

[웹, 블록체인] 이더리움 Web3 스마트 컨트랙트 주니어 엔지니어



성종현 1995년 (28세/만 27세) | 남

✉ nellow1102@gmail.com

☎ 010-2985-2984

☎ 010-2985-2984

🏠 (06097) 서울 강남구 봉은사로

학력사항	경력사항	희망연봉	희망근무지/근무형태	포트폴리오
대학교(4년) 졸업	신입	면접 후 결정	서울전체 정규직	-

학력 최종학력 | 대학교 **4년** 졸업

재학기간	구분	학교명(소재지)	전공	학점
2014.03 ~ 2020.08	졸업	경북대학교 (대구)	경영학과	4.14 / 4.5

경력 신입

근무기간	회사명	부서/직급/직책	지역	연봉
2022.06 ~ 2022.08 (3개월)	아이언플래그	Blead studio 개발 / 사원 (팀원) / 1년차	서울	4,500만원
<div>담당업무</div> 블록체인 스마트 컨트랙트 개발 <div>퇴사사유</div> 계약만료 <div>프로젝트</div> <ul style="list-style-type: none">● IF-NET 2022.06~2022.07(2개월) 사내에서 진행한 블록체인 NFT 테스트 프로젝트입니다. 당시 블록체인 기업임에도 불구하고 전체 개발팀(프론트 2, 백엔드 2, 리드 1, 블록체인 1) 중에서 저를 제외한 5명의 팀원들이 블록체인 지식이 전무하고, 입사 이후 처음 공부를 진행하는 상황이었습니다. 이에 대응하기 위해 적극적으로 제가 알고 있는 지식을 공유하는 동시에 진행된 테스트 프로젝트(리드 1, 블록체인 1)입니다. 프로젝트에서 맡은 역할은 아래와 같습니다. - 리드 개발자와 기획 회의 및 기능 정의 - 스마트 컨트랙트 테스트 주도 개발 도입 - 배포 이후 수정 가능한 스마트 컨트랙트 작성 및 테스트 케이스 작성: ERC20, ERC721, 테스트 케이스 20개 - 블록체인 지식 공유: 아키텍처, 컨트랙트 상호 작용, off-chain NFT 미디어 저장 등 해당 프로젝트는 리드급 개발자가 인프라 설계를 위해 테스트 프로젝트를 중단하는 것으로 결정을 내린 후 마무리 되었습니다. ● Sharking-swap 2022.07~2022.08(2개월) 사내에서 진행한 블록체인 DeFi 테스트 프로젝트입니다. 위 프로젝트에서 설명한 상황에 대응하기 위한 개발 팀 내 자체 프로젝트입니다. 프로젝트에서 맡은 역할은 아				

	<p>래와 같습니다.</p> <ul style="list-style-type: none"> - 전체 개발 팀과 기획 회의 - Uniswap Ver2 프로토콜 및 컨트랙트 분석 - 스마트 컨트랙트 구조 설계 - 토큰 스왑 기능 구현 및 테스트 케이스 작성: ERC20(USDC, DAI) Ether, 테스트 케이스 20개 - Chainlink Orade(Data Feeds) 도입 및 테스트 케이스 작성 - 블록체인 지식 공유: DeFi pool, yield farming, service example <p>프로젝트 기획/진행 중에서도 팀원들의 블록체인 이해도 부족으로 인한 프론트엔드 블록체인, 백엔드 블록체인 간의 협업이 불가능한 상황이 계속해서 발생했습니다.</p> <p>최종적으로 해당 프로젝트는 중단되었고, 저를 제외한 나머지 팀원들은 문서를 통한 블록체인 학습을 한 달 동안 진행하게 되었습니다.</p> <p>두 개의 중단된 프로젝트 및 소통 불가는 자연스럽게 이직 및 퇴사 결정으로 이어지게 되었습니다.</p>
--	--

자격증/어학/수상내역

취득일/수상일	구분	자격/어학/수상명	발행처/기관/언어	합격/점수
2021.08	수상내역/공모전	블록체인 해커톤 피칭전 최우수상	과학기술정보통신부, 한국교육원	-

