남승현

Web Developer

nsh_823@naver.com

010-9823-0189

서울, 대한민국

https://github.com/nsh0823

도전을 즐기며 끝까지 문제를 해결하는 끈기 있는 개발자입니다. 빠른 학습력으로 새로운 기술에 적응하고, 적극적인 소통과 협업을 통해 복잡한 문제를 구조적으로 개선하며 꾸준히 성장하고 있습니다.

교육

네이버 부스트캠프 웹 풀스택 10기 | 네이버 커넥트제단 2025.06 ~ 2026.02

- AI 시대 개발자로서 문제를 정의하고 해결 방향을 설계하는 역량 중심으로 학습
- 컴퓨팅적 사고와 CS 기초를 바탕으로 문제 분석·설계 및 해결 능력 강화
- 운영체제, 자료구조, 네트워크, DB, 객체지향 등 핵심 CS 전반을 실습 기반으로 학습하고 프로젝트에 적용
- 클라이언트–서버 전반의 동작 원리를 이해하며 풀스택 개발 역량 확보 및 성능/구조 개선 경험
- 스크럼, 페어 프로그래밍, 코드 리뷰, 백로그 관리 등 협업 기반 개발 프로세스를 실전 프로젝트에서 경험
- 다양한 동료와의 지속적인 소통과 지식 공유를 통해 문제 해결 능력을 강화하고, 기획–테스트–배포까지 전 과정 협업 경험 축적

인공지능 응용 SW 개발 및 데이터 분석 과정 | 엔코아 플레이데이터 2021.02 ~ 2021.08

- Java, Spring Boot 기반의 웹 애플리케이션 구조 설계 및 RDBMS 연동 학습
- Python을 이용한 데이터 전처리, 시각화 및 머신러닝/딥러닝 모델 개발 과정 수료
- 웹 기술과 AI 모델을 결합한 지무 중심의 프로젝트 수행 (글쓰기 보조 및 주문 시스템 등)

프로젝트

Funda | CS 지식 공부를 게임처럼 재미있게 학습할 수 있는 플랫폼 2025.12 ~ 2026.02

사이트 주소: https://www.funda.website/

프로젝트 Repository: https://github.com/boostcampwm2025/web21-funda

주요 기여 부분:

- **첫 진입 경험 개선**: 랜딩/메인 페이지 로딩 경험, 인증 상태 flicker 제거, 스크롤/반응형 개선
- **학습 동기화 UX**: 로드맵 진행률/정답률, 스텝 완료 표시, 비로그인 unlock 로직, 토스트 피드백
- **퀴즈 상호작용/피드백**: 애니메이션, 사운드 효과음, 코드 블록 렌더링 가독성 개선
- **운영 도구 사용성**: 관리자 퀴즈 업로드 다중 파일 지원, 로드맵/사이드바 UI 개선
- **실시간 대결 경험**: 배틀 시작/배틀 퀴즈 UI/로직 안정화, disconnect 로직, countdown 시간 동기화

주요 문제 해결 경험:

1. **인증 상태 준비 이전 렌더링으로 인한 UI Flicker 구조적 해결**
   - 문제: 로그인 상태임에도 초기 진입 시 비로그인 UI가 잠시 노출되는 UX 결함 발생
   - 원인 정의: 인증 상태가 준비되기 전에 컴포넌트가 먼저 렌더링되는 구조적 문제
   - 해결:
     - 인증 서버 상태를 TanStack Query로 일원화하여 상태 책임 분리
     - useSuspenseQuery 기반으로 데이터 준비 시점까지 렌더링 지연
     - isAuthReady 가드 설계로 렌더링 조건 명확화
   - 결과:
     - 초기 UI Flicker 제거
     - 인증 흐름 안정화 및 불필요한 리렌더링 감소
     - 서버 상태와 UI 상태의 책임 경계 명확화

