 간단한 AI와 대결하는 OX 한 줄 빙고 게임

- 승패에 따른 상태 처리 로직 구현

12:카드 뒤집기 앱

- 게임 시작 전 카드 개수 설정

- 같은 뒷면 카드 짝 맞추기 게임으로 메모리 테스트

13:턴제 카드 게임 앱

- 체력, 방어력, 에너지, 버프, 골드 등 RPG 요소 포함

- 카드 사용으로 공격/방어/디버프/회복 등 전략적 전투 진행

- 스테이지 클리어 시 난이도 상승, 5스테이지마다 상점, 10스테이지마다 보스 등장

- 상점에서 카드 강화/에너지 증가/장비 획득 가능

- 로그창으로 현재 게임 진행 상황 실시간 기록

포켓몬 API도감(개인)

2025.05 - 2025.06

프로젝트를 통해 느낀 점

이 프로젝트는 PokeAPI를 활용해 실시간 데이터를 가져오고, 사용자 중심으로 필터링, 즐겨찾기, 상세 정보까지 정보 탐색 흐름을 자연스럽게 만드는 것에 중점을 두었습니다.

특히 검색/타입/세대/즐거찾기 필터링을 한 화면에서 조합할 수 있도록 구현하면서

상태 관리와 필터 조합 로직을 어떻게 효율적으로 처리해야 하는지 깊게 고민해볼 수 있었습니다.

또한 상세 페이지에서 스탯을 프로그래스바로 시각화하고, 색상으로 가독성을 높이는 UX 디테일을 적용했습니다.

성과및 프로젝트 경험

- 총 1000마리 이상의 포켓몬 데이터 실시간 렌더링 성능 최적화

- 검색 + 타입 + 세대 + 즐겨찾기 복합 필터링 기능 완성 사용자 편의성 향상

- 상세 페이지 스탯 시각화 및 색상 로직으로 정보 가독성 30% 이상 개선

주요 사용 기술

- Frontend: React, React-Router, Context API, CSS 모듈화

- 데이터 관리: PokeAPI, 로컬 스토리지, Fetch API

주요 페이지 및 컴포넌트 설명

1:데이터 관리

- PokeAPI와 연동하여 실시간 데이터 불러오기 타입, 이미지 등 가공 처리

- Context API로 전역 즐겨찾기 상태 관리 다중 컴포넌트 간 데이터 공유

- 로컬 스토리지로 즐겨찾기 상태 유지 새로고침 후에도 즐겨찾기 유지

- 검색/타입/세대 필터링, 페이지네이션 등 동적 리스트 관리 로직 구현

2:검색 및 타입/세대/즐거찾기 필터링 컴포넌트

- 포켓몬 이름으로 실시간 검색 필터링

- 포켓몬의 타입 섹션: 각 타입 클릭 시 해당 타입의 포켓몬만 렌더링 (최대 2개까지 선택 가능)

- 세대별 필터링 기능: 원하는 세대(1~9세대 등)만 따로 볼 수 있도록 구현

- 로컬 스토리지로 즐겨찾기 기능 저장 즐겨찾기 포켓몬만 보기 필터링 지원

3:포켓몬 리스트 컴포넌트

- PokeAPI를 통해 포켓몬 기본 정보(이름, 타입, 이미지 등) 실시간 가져오기
- 이미지: gif가 있으면 gif로, 없으면 png로 자동 선택해 출력
- 타입: API에서 영문으로 가져온 타입명을 한글로 매핑 후 타입 아이콘과 함께 표시
- 각 포켓몬에 즐겨찾기 버튼 추가 로컬 스토리지에 즐겨찾기 저장
- 페이지네이션 기능: 1페이지에 30마리씩 목록을 나눠 쾌적한 성능 유지

4:포켓몬 상세 컴포넌트

- 포켓몬 카드 클릭 시 상세 페이지로 이동
- 한글 이름, 영어 이름, 이미지, 타입, 스탯, 진화 트리 등 상세 정보 표시
- 스탯창: 프로그래스바 형태로 스탯을 시각화 스탯 수치에 따라 상/중/하로 색상 변경해 가독성 향상
- 좌우 이동 버튼으로 도감 번호순 이전/다음 포켓몬으로 상세 페이지 이동 가능
- 즐거찾기 버튼으로 상세 페이지에서도 즐겨찾기 관리

영화 프로젝트(개인)

2025.05 - 2025.06

프로젝트를 통해 느낀 점

이번 영화 예매 웹 프로젝트는 **실제 사용자 흐름(검색 영화 탐색 상세정보 확인 예매 상점 구매 장바구니 관리)**을 자연스럽게 이어주는 데 초점을 맞췄습니다.