보유기술

보유기술명/수준/상세내용
Github Actions
Agile
TDD
JIRA
Github
Git
Ethereum
Web3.js
Ethers.js
Upgradeables
Chai
Mocha
Hardhat

Openzeppelin
Typescript
Javascript
Solidity

포트폴리오/기타문서

파일 구분	파일명
기타	https://github.com/developerasun

자기소개서

[Github 1년 커밋 10,000+, 폭풍 성장형 주니어]

* 연락처: nellow1102@gmail.com, 010.298오.298사

===== 이런 일을 하고 있어요 =====

1. 깃허브

- 프로필 : <https://github.com/developerasun>
- github organization(personal): <https://github.com/asunlabs>

2. 개인 프로젝트 : PawCon (파우콘), 웹 + 블록체인

- 기간 : 2021.09 ~ 총 인원 1명 (진행 중)
- 역할 : 기여도 100%
- 링크
- * Migration: <https://github.com/asunlabs/pawcon-monorepo/tree/develop>
- * Docs: <https://docs-pawcon.netlify.app/>
- * Legacy: <https://github.com/developerasun/pawcon>

3. 팀 프로젝트 : MeowRun (묘오런), 웹 + 유니티 3D

- 기간 : 2022.02.15 ~ 2022.05.31 총 인원 3명 (현재 중단)
- 역할 : 유니티 메인 + 웹 클라이언트 메인
- 링크
- * Unity map design: <https://github.com/DaeguDev/MeowRun/discussions/5>
- * Unity enemy coding: <https://github.com/DaeguDev/MeowRun/discussions/4>
- * Web landing page: <https://meowrun.netlify.app/>

===== 개인 프로젝트 섹션별 설명 =====

1. 깃허브

- Agile 적용: Github project Team backlog template
- Git hook 적용: branch naming enforcing
- Pandas git commit template 적용
- Git flow 적용: main, develop, feature, bugfix, hotfix
- Issue template 적용: output log 기반 트러블 슈팅
- Pull request template 적용: 반복되는 체크리스트/텍스트 자동화
- Github Actions: 라벨링 자동화, 브랜치 별 CI 유닛 테스트 자동화

1. 문서화

- HUGO: 기존 파편화 된 프로젝트 문서들 => 정적 웹사이트 업로드 및 통합 처리 중. 생산성을 위해 직접 코딩하지 않고 template 적용함.
- Netlify: HUGO 호스팅, Build hook API key 발급 => Github Actions CD workflows 연계 및 자동화

1. 블록체인 :

Latest ver 0.3.0

- * Token economy 설계: ERC777 토큰 => 거버넌스 토큰, LP/스테이킹 리워드 토큰으로 활용. ERC721 => NFT 상품 및 스테이킹으로 활용
 - * Upgradeable: UUPS 프록시 패턴 적용, 배포 이후 수정 가능한 스마트 컨트랙트 작성. ERC777/721 prototype 완료
 - * Test: ethers.js / smock / hardhat-network-helpers 활용 테스트 주도 개발 적용, 빠른 feature 개발 및 기능 안정성 확보(ERC777: test coverage 95%).
 - * Lint: hardhat-solhint, slither, hardhat-contract-sizer, hardhat-remove-logs 플러그인 조합, 이상 발생 시 CI 상에서 자동 오류 throw.
 - * CI: Turborepo + Github Actions 조합, feature 브랜치 자동 테스트 실행
 - * NFT media management: on-chain Base64 토큰 URI 저장(완료), off-chain IPFS/Pinata 토큰 URL API 서빙(예정)
 - * Oracle: Chainlink Data Feeds 솔루션 도입 및 NFT 동적 가격 설정. 스마트 컨트랙트 <=> 외부 네트워크 연결된 하이브리드형 컨트랙트 작성.
 - * Configuration: 커스텀 hardhat 태스크 작성, 컴파일러 최적화 및 솔리디티 버전 맵핑 적용, Hardhat forking 적용(블록체인 네트워크 state 복사, DeFi 테스트 환경 구성에 유리), Ethereum merge 이후 deprecated 된 네트워크 분류 주석 처리
- <https://github.com/asunlabs/pawcon-monorepo/blob/develop/apps/blockchain/hardhat.config.ts>

