

황현규

이메일: joseph95501@gmail.com

휴대폰: 010-9049-9550

포트폴리오: <https://01joseph-hwang10.github.io/>

Github: <https://github.com/01Joseph-Hwang10/>

주니어 개발자 황현규입니다. 창업과 스타트업에 관심이 많습니다.

창업 아이템을 내 손으로 직접 구현하기 위해 개발을 시작했으며, 나의 아이디어를 직접 구현하여 세상에 가치를 제공할 때 느끼는 보람이 매우 커, 학부 신입생 때부터 교내외의 스타트업, 다수의 해커톤 및 창업 활동에 참여하여 팀의 핵심 인원으로서 활약하였습니다.

"최소 자원으로 최대 효용"이라는 린 스타트업의 철학에 깊은 공감을 하며, 업무를 진행할 때도 어떻게 하면 최소한의 자원으로 최대의 효과를 이끌어낼 수 있을지 끊임없이 고민합니다. 또한 항상 "왜?"라는 질문을 던지며, 무엇을 하는지에 대한 뚜렷한 목적을 가지고 업무를 진행합니다.



Start-ups

스타트업과 창업에 관심이 많습니다. 심층 인터뷰 등의 시장조사를 통해 잠재적 Pain Point를 찾고 린 스타트업 방법론에 기반해 MVP를 개발해나가는 과정을 즐깁니다.



Lean Programming

최소의 자원으로 최대의 효과를 추구합니다. 기능을 구현할 때는 그 목적을 항상 검토하고, 구현에 있어 불필요한 프로세스가 있는지를 유심히 살펴봅니다.



Code Aesthetics

가독성이 높은 코드, 유지보수가 용이한 코드를 지향합니다. OOP를 비롯한 다양한 디자인 패턴을 적재적소에 활용해 처음 보는 코드라도 금방 이해하고 기여할 수 있게끔 노력합니다.

학력 사항

울산과학기술원 (2021.03 - 휴학 중)

- 컴퓨터공학과 학사과정 (주전공)
- 경영과학부 학사과정 (부전공)

경력 사항

트립빌더 (@tripbuilder)

- 2021.07 - 2021.12
- 개발팀 / Front-end Developer
- AI 기반 여행 계획 추천 모바일 앱 TripBuilder 개발 참여

랩이즈 (@labis)

