NGUYEN QUOC KHANH

SOFTWARE DEVELOPER

0948573111
nquockhanh0209v@gmail.com
Ha Dong, Hanoi
02/09/2001
Nguyen Quoc Khanh
nquockhanh0209

ABOUT ME

A young person who is able to quickly learn and apply new technologies. Seeking the chance to bring further success to the organization's products through back-end performance.
Short term: Become a Back-end developer with languagues: Python, Nodejs and learn micro service from scratch in 3 months.
Long term: Expert in not only software architechture but also system architechture in 5 years.

SKILLS

Back-end

- **Database(SQL, NoSQL):** Advance
- **Python(Flask, Django, Selenium)**: Advance
- **Nodejs(Nestjs, Expressjs)**: Advance
- **Solidity**: Advance
- **AWS ec-2**: Basic
- **Docker**: Intermediate
- **Queue Service (RabbitMQ, Redis)**: Intermediate

Front-end

- **Reactjs (Nextjs)**: Intermediate
- **HTML/CSS**: Basic

Soft skills

- Critical thinking
- Problem-solving
- English Written and Oral skills

WORK EXPERIENCE

Savvycom Software

R&D Intern • 6/2022- 12/2022

- Built a PoC HDWALLET app back-end to create several wallet addresses from a provided private key (**Nodejs, MongoDB**).
- Built a PoC NFT Marketplace: Back-end (**Nodejs, MongoDB**), Front-end (**Reactjs**), Smart contract (**Solidity**).
- Maintained a PoC Cross-Chain Bridge service and handling service error: Back-end (**Nodejs, PostgresSQL, Redis**)
- Wrote documents and created sequence diagrams for the above projects.

Graduation thesis

• 1/2023- 6/2023

Building a betting website using Blockchain token for betting and using Chainlink nodes for transferring real-world data to Smart contracts to handling.

- Wrote documents and created sequence diagram for the above projects.
- Built website: Back-end (**Expressjs, MongoDB**), Front-end (**Reactjs**).
- Customized and Run Chainlink node using Docker.
- Wrote the smart contracts for storing betting data (**Solidity**).
- Wrote units test for each function of the project

Grade **9.2/10.0**.

MATA TECHNOLOGY SERVICES AND TRADING

Software Developer • 6/2023–Now

- Built NFT Marketplace:
  - Back-end (**Nestjs, PostgresSQL**):
    - Referral function.
    - Backup data to another database using a cron job.
    - Verify transactions: Buy, Mint NFTs and save to database.
    - Event functions: award the users who had bought and minted the number of NFTs in a week or a month.
    - Create API to create, read, update, delete NFTs and Buy NFTs.
  - Smart contract (**Solidity**):
    - Create contracts: NFT (ERC721), Token(ERC20).
    - Create contract buy NFT with Signature.
    - Implement upgradable contract.
    - Auction contract.

The website had been deployed to BSC Testnet. At least 100 users have joined this website, and nearly 500 NFTs minted in 2 months. This website has been also deployed to OpBNB Testnet and going to deploy to BSC Mainnet.

- Built Open Token Presale Platform:
  - Back-end (**Nestjs, PostgresSQL**):
    - Setup Database.
    - Backup data to another database using a cron job.
    - Verify transactions create tokens and presales.
    - Create API to create, read, update,delete tokens and presales.
  - Smart contract (**Solidity**):
    - Create contracts: Token(ERC20).
    - Create contract Presale and implement the upgradable contract.
  - Front-end (**Nextjs**):
    - Create tokens and presales.
    - Display the presales information.
    - Checking time for start/end presale.
    - Checking if presales are available.
    - Handle call to smart contract functions: Buy token, Claim token, Refund token, End presale and auto add liquidity.

A tester had tested the website and is waiting for release.

- Built the service to get all the NFTs and NFTs attributes on-chain:
  - Back-end **(Nestjs, MongoDB, RabbitMQ)**:
    - Handle crawl data from block 0 to current blocks for data of NFTs (ERC721 or ERC1155).
    - Handle get attributes of NFTs using RabbitMQ.
    - Classify the attributes of NFTs and send them to front-end or another service through RestAPI.

Service had run perfectly in BSC Testnet with at least 100k Nfts and NFTs attributes saved.

- Building tool for increasing impression to Twitter accounts and automation running the project accounts like a real human:
  - Back-end **(Python, MongoDB, Selenium, RabbitMQ)**:
    - Divide into 2 back-end services: Order service for receive and publish orders to Bot service. Bot service consumes the messages received and runs bot to increase impressions for accounts in orders.
    - Using Selenium for emulation of the account action as a person.
    - Using Selenium to crawl data on Twitter to get data of service-ordered accounts.
    - Integrate ChatGpt and Google Bard to generate comments for Twitter posts.
    - Using RabbitMQ to publish messages from the Order server. Consume the messages in the Bot server.
    - Create API for reading data of new bot accounts from CSV file and save to database.

The project is running and receiving up to 800 orders in 7 days.

EDUCATION

Bachelor of Science in Computer Science

Vietnam National University• 2019–2023

CERTIFICATIONS

TOEIC Reading and Listening: 860

IIG Vietnam. • 2023