Legacy ver 0.2.0

! ERC20/721 컨트랙트 작성

! whitelist 명단 giveaway 수량 설정

! 민팅 희귀도에 따른 토큰 삭제 제한

! 민팅 횟수에 따른 동적 토큰 가격 설정

! Role-based 토큰 접근 제한 설정 : admin, minter, burner, destructor

! Ropsten testnet 배포

! Churusol : <https://ropsten.etherscan.io/address/0x0f4d7069B9a58699D7c369F9ac97777f8De4e8e4#code>

! Curious Pawoneer : <https://ropsten.etherscan.io/address/0x0D684a95D43169Dca01eBf6d78ac31079e6f6a31>

! NFT media management: IPFS/Pinata => CID 및 gateway 설정. 컨트랙트 내 메타데이터 / 이미지 파일 확장자 설정

1. 웹

Latest ver 0.3.0

* <https://docs-pawcon.netlify.app/web/>

Legacy ver 0.2.0

클라이언트

! HTML, CSS, JS => 생산성 증가 및 디버깅 효율을 위해 React/TS 및 Sass 도입

! React public 디렉토리 Index.html => SEO 메타 태그 작성

! React gallery.tsx => 서버로부터 데이터 API(artwork1/2/3.json) fetching 후 서버 컴포넌트에 맞게 렌더링. 서버 API 직접 작성함.

! React login.tsx => 서버 jsonwebtoken & 클라이언트 리덕스를 통한 로그인 화면 conditional rendering.

! Redux login reducer => 서버로 로그인 HTTP POST 요청 시 유저 이메일 정보 dispatch. 4)의 conditional rendering에서 사용됨.

! Redux cart reducer => 쇼핑 카트 아이템 추가/삭제, 수량 업데이트

! React router => 로그인/회원가입 요청을 제외한 모든 유저 요청을 클라이언트에서 관리. gallery.tsx에서 version 6 nesting routing.

! React <=> client side socket io 연결 후 리얼 타임 채팅 렌더링

! Docker : 클라이언트 섹션 Dockerfile 작성 및 Docker hub 이미지 배포함, Docker-compose 연계.

서버

! Node.js/Express, MongoDB => 로그인, 유저 정보 저장, 채팅 등을 위해 도입

! Express => jsonwebtoken 로그인 구현, '/login', '/logout', '/signup' 라우트에서 토큰 생성 후 쿠키 세팅. httpOnly : true 옵션 적용, 서버에서만 쿠키 접근 허용

! Express => '/apis/artworks/id' 동적 라우트로 프론트엔드로 json 파일 전달. API 이미지는 호스팅 플랫폼에 업로드 후 url 가져옴

! Mongoose => static 메소드 + mongoose pre hook으로 데이터베이스 저장 이전 비밀번호 해싱 적용.

! Express => email + findOne DB 유저 서칭 후 결과에 따라 1) jwt 토큰 세팅 또는 2) 클라이언트로 error 오브젝트 전송

! Google Cloud Platform, Passport 미들웨어 활용 OAuth 로그인 (구글)

! Express <=> server side socket io 연결, 클라이언트에서 송신된 메시지 소켓 별 전달

! Express => 미들웨어 활용 CORS(*, GET, POST, DELETE) & aggressive cache(GET, 1년) 세팅, 커스텀 logger 미들웨어 작성.

! Docker : 서버 섹션 Dockerfile 작성 및 Docker hub 이미지 배포함, Docker-compose 연계.

1. CS

* 내용 : 주말 활용 공부 중. HTTP 프로토콜, 운영체제, 네트워크, 데이터베이스, 디자인 패턴, 웹 브라우저, 자료 구조 등.

* 정리 : <https://github.com/developerasun/myCodeBox-web/tree/main/myCSknowledge>

===== 이런 걸 고려해주세요 =====

* 면접 : 1차로 비대면 zoom / google meet 희망해요

* 복지 : 사내 복장이 편했으면 좋겠어요