- 2021.12 - 2022.11
- 개발팀 / Development Lead & Co-Founder

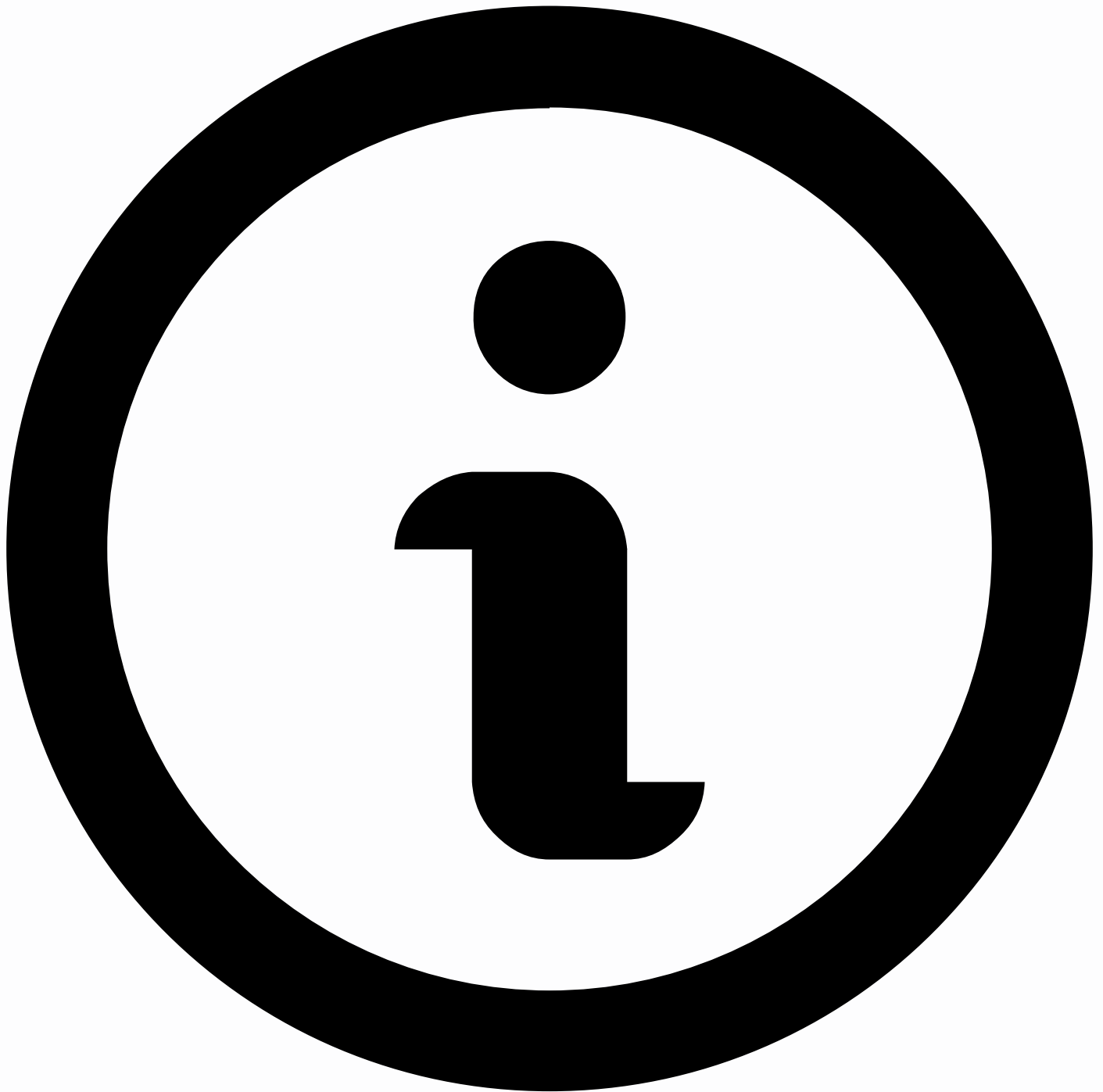
- 하드웨어 기반의 블록 코딩 교육 솔루션 [LearnQue](#) 기획 및 개발

셰퍼드23 (@shepherd23)

- 2022.07 - 현재 재직중
- 개발팀 / Product Manager & Software Engineer
- E-Commerce 쇼핑몰을 위한 개인화 상품 추천 플러그인 [PickHound](#) 기획 및 개발

업무 경험

PickHound [↗](#)



노트

PickHound는 쉐퍼드23에서 개발한 Cafe24 온라인 쇼핑몰을 대상으로 하는 **상품 추천 플러그인**입니다. **Contextual Bandit**에 기반해 팝업 배너를 통해 각 쇼핑몰의 상품을 사용자 개인의 선호, 취향 등을 고려한 개인화 상품 추천을 제공합니다. 베타 테스트 단계를 거치고 있으며, 현재 **60곳 이상의 실사용 고객사의 25k 가량의 쇼핑몰 방문객에게 상품 추천 서비스를 제공하고** 있습니다. **서비스의 기획, 개발, 운영을 총체적으로 담당**하면서, 심층 인터뷰 및 고객 데이터 수집을 통한 피드백 수집, 플러그인 및 추천 시스템 개발 및 유지보수, 서비스 운영을 진행하고 있습니다. 이러한 과업을 진행하며 현재도 수많은 기술적, 사업적 문제를 맞닥뜨리고, 이를 해결해나가는 과정에서 많은 배움과 성장을 얻고 있습니다.

=> 심층 인터뷰를 통한 수요 파악 및 제품 방향성 수립

Project Management Communication @shepherd23

- 약 1달에 걸쳐 다양한 분야에 종사하는 12명의 쇼핑몰 관리자를 대상으로 인터뷰 진행함.