2. **6,000문제 생성 자동화 파이프라인 설계 및 안정성 확보**
   - 문제: AI 기반 문제를 수동 생성·검수 시 약 60시간 이상 소요되는 병목 발생
   - 원인 정의: 반복 작업 중심의 비확장적 구조와 오류 발생 시 전체 중단되는 워크플로우
   - 해결:
     - n8n 기반 메인-서브 계층형 워크플로우 설계
     - JSON Schema 기반 Structured Output 검증 도입
     - 5개 Unit 단위 중간 저장 분기 구성으로 부분 실패 허용 구조 구현
     - malformed function call 대응 및 재실행 가능 구조 설계
   - 결과:
     - 750문제 30분 생성, 6,000문제 약 4시간 내 제작 가능
     - 오류 발생 시 전체 중단 없는 안정적 파이프라인 구축
     - 콘텐츠 제작 병목 제거 및 팀 생산성 대폭 향상

3. **배포 환경에서 발생한 실시간 카운트다운 시간 불일치 문제 해결**
   - 문제: Socket.io 기반 배틀 카운트다운이 배포 환경에서 클라이언트마다 다르게 표시되는 현상 발생
   - 원인 정의: 클라이언트 로컬 시간을 기준으로 타이머를 계산하여 네트워크 latency와 기기 시간 오차가 반영되는 구조
   - 해결:
     - 서버 시간을 함께 emit하여 단일 시간 기준(Source of Truth) 설정
     - 클라이언트에서 serverTime - Date.now() 기반 offset 계산
     - 보정된 시간(Date.now() + offset)을 기준으로 카운트다운 수행
   - 결과:
     - 클라이언트 간 시간 오차 제거
     - 배틀 종료 시점 및 점수 반영 타이밍 동기화 안정화
     - 배포 환경에서도 일관된 실시간 사용자 경험 확보

ONEGO | 글쓰기를 쉽게할 수 있도록 도와주는 블로그 플랫폼 2021.07 ~ 2021.08

사이트 주소: https://www.onego.qzz.io/

프로젝트 Repository: https://github.com/nsh0823/onego

주요 기여 부분:

- 헤더/푸터/사이드바, 검색/검색 결과, 계정설정, 프로필 수정, 비밀번호 변경, 내프로필, 저장글/발행글, 글쓰기 화면 구현

주요 문제 해결 경험:

1. **대규모 컴포넌트 간 상태 전달 복잡도 해결 및 전역 상태 아키텍처 재설계**
   - 문제: 글쓰기 핵심 화면에서 Props, Emit, EventBus, Router를 혼합 사용하면서 컴포넌트 간 데이터 흐름이 과도하게 복잡해지고 개발 속도 저하 및 유지보수 어려움 발생
   - 원인 정의: 상태 공유 구조 없이 분산된 방식으로 데이터 전달을 처리하여 컴포넌트 의존성이 증가하고 단방향 데이터 흐름이 깨지는 구조적 문제
   - 해결:
     - 전역 상태 관리 도구(Vuex) 도입으로 상태 접근 경로 일원화
     - 컴포넌트 간 직접 전달 제거 → Store 기반 참조 구조로 변경
     - 핵심 글쓰기 상태를 모듈 단위로 분리하여 책임 범위 명확화
     - 데이터 흐름을 단방향(Store → View)으로 재설계하여 예측 가능성 확보
   - 결과:
     - 컴포넌트 간 결합도 감소 및 코드 복잡도 대폭 완화
     - 글쓰기 핵심 화면 안정적으로 완성 및 개발 속도 향상
     - 상태 관리 패턴에 대한 이해 확보 및 협업 효율 개선

