HỒ TUẤN KHANH

CONTACT ME

https://hoofkhanh.netlify.app/

hotuankhanh20112016@gmail.com

https://github.com/hoofkhanh

www.linkedin.com/in/khanh-hồ-b006b3312

0329500036

18/08/2003

Male

SKILLS SUMMARY

●●●● Backend: Java (Spring boot, Spring cloud)

●●●● SQL: SQL Server, MySQL, PostgreSQL

●●●● NoSQL: Mongodb

●●●● Frontend: Nextjs, React, Typescript, Javascript, HTML, CSS

●●●● AI (Python, TensorFlow, Explainable AI)

●●●● Docker
Git
Amazon Web Services (EC2)

EDUCATIONAL PROCESS

Natural Resources and Environment University
Fourth year student
Software Technology Major
TOEIC: 670
GPA: 3.4

INTRODUCTION

I am a student at the University of Natural Resources and Environment in Ho Chi Minh City just finished year 4. My major is software technology. This is my portfolio: https://hoofkhanh.netlify.app/

WORK EXPERIENCE

Intern at “Bệnh viện nhi đồng 2”

- *Duration*: 3 months.
- *Contributions*: Perform small modules such as code debugging, code testing, learning hospital procedures, and coded the pneumonia prediction application under the guidance of the company's seniors.

PERSONAL PROJECT

Artist Circle Web (Microservice) - Ongoing

- *Summary:* A microservices-based web platform enabling artists to connect by GPS location, manage profiles and projects, with secure conversations using Elliptic-curve Diffie–Hellman key exchange algorithm.
- *Github link:* Repo
- *Services*: User, Artist, Search, Conversation, Notification, ConfigServer, Discovery.
- *Tech Stack:*
  - ***Frontend**: Nextjs, Shadcn (Tailwind), Redux, Apollo Client, WebSocket Client.*
  - ***Backend**:* Java, Spring Boot, Spring Cloud (Config, Eureka, LoadBalanced, Gateway), Graphql, RabbitMQ, Kafka, Websocket Server, MailDev, Flyway, Maven.
  - **Security**: Spring Security,JWT, ECDH key exchange algorithm.
  - ***Database & Caching**:* PostgreSQL, MongoDB, Redis, Elasticsearch.
  - ***Media**:* Cloudinary (image, audio).
  - ***DevOps & Tooling**:* CI/CD (Github Actions), Docker, Git, Github.
- *Contributions:*
  - Designed scalable microservices with Spring Boot, Spring Cloud, and GraphQL APIs.
  - Implemented secure authentication and authorization with JWT (access & refresh token) and ECDH key exchange for conversation privacy.
  - Used Elasticsearch for search, Redis for caching, and indexing for fast data access.
  - Used RabbitMQ for asynchronous communication between microservices.
  - Used Kafka for bidirectional communication between the conversation and encryption services.
  - Containerized with Docker, automated CI/CD via GitHub Actions.
  - Built SEO-friendly UI with Next.js and Shadcn, managed state with Redux, enabled real-time chat (WebSocket), and integrated GraphQL APIs (Apollo Client).
  - Managed code with Git/GitHub and wrote clean, scalable code.

Sound Service Web (Microservice)

- *Summary*: A microservices-based platform connecting customers with music artists. Key features include beat purchasing, artist hiring, job posting, real-time chat, reviews, and notifications.
- *Github link*: Repo
- *Services*: User, Artist, Customer, Job, Beat, Purchase Beat, Hire, Notification (real time), Payment, Favorite, Review, Conversation (real time), Config Server, Discovery.
- *Tech Stack*:
  - **Frontend**: ReactJS, Javascript, HTML, CSS.
  - **Backend**: Java, Spring Boot, Spring Cloud (Config, Eureka, OpenFeign, Gateway).
  - **Databases & Migration**: PostgreSQL (JPA + Flyway), MongoDB.
  - **Communication**: Kafka (services), WebSocket (realtime).
  - **Security**: OAuth2 (Keycloak).
  - **Email Handling:** Thymeleaf (templating), MailDev (testing).
  - **Tool**: Docker, Maven, Git, Github.
- *Contributions*:
  - Designed scalable microservices with Spring Boot, Spring Cloud.
  - Implemented secure auth using OAuth2 & Keycloak (access/refresh tokens).
  - Used Kafka (async services) and WebSocket (real-time messages, notifications)
  - Built responsive UI with ReactJS.
  - Containerized apps using Docker & Docker Compose.
  - Used Thymeleaf with MailDev to simulate and test email templates in dev environment.
  - Managed source code with Git and GitHub.

Antimicrobial Peptide Prediction (Transformer + XAI + Amazon EC2)

- *Summary*: Predicted antimicrobial peptides using a Transformer model with XAI (LIME) for interpretability, trained on EC2 