- 인터뷰를 통한 피드백 수집으로 쇼핑몰의 규모에 따른 서비스의 수요, 운영 방식의 차이 등을 파악할 수 있었음.
- => 정리한 피드백에 기반한 **불필요한 개발 단계 제거로 사내 개발 자원 낭비 방지.**



팁

아래 링크에서 위와 관련한 더 자세한 내용을 확인할 수 있습니다.

<https://01joseph-hwang10.github.io/posts/deep-interview>

=> Lerna에 기반한 프로젝트 관리 및 공유 코드 라이브러리화

Automation

Improvement

@shepherd23

- Lerna를 활용한 모노리포 구성, 패키지 버저닝 및 `commitlint`를 활용한 커밋 메시지 규칙 적용.
 - Github Actions, Bash Script, Husky를 활용해 버저닝 자동화 및 커밋 메시지 규칙 강제.
- 공유가 필요한 코드는 따로 패키지로 분리해 및 Google Artifact Registry를 활용해 private `npm` 패키지 배포.
- => **프로젝트 관리 체계화 및 자동화**
- => **비효율적인 복사 붙여넣기 방식의 코드 공유를 최소화.**

=> Jest, Pytest, Cypress를 활용한 TDD로 개발 흐름 효율화

Frontend

Backend

Improvement

@shepherd23

- Jest, Pytest를 활용해 주요 엔드포인트 및 서비스 로직에 대한 Unit/E2E 테스트 작성.
- Insomnia 등의 툴을 이용한 수동 테스트 시보다 훨씬 효율적인 개발 흐름을 구성할 수 있었음
 - => **기존 수동으로 테스트를 진행했을 때와 비교해 작업 시간 약 90% 감소. (기존 약 20분 → TDD 도입 후 최대 2분)**
 - => **다른 팀과 버그 수정 관련한 불필요한 대화가 오고 가는 빈도 약 80% 감소. (기존 평균 4 ~ 5회 → TDD 도입 후 0 ~ 1회)**
- Cypress를 활용해 복잡한 인증/인가 로직을 포함하는 컴포넌트에 대한 Unit 테스트 작성.
 - => 테스트를 통한 Iteration 속도 향상으로 빠른 기능 구현 및 버그 수정.

=> Google Sheets를 활용한 간이 어드민 대시보드 구축

Feature

@shepherd23

- NestJS는 Django처럼 Admin Dashboard를 제공하지 않고, Thrid Party 라이브러리 (e.g. AdminJS) 도 Firestore 를 DB로 사용하는 경우를 고려한 라이브러리는 없었음.
- 이에 Google Sheets, Sheets API, Cloud Functions 를 활용해 사용자 데이터 시각화 및 쇼핑물 관리 기능을 포함하는 간이 대시보드 구축함.
- => UI 구현 필요 없이 간단한 로직 구현만 필요했기에 빠른 대시보드 구현이 가능했음.
- => 엑셀에 기반한 대시보드 구현으로 비개발 인원도 쉽게 지표를 확인하고 쇼핑몰을 관리할 수 있었음.

=> 코드주석과 Notion을 연계한 사내 문서화 작성 효율화 및 생성 자동화

Devops

Improvement

@shepherd23

- 기존에는 비개발 인원이 개발 상황을 파악하기 위해 개발자에게 직접 문의하거나, 개발자가 이미 코드 주석으로 작성된 내용을 Notion에 다시 작성하는 등의 비효율이 존재했음.
- 이에 Swagger, Typedoc, Sphinx 등의 툴을 활용해 문서화를 생성하고, 노션 페이지와 문서화 페이지의 웹 링크를 연계하는 문서화 작성 컨벤션 수립.
 - Github Actions, Cloud Build, Cloud Run을 활용해 문서화 페이지 배포 자동화.
- => **중복되는 주석 작성을 최소화하고, 개발 인원의 개입 없이도 비개발 인원이 개발 상황을 파악할 수 있도록 함.**

See Also

- 정적 웹페이지 보호를 위한 Express 애플리케이션 개발

=> Redux를 이용한 FLUX/MVC 기반의 React & StencilJS 상태 연동

Frontend

Improvement

@shepherd23

- React와 StencilJS로 빌드된 Web Component의 상태 관리 방식이 다르기 때문에 각각의 경우에 맞는 상태 관리를 구현해야 했음.