경력

데이터스프링코리아 | Survey & Solutions 팀, 대리 **2022.02 ~ 2025.06**

내부 견적 시스템 설계 및 개발을 통한 프로세스 간소화

- 문제: 기존 견적 작성 프로세스가 수작업으로 이루어져 비효율적이며, 작성 시간이 오래 걸리고 오류 발생 빈도가 높음.
- 해결책: Figma를 활용해 UI/UX를 설계한 후, Google Apps Script, HTML, CSS, JavaScript를 사용하여 내부 견적 시스템 instaQuote 개발.
- 효과: 견적 작성 시간이 평균 40% 단축되고 오류율도 감소하였으며 , 사용자 경험이 개선되어 내부 활용도가 증가. 자동화된 시스템 덕분에 적은 인원으로도 더 많은 견적을 처리할 수 있게 됨.

instaQuote 정보:

- 사이트 주소: https://insta-quote-six.vercel.app/
- 프로젝트 Repository: https://github.com/nsh0823/instaQuote

매크로 및 자동화 툴을 활용한 업무 최적화

- 문제: 반복적인 업무 프로세스로 인해 업무 시간이 증가하고, 지속적인 수작업으로 손목 부담이 가중됨.
- 해결책: 매크로 및 자동화 툴을 활용하여 데이터 입력 및 검증 프로세스를 구축, 반복 작업을 자동화.
- 효과: 업무 처리 속도가 2배 이상 향상되었으며, 수작업 감소로 인해 손목 부담이 크게 줄어듦.

JavaScript 자동화를 통한 부서 업무 효율 개선

- 문제: 노후된 내부 툴로 인해 반복적인 데이터 입력 및 가공 작업이 많아 업무 시간이 과도하게 소요됨.
- 해결책: JavaScript 및 Google Apps Script를 활용하여 데이터 입력 및 변환 프로세스를 자동화하는 스크립트를 개발.
- 효과: 반복 업무 소요 시간이 평균 60% 단축되었으며, 수작업으로 인한 오류 발생률이 감소하여 업무 정확도 향상.

학력

Furman University | 일본학과 2014.08 ~ 2018.05

- 사우스캐롤라이나주, 미국
- 학점: 3.89 / 4.0
- Phi Beta Kappa 회원 (미국 대학생이 받을 수 있는 최고 수준의 학문적 영예)
- Dean's List (4년 연속 우등생 명단 유지)

Seinan Gakuin University | 일본학과(교환유학) 2016.08 ~ 2017.05

- 후쿠오카, 일본

R. L. Paschal High School 2010.08 ~ 2014.05

- 텍사스, 미국

스킬

Frontend: React, TypeScript, JavaScript, Tailwind CSS, Emotion, Zustand, TanStack, HTML5/CSS3, Vite
Backend: Node.js, Express, NestJS, Socket.io
Database: MySQL, TypeORM
DevOps: AWS(EC2, RDS, IAM), NCloud, Docker, Nginx, GitHub Actions
Testing: Jest, Vitest, Storybook
Tools: n8n, Git, Figma, Jira, Postman, VS Code, Cursor

어학

영어 | 원어민 수준

일본어 | 비즈니스 회화가능 (JLPT N1)

자기소개서

자기소개

≪도전을 두려워하지 않습니다!≫

저는 어린 시절부터 가족과 동반으로 미국으로 유학을 가서, 다양한 경험과 새로운 도전을 통해 넓은 시야를 형성할 수 있었습니다. 처음 미국에 도착했을 때, 한국과 문화도 너무 다르고 모든 환경이 낯설어서 적응하기가 너무 힘들었습니다. 또 당시 저의 내성적이고 소극적인 성격 때문에 학교에서 친구를 새로 사귀는 것조차 어려워서 외롭고 우울하기도 했습니다. 하지만, 불평불만 없이 묵묵히 다양한 경험을 긍정적인 마인드로 즐기시는 부모님의 모습을 보고 조금씩 도전정신과 자신감을 갖춰 나아갈 수 있었습니다. 그 결과 저의 부정적이고 소극적인 태도는 긍정적이고 적극적인 태도로 바뀌었고, 저의 도전정신과 자신감은 힘들었던 미국 생활에 큰 버팀목이 되었습니다. 저는 학업 외에도 테니스팀, 오케스트라, 유학생회, 시각장애인들을 위해 점자성경을 만드는 봉사활동, 난민들에게 바이올린을 가르치는 봉사활동 등 다양한 활동에 적극적으로 도전을 하면서 시야를 넓힐 수 있었고, 이 활동을 통해 만난 다양한 사람들과 좋은 관계를 형성할 수 있었습니다. 지금의 저는 도전을 두려워하지 않고 오히려 도전하는 것을 즐길 수 있게 되었고, 이러한 도전과 경험 끝에 얻은 넓은 시야를 통해 어떤 상황에서도 잘 적응하고 어떤 일이든 빨리 배우고 습득할 수 있는 능력과 자신감이 생겼습니다.