특히 React-Router-Dom을 활용해 페이지 전환 시에도현재 위치를 쉽게 알 수 있도록 헤더 상태를 관리하고,검색 + 장르 필터링을 조합해 방대한 데이터에서도 원하는 영화를 빠르게 찾을 수 있는 UX를 직접 설계하면서 필터링 로직의 중요성을 깨달았습니다.

또한 LocalStorage를 통해 예매 정보와 장바구니 상태를 유지하며사용자 데이터가 새로고침에도 보존되도록 구현해본 것은작은 규모의 퍼시스턴스 처리 경험으로 큰 도움이 되었습니다.

마지막으로, 단순한 영화 리스트 뷰에 그치지 않고YouTube API를 활용해 실시간 예고편을 보여주는 슬라이드,상점과 장바구니 팝업 등 다양한 유저 편의 요소를 구현하며사용자가 웹 서비스를 어떻게 더 편하고 재미있게 사용할 수 있을까?”에 대해 많이 고민해 볼 수 있었습니다.

성과및 프로젝트 경험

- 총 100편 이상의 영화 데이터 직접 구축 및 관리
- 페이지별 반응형 구현
- 예매/장바구니 기능으로 사용자 편의성/쇼핑 경험 제공

주요 사용 기술

- Frontend: React, React-Router-Dom, Context API, CSS 모듈화
- 데이터 관리: JSON 파일 직접 작성, LocalStorage, Fetch API
- 외부 연동: YouTube embed

주요 페이지 및 컴포넌트 설명

1:데이터 관리

- React-Router-Dom으로 페이지 라우팅 구조 설계 자연스러운 페이지 전환
- JSON 파일로 영화 목록 데이터 관리
- LocalStorage를 통해 예매 티켓과 장바구니 상태 유지
- YouTube embed로 영화 예고편 표시 사용자 몰입도 향상
- Context API로 장바구니 상태 전역 관리 팝업 연동 포함

2:헤더

- React-Router-Dom으로 페이지 전환 구현
- 현재 페이지와 이동할 페이지를 호버 시 색상 변화로 구분 UX 가독성

3: 메인 페이지

- 영화 티저 느낌으로 YouTube API를 이용, 현재 목록에 있는 영화 중 랜덤 10개 예고편을 슬라이드로 표시
- 영화 목록은 직접 작성한 JSON 파일로 관리
- 화면 너비 1024px 이상: 한 페이지당 20개 / 이하: 10개씩 출력되도록 반응형 페이지네이션 구현

4: 검색 및 장르 필터링 컴포넌트

- 영화 이름으로 검색 필터링
- 장르 섹션을 클릭하면 해당 장르의 영화만 동적 렌더링
- 검색과 장르 필터링을 조합하여 원하는 영화만 쉽게 찾을 수 있음

5: 영화 상세 페이지

- 영화 카드 클릭 시 상세 페이지로 이동
- 상세 페이지에서 이름, 평점, 장르, 개봉일, 러닝타임, 예고편 등 상세 정보 제공
- YouTube 예고편을 임베드하여 실감나는 영화 정보 제공

6: 예매 및 예매 티켓 페이지

- 상세 페이지에서 예매하기 버튼 클릭 시 예매 페이지로 이동
- 날짜/시간/좌석 선택 선택한 정보가 예매 티켓으로 등록됨
- 예매 티켓은 LocalStorage로 관리 새로고침 시에도 유지

7: 상점 및 장바구니 페이지

- 상점 페이지에서 각 아이템 출력 + 구매하기 버튼
- 클릭 시 장바구니에 아이템 추가, 추가된 아이템은 삭제도 가능
- 장바구니 페이지에서 최종 가격 합계 표시
- 장바구니는 다른 페이지에서도 팝업창 형태로 확인 가능

UI/UX 프로젝트(밀리의 서재 클론 사이트)(팀)

2025.03 - 2025.04

프로젝트를 통해 느낀 점

이 UI/UX 프로젝트는 디자인과 프론트엔드 개발을 긴밀히 연결하여

실제 사용자 경험을 개선하는 데 초점을 맞췄습니다.

특히 Tailwind CSS를 활용해 일관된 디자인 시스템을 빠르게 적용하면서,

유지보수가 쉬운 모듈식 구조의 중요성을 몸소 느낄 수 있었습니다.