- React, `@reduxjs/toolkit`의 `createSlice()` API를 활용해 `FLUX` 패턴을 기반으로 디자인 수정 페이지 구현
- MVC 패턴에 기반해 StencilJS Web Component의 style을 관리하는 `Banner Manager` 객체 작성
 - 스타일 객체를 `M`, `Redux Slice`를 `V`에 대응하게끔 구현하고 `C`에 해당하는 `Banner Manager` 객체를 작성.
 - `structuredClone`을 활용한 새로운 `Object Reference` 생성으로 상태 업데이트
- `FLUX` 중 `Store`에 해당하는 `Redux Slice`를 `MVC` 중 `V`에 대응하여 두 컴포넌트의 상태를 연동.
- => 적절한 코드 패턴의 적용으로 간결하고 직관적인, 유지보수에 용이한 상태 관리 코드 작성.
 - => 확장성 있는 코드 작성으로 *유저가 피드백한 배너의 위치 변경 기능을 하루 만에 구현, 테스트 및 배포할 수 있었음.*
- => MVC, FLUX의 각 요소와 요소 간의 상호작용에 관한 구체적인 이해 함양.



팁

아래 링크에서 위와 관련한 더 자세한 내용을 확인할 수 있습니다.

<https://01joseph-hwang10.github.io/posts/react-stenciljs-integration>

=> 상품 정보에 대한 Caching 구현

Backend **Optimization** @shepherd23

- 추천에 필요한 상품 정보를 Firestore에서 가져오는 과정에서 발생하는 불필요한 읽기 과정을 줄이기 위해 인메모리 캐시 도입.
- 상품 정보에 대한 캐싱 및 캐시 무효화 기능을 제공하는 NestJS 모듈 `ActionProviderModule` 구현.
- => **도큐먼트 읽기 횟수** `30,000,000/일` -> `1,500,000/일`로 감소 -> **매우 큰 비용 절감 효과**

See Also

- PickHound: <https://01joseph-hwang10.github.io/docs/career-description/pickhound>
- `ActionProviderService`: <https://01joseph-hwang10.github.io/docs/career-description/pickhound#상품-정보에-대한-caching-구현>

=> Leaky Bucket이 적용된 엔드포인트를 위한 HTTP Client 구현

Feature @shepherd23

- 데이터 요청이 필요한 API의 Leaky Bucket Policy로 인해 1초에 2회 수준으로 API 호출이 제한됨.
- 각 상품의 상세 정보를 하나씩 전부 요청해야 하는 상황에서 단순 API 호출로는 HTTP 429 에러 발생.
- `LeakyBucketClient` 클래스를 구현해 호출의 빈도를 제한하고, 429 에러 발생 시 리퀘스트 재시도.
 - HTTP 에러 코드별로 `Exception` 클래스를 만들고 `instanceof` 연산자를 활용해 에러를 구분하고 대기 시간을 조절.
- => 상품 추천에 필요한 데이터인 상품의 상세 정보를 전부 요청하는 작업을 가능케 함.