성장과정

≪꾸준히 성장하는 개발자가 되고 싶습니다≫

저의 학창시절을 돌이켜 보면 제가 가장 좋아하고 잘했던 과목은 수학이었습니다. 대학교 때는 수학 관련 수업도 많이 듣고 수학을 전공하려고 했었던 것 만큼 수학 관련 과목들을 좋아했습니다. 그리고 대학교 마지막 학기에 선택과목으로 컴퓨터 프로그래밍 수업을 들었는데, 처음부터 이 수업을 안 들은 것이 후회될 정도로 프로그래밍이 재미있었고 저의 성향과 매우 잘 맞았습니다. 이 수업에서 저는 랩실에 혼자 남아 밤을 세워가면서 프로젝트를 한 적이 있고, 교수님께서 이러한 저의 노력을 인정해 주셔서 좋은 학점을 받을 수 있었습니다. 또 이 수업을 통해서 개발자가 되고 싶은 꿈이 생겼습니다. 저는 이 꿈을 이루기 위해 스스로 무료 온라인 강의를 듣고 간단한 프로젝트를 구현해보면서 다양한 프로그래밍 언어를 공부하려고 노력했고, 또 엔코아 플레이데이터와 네이버 부스트캠프 교육과정을 통해 열심히 공부해서 단기간에 코딩 실력이 많이 향상 되었습니다. 제가 수학과 프로그래밍을 재미있어하고 좋아하는 이유는 문제를 분석하고 해결하는 과정에서 마지막에 해결했을 때 느끼는 희열과 성취감 때문입니다. 저는 많은 사람들의 문제를 해결하고 그들이 조금 더 편하게 살아갈 수 있도록 도와주는 개발자로 성장하고 싶습니다. 이러한 목표와 열정을 가지고 즐기면서 일을 한다면 오랫동안 지치지 않고 꾸준히 성장할 수 있을 것이라고 자신합니다.

성격의 장단점

≪끝까지 포기하지 않는 근성≫

저는 무엇이든 한번 시작하면 끝을 보는 성격을 소유하고 있습니다. 이번 부스트캠프 과정을 수행하면서, 비전공자로서 시작했기 때문에 기본적인 컴퓨터 공학 지식이 많이 부족했고, 특히 챌린지 기간에는 매일 주어지는 미션을 수행하는 과정이 쉽지 않았습니다. 학습 내용을 따라가는 것조차 벅차게 느껴질 때가 많았고, 여러 번 포기하고 싶은 마음이 들 정도로 어려움을 겪었습니다. 하지만 저는 이러한 상황에서도 포기하지 않고 문제를 끝까지 해결하려고 노력했습니다. 이해가 되지 않는 개념은 반복해서 찾아보고, 다양한 자료를 참고하며 스스로 납득할 때까지 학습했습니다. 또한 매주 주기적으로 회고를 하면서 제가 부족했던 점과 개선할 점을 정리하고, 이를 다음 학습에 반영하면서 조금씩 보완해 나갔습니다. 과제를 수행할 때에는 시간에 구애받지 않고 몰입하여, 거의 매일 새벽 4~5시까지 학습을 이어가며 끝까지 해결하려고 노력했습니다. 또한 이러한 어려움 속에서도 개발에 대한 재미를 발견하게 되었고, 이는 지속적으로 학습을 이어갈 수 있는 큰 원동력이 되었습니다. 그 결과, 처음에는 막막하게 느껴졌던 문제들도 점차 스스로 해결할 수 있게 되었고, 학습 내용을 이해하는 깊이 또한 점점 깊어졌습니다. 이러한 경험을 통해 저는 어려운 상황에서도 쉽게 포기하지 않고 끝까지 해결하려는 태도를 갖추게 되었으며, 끈기와 꾸준함이 결국 성장을 만든다는 것을 몸소 느낄 수 있었습니다. 저의 단점은 내성적인 점입니다. 처음에는 낯설어 하는 경향 때문에 주변 사람들과 어울리기가 어려울 수 있지만, 시간이 지나고 가까워지면 돈독한 인간관계를 맺는 편입니다. 부스트캠프 과정에서도 협업과 피드백을 통해 소통의 중요성을 느꼈고, 이를 개선하기 위해 동료들과 적극적으로 의견을 나누려고 노력하고 했습니다. 앞으로도 이러한 점을 보완하여 더 원활한 협업을 만들어 나가겠습니다.