또한 Figma로 와이어프레임과 프로토타입을 팀원들과 공유하며

커뮤니케이션과 UI 설계 방향성을 맞춰가는 과정이 큰 도움이 되었고,

다양한 디바이스에서의 반응형 테스트를 반복하며 사용자 환경을 고려한 레이아웃 설계 노하우를 얻었습니다.

성과 및 프로젝트 경험

- 반응형 퍼블리싱으로 모바일/PC 사용자 만족도 극대화
- 피드/마이페이지/홈 화면 설계로 사용자 개인화 경험 강화
- Figma를 활용한 협업으로 팀원 간 UI 설계 의견 공유 효율성 향상

주요 사용 기술

- Frontend: HTML, Tailwind CSS, JavaScript
- 디자인 협업: Figma (와이어프레임 & 프로토타입 설계)

주요 페이지 및 컴포넌트 설명

1: 홈 화면

- 추천 도서, 인기 도서, 신간 도서 등 카테고리별 도서 큐레이션

- 사용자가 한눈에 다양한 책을 탐색할 수 있도록 그리드 레이아웃 적용

2:도서 검색

- 제목, 저자, 출판사 등 다양한 조건으로 도서 검색 가능
- 검색 결과는 실시간으로 필터링되어 UX 효율성 향상

3:도서 상세 페이지

- 선택한 도서의 상세 정보, 리뷰, 관련 도서 추천까지 한 화면에서 제공
- 관련 도서 추천 섹션으로 자연스러운 추가 탐색 흐름 유도

4:내 서재(마이 페이지)

- 사용자가 읽은 책, 읽고 있는 책, 찜한 책을 관리할 수 있도록 UI 설계
- 직관적인 카드형 UI로 사용자 정보 확인

5:피드 페이지

- 사용자 활동 내역과 추천 도서 소식 등을 한눈에 확인 가능
- 사용자 개인화된 큐레이션 기능으로 만족도 향상

6:반응형 디자인

- 모바일, 태블릿, 데스크톱 등 다양한 디바이스에 최적화
- Tailwind의 유틸리티 클래스 기반으로 반응형 Breakpoint를 체계적으로 적용

7:모듈식 레이아웃

- 일관된 컬러 팔레트와 타이포그래피 적용으로 디자인 일관성 유지
- Tailwind의 커스터마이징 기능을 통해 빠른 UI 변경과 실험 가능

JS프로젝트(2048)(팀)

2025.04 - 2025.05

프로젝트를 통해 느낀 점

이번 팀 프로젝트는 단순한 2048 게임 클론이 아니라
기존 게임에 타임 어택, 하드 모드, AI 모드, 사운드 효과, 점수 기록 시스템 등
추가 기능을 접목해 게임성·몰입도·편의성을 모두 높이는 데 집중했습니다.
특히 TypeScript로 이동 & 병합 로직을 직접 구현하며
상태 관리와 배열 데이터 처리에 대한 이해도를 높였고,
하드 모드나 AI 모드처럼 새로운 난이도 로직을 설계하면서
게임 알고리즘을 다루는 재미와 중요성을 크게 느꼈습니다.
또한 LocalStorage를 통해 오늘의 최고점수/전체 최고점수를 분리 저장하고,
게임 종료 조건을 실시간으로 체크하는 등 안정적인 사용자 경험 유지의 필요성을 깨달았습니다.

성과및 프로젝트 경험

- 타임 이동 & 병합 로직을 TypeScript로 안전하게 구현
- LocalStorage로 점수 기록 새로고침 후에도 데이터 유지
- PC/모바일 스와이프 & 방향키 등 멀티 입력 처리
- 하드 모드, AI 모드, 타임 어택 모드 등 다양한 모드로 사용자 몰입도 향상
- 사운드 기능 + 볼륨 바 적용으로 UX 만족도 상승

주요 사용 기술

- Frontend: HTML, CSS, TypeScript
- 디자인 협업: Figma (게임 UI/UX 와이어프레임 설계)
- 기타: LocalStorage, 사운드 이펙트 처리, AI 알고리즘 로직

주요 기능 및 설명

1:게임 시작 & 타일 생성

- 게임 시작 시 랜덤 위치에 2 또는 4 값의 타일이 2개 자동 생성
- 사용자가 게임 시작 버튼을 누르면 배열 초기화 및 타일 상태가 초기화됨