See Also

- `LeakyBucketClient`: <https://01joseph-hwang10.github.io/docs/career-description/pickhound#leaky-bucket이-적용된-엔드포인트를-위한-http-client-구현>
- 긴 프로세스를 담당하는 Worker 인프라 구축 및 배포 자동화

=> 긴 프로세스를 담당하는 Worker 인프라 구축 및 배포 자동화

Devops **Feature** **Automation** @shepherd23

- 기존에는 모든 서비스를 Google Cloud Run을 이용해서 배포했음.
 - 데이터 요청이 필요한 API의 Leaky Bucket Policy로 인해, 상품 정보를 자사 Firestore DB로 가져오는 프로세스가 길어지면서 Cloud Run의 프로세스 제한 시간인 1시간을 초과하는 문제 발생.
- 프로세스 제한 시간이 없는 GCE 인스턴스 그룹과 Cloud Load Balancer를 활용한 Worker 인프라를 구축하여 이슈 해결.
 - Cloud Build Substitution Variable 과 Docker Build Args를 활용해, Cloud Run과 Worker 인프라 배포 코드 일반화.
- => Cloud Build와 Docker를 이용한 **CI/CD 자동화로 인력 소비 최소화** 및 GCE Instance Group과 GCLB를 활용한 **시스템의 유연성 확보**.
- => 인프라 구축 과정에서 외부 API의 제약사항을 고려한 Worker 인프라 구축과정에서 실제 프로덕션 환경에서의 문제 해결 능력 향상

See Also

- `cloudbuild.yaml`, `deploy.sh`, `Dockerfile`: <https://01joseph-hwang10.github.io/docs/career-description/pickhound#긴-프로세스를-담당하는-worker-인프라-구축>

=> GCE MIG Autoscaler 의 인스턴스 종료로 인한 프로세스 중단 문제 해결

Backend Devops Troubleshooting @shepherd23

- GCE 관리형 인스턴스 그룹 (MIG) Autoscaler가 수평 축소를 하면서 인스턴스를 종료하는 과정에서, 종료되는 instance에 할당된 프로세스를 중단하는 문제가 발생함.
 - 수평 축소 제어를 사용하지 않는 Autoscaler의 경우 최근에 관찰된 부하를 처리하는 데 필요한 인스턴스만 유지. (Reference)
 - 긴 시간이 소요되는 프로세스의 인스턴스가 종료의 대상이 되어 프로세스가 중단되는 문제 발생.
- NestJS의 `onApplicationShutdown` 인터페이스를 구현하여 프로세스 종료 시점에 진행중인 작업을 재시작하는 로직을 구현해 이슈 해결.
 - Fire and forget 방식으로 request를 보내 종료 프로세스에 영향을 주지 않도록 함.
- => GCE MIG와 Autoscaler에 관한 깊은 이해를 얻음. 각 Cloud Service의 특성을 고려한 시스템 구현.

See Also

- `MailSnapshotService.onApplicationShutdown`: <https://01joseph-hwang10.github.io/docs/resume/backend-agnostic#gce-instance-group-autoscaler-의-instance-termination으로-인한-프로세스-중단-문제-해결>