어려움을 극복한 사례

≪AI 기반 대량 문제 생성 자동화를 통한 병목 해결≫

부스트캠프 최종 그룹 프로젝트로, CS 학습을 게임처럼 즐길 수 있는 게이미피케이션 플랫폼 Funda를 개발했습니다. 저는 대량 문제 생성을 자동화하고 출력 포맷의 안정성을 확보하는 데이터 파이프라인 구축을 맡았습니다. 문제 데이터는 서비스의 핵심 자산이었고, 콘텐츠 제작 속도는 전체 일정과 직결되는 병목 구간이었습니다. Funda는 Field → Unit → Step → Quiz 구조로 설계되어 다양한 분야의 고품질 문제를 대량으로 확보해야 했습니다. 초기에는 AI에게 문제를 10개씩 요청해 수동 검수 후 저장했지만, 이를 6,000개 규모로 확장하면 60시간 이상이 소요될 것으로 예상되었습니다. 또한 반복 작업에 따른 비효율과 휴먼 에러 위험도 컸습니다. 이에 n8n 기반 자동화 파이프라인을 설계했습니다. Google Sheets의 커리큘럼을 읽어 Field·Unit 단위로 Loop를 구성하고, 생성과 파싱 단계를 분리했습니다. Structured Output Parser로 JSON Schema 검증을 적용했지만, AI가 간헐적으로 포맷을 깨뜨려 malformed function call 오류가 발생했고, 한 번의 오류로 전체 루프가 중단되는 치명적인 문제가 있었습니다. 이를 해결하기 위해 메인-서브 워크플로우 계층 구조로 재설계했습니다. 5개 Unit마다 중간 저장 분기를 두어 일부 오류가 발생해도 생성 데이터가 보존되도록 했고, 실패 구간만 재실행 가능하도록 구성했습니다. 또한 프롬프트를 더 구체화하고 Temperature를 낮춰 출력 일관성을 높였으며, Delay 노드로 RPM 초과도 방지했습니다. 그 결과 750문제를 약 30분 만에 생성할 수 있었고, 6,000문제도 약 4시간 내 제작 가능한 구조로 개선했습니다. 반복 작업을 제거해 팀 생산성을 크게 향상시켰으며, 오류에도 전체 프로세스가 중단되지 않는 안정적인 파이프라인을 구축했습니다. 향후에는 생성된 문제를 즉시 DB에 적재하는 단계까지 파이프라인을 확장해, 파일 기반 중간 저장 과정을 제거하고 데이터 흐름을 더욱 일관되게 만들고 싶습니다. 또한 escape 이중 처리로 인해 발생할 수 있는 데이터 왜곡 문제를 구조적으로 개선해, 생성부터 저장까지의 정합성과 안정성을 더욱 강화하고 싶습니다.