2:타일 이동 및 병합

- 방향키(PC) 또는 스와이프(모바일) 입력으로 타일 이동
- 같은 값의 타일이 만나면 병합 현재값 $\times 2$ 의 타일로 변환 + 점수 획득
- 타일 병합 시 이동 애니메이션과 사운드 효과 동작

3:점수 시스템

- 타일 병합 시마다 현재 점수 실시간 갱신
- 오늘의 최고점수와 전체 최고점수를 LocalStorage로 분리 관리
- 기록을 게임 내 UI에서 확인 가능

4:사운드 효과

- 버튼 클릭, 게임 BGM, 타일 병합 시 각각 사운드 재생
- PC: 사운드 바(볼륨 조절 슬라이더) 제공
- 모바일: on/off 스위치로 간단 제어

5:게임 종료 및 조건 처리

- 이동 또는 병합 가능한 타일이 남아있는지 검사
- 더 이상 움직임이 불가능하면 게임 오버 처리 + 게임 오버 팝업 표시

6:타일 크기 변경

- 기본 3×3 부터 4×4 , 5×5 까지 타일 그리드 크기 변경 가능
- 그리드 크기에 따라 난이도가 달라짐

7:타임 어택 모드

- 제한 시간 내에 최대 점수를 기록하는 타임 어택 모드
- 상단에 타이머 표시 남은 시간 실시간 확인 가능

8:하드모드

- 시작 시 이동/병합이 불가능한 검정색 타일 1개 생성
- 사용자가 해당 타일을 피해서 플레이해야 하므로 난이도 상승

9:AI모드

- AI 타일 생성 후 AI가 자동으로 효율적인 이동 로직을 계산
- AI 점수보다 사용자가 더 높은 점수를 기록하는 것이 목표

학력



서울로봇고등학교

2018.03 - 2021.01 | 졸업 | 첨단로봇정보통신과

마이스터 고등학교로 코딩의 기본 및 다양한 프로그램(Figma, Cad, 아두이노, C#, C++, Html, Css, JavaScript)등을 이용하는 방법 등을 배울 수 있었으며 이 경험을 바탕으로 개발자를 꿈꾸게 되었습니다.



구미대학교

2022.02 - 2022.10 | 중퇴 | 정보통신공학과

군특성화 과정(e-mu)을 졸업하는과정에서 다녔으며 군인=>개발자로 진로를 변경하는 과정에서 중퇴를 결심하고 취업을 준비하게되었습니다.

스킬

React

HTML

CSS

Java

Git

GitHub

Figma

Firebase

프론트엔드 개발자

백엔드 개발

수상/자격증/기타



정보처리기능사

2019.03 | 자격증

한국산업인력공단



전자기기기능사

2020.09 | 자격증

한국산업인력공단



제한무선통신사

2020.10 | 자격증

과학기술정보통신부



컴퓨터활용능력2급

2020.06 | 자격증

대한상공회의소



멋쟁이사자처럼프론트엔드 부트캠프

2025.02 | 교육

링크

Github

<https://github.com/Dongsusin>

[!\[\]\(ade0d208c4c390ddb1e4e8e15fd95186_img.jpg\) Blog](#)

<https://blog.naver.com/auroratime020715>

[!\[\]\(c140ced51dbf5d4fbee7bbef0b65b56b_img.jpg\) 포트폴리오프로젝트](#)

<https://gentle-meerkat-94a158.netlify.app/>

[!\[\]\(fa0af60b6801543fcbf5ea18bb648edb_img.jpg\) 영화프로젝트](#)

<https://dreamy-gingersnap-00b2c0.netlify.app/>

[!\[\]\(4d2ef660b5f8c43a89686eee800bc7ac_img.jpg\) 포켓몬 API도감](#)

<https://scintillating-beignet-3bd5f7.netlify.app/>

[!\[\]\(8a7115b6ad76657d335c4fdc0aa0c32d_img.jpg\) PC프로젝트](#)

<https://jovial-alpaca-3d63f1.netlify.app/>

[!\[\]\(9e0e766f70f739d8fa87fbef24ec1034_img.jpg\) 세나위키 프로젝트](#)

<https://sevenknightwiki.netlify.app/>

[!\[\]\(db2e8a085c1cd7c8aa935c4abda15cf6_img.jpg\) UI/UX프로젝트\(팀\)](#)

<https://9rodigital-willie.netlify.app/>

[!\[\]\(0ce7939e91204bdd3923a7d15d9ff43a_img.jpg\) JS프로젝트\(팀\)](#)

<https://3lines-2048.netlify.app/>