=> 데코레이터 패턴을 활용한 AI 팀과의 협업 효율화

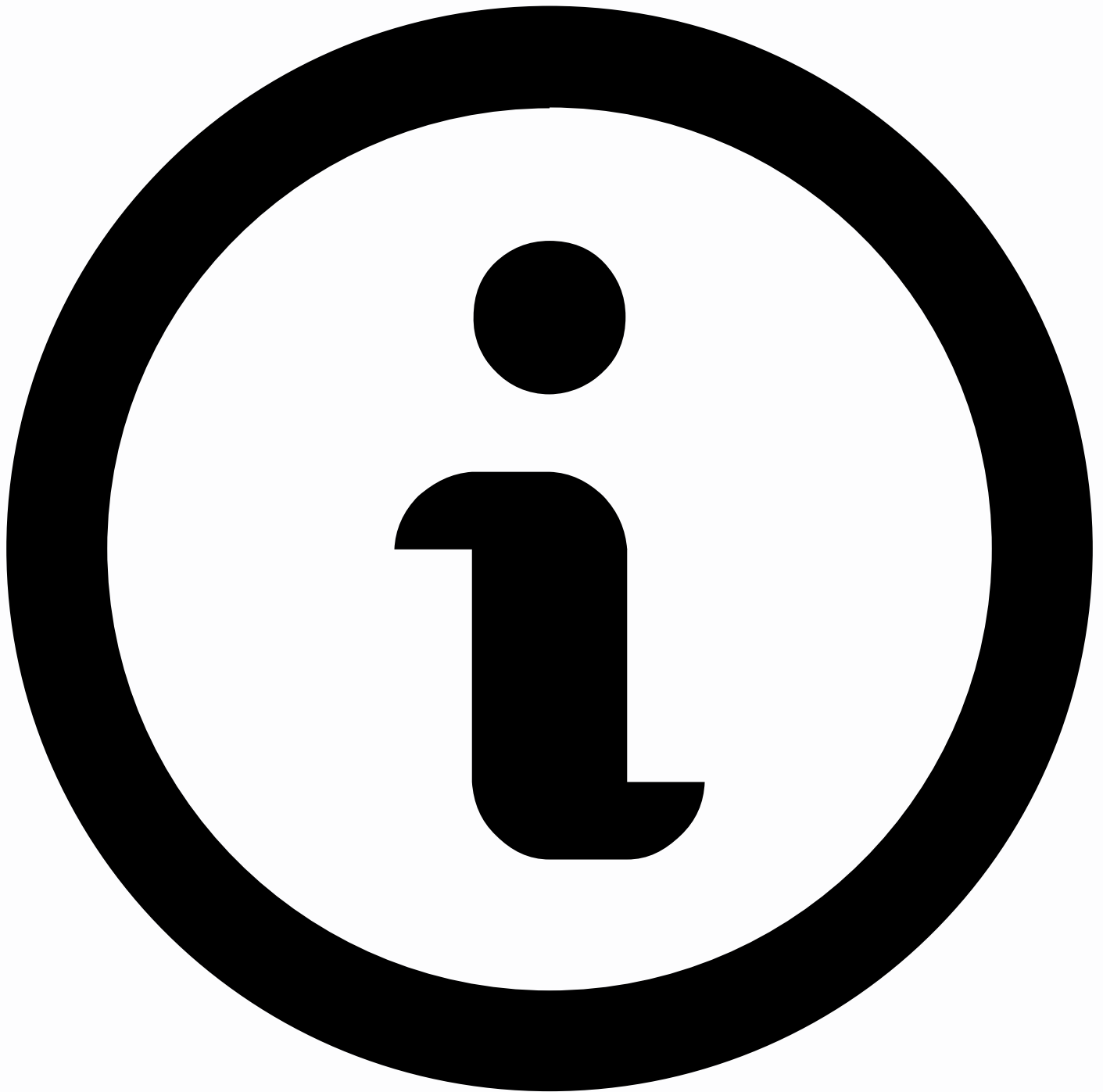
Backend Improvement @shepherd23

- 데코레이터 패턴을 활용해 각 엔드포인트의 핸들러 함수를 데코레이터로 감싸 에러 핸들링, 인증 등의 공통된 로직을 분리함.
- `TypedDict`를 활용한 DTO 작성 및 Python Type Hint 기능을 활용해 각 핸들러 함수의 인자와 반환값을 명시함.
- => 웹과 관련한 요청/응답 로직을 데코레이터를 통해 추상화하여 단순한 IPO (Input-Process-Output) 프로그램으로 문제를 단순화.
- => **신입 AI 팀원도 별도의 웹 관련 교육 없이 바로 업무에 참여 가능한 수준으로 웹에 익숙하지 않은 AI 팀의 업무 효율화.**

See Also

- Before & After: <https://01joseph-hwang10.github.io/docs/resume/backend-agnostic#데코레이터-패턴을-활용한-ai-팀의-업무-효율화>

LearnQue 



노트

LearnQue는 랩이즈에서 개발한 완전 모듈형 드론 기반의 체감형 블록코딩 교육 솔루션으로, 단순히 드론 조종, 기본 프로그래밍에서 끝나는 것이 아니라 택배모듈, 소방모듈 등 실제 현장에서 만나볼 수 있는 교육을 지향하였습니다. 공동창업자로서 심층 인터뷰와 정책 동향에 기반한 설득력있는 투자 유치 활동으로 울산청년창업사관학교, 창업 300 밸류-UP 등 국가/민자 엑셀러레이팅 프로그램을 유치, 수료하였으며, 이 과정에서 창업과 벤처 업계에 관한 심도 있는 이해를 가질 수 있었습니다. 또한 하이브리드 애플리케이션을 개발하며, 블록코딩 UI, 블록코드 컴파일러, 시리얼 통신 등 처음 접하는 문제를 해결해 나가는 과정에서 기술적으로 많은 것을 배울 수 있었습니다.

=> 서비스 기획을 위한 심층 인터뷰 진행

Service Planning

Communication

@labis

- 울산광역시 중구/북구/문수/성남/동구 청소년문화의집 등 울산 내의 많은 방과후 교육 기관을 방문해 데이터 수집함.

- 시장의 수요 파악을 위해 타겟인 프리랜서 강사 및 청소년문화센터 담당자와의 심층 인터뷰 진행함.
- => 심층인터뷰를 통한 데이터 수집 과정을 통해 인터넷 서칭으로는 찾을 수 없는 실질적인 현장의 Pain Point를 파악할 수 있었음.
- => 이후 엑셀러레이팅 프로그램 유치에 큰 영향

=> 엑셀러레이팅 프로그램 유치를 위한 사업계획서 및 피치덱 작성

Project Management @labis

- 심층 인터뷰를 통해 수집한 데이터를 바탕으로 설득력 있는 사업계획서 및 피치덱을 작성함
- => 2022 울산청년창업사관학교 엑셀러레이팅 프로그램 유치 성공
- => 창업 300 밸류-UP 엑셀러레이팅 프로그램 유치 성공

아래 링크에서 위의 피치덱을 확인할 수 있습니다.

<https://drive.google.com/file/d/1HczhzUcrE3G31cn9tiu6fXh3vUofvldq/view?usp=sharing>

=> ESP32와 react-native-ble-plx를 활용한 드론 블루투스 통신부 개발

Frontend Feature @labis

- react-native-ble-plx의 제공 API가 Bluetooth 연결 가능한 디바이스를 활용하는데 있어 비교적 저수준의 API만 제공해 코드가 복잡해지는 문제가 있었음.
- Prototype extension을 통해 react-native-ble-plx의 API를 확장하고, 이를 활용해 ESP32와의 블루투스 통신부를 깔끔하게 구현함.
- => Javascript가 prototype based language라는 특성을 활용해 라이브러리의 API를 확장함으로써 코드의 복잡성을 줄임.

스킬셋

Typescript HTML5 CSS3 ES6 CommonJS Webpack Babel Rollup Vite Lerna
 Python Numpy Pandas Scikit Learn Django Flask PyTest Selenium Sphinx React
 NextJS Redux RTK Query TailwindCSS MUI Emotion Cypress Storybook StencilJS
 D3 Figma NestJS Express Jest Swagger TypeORM Typedoc SQL Firestore
 Cloud Computing Firebase Cloud Run Cloud Build Cloud Storage Cloud Logging
 Cloud Compute Engine Docker Linux Git Github Slack Notion Korean (Native)
 English (TOEFL iBT 98 @ 2023.02)

수상 및 기타

- 2022 울산청년창업사관학교 수료
- 2022 창업유망 300 밸류-Up 프로그램 수료
- 동남권실험실창업혁신단 주관 UNIST-Pre 아이코어 프로그램 장려상
- 2022 울산 스마트 해상물류 창업 오디션 예선 프로그램 수료
- 2022 울산 스마트 해상물류 창업 오디션 본선 프로그램 수료
- 경기콘텐츠진흥원 주관 2021 애자일 해커톤 우수상
- 한국관광공사 & 카카오 주관 2021 관광데이터 활용 공모전 장려상
- 2021 울산지역대학 YOUTH 창업우수 아이디어 경진대회 대상
- 3rd PATHHACK : 쉬운 해커톤 스폰서상
- JUNCTION ASIA 2022 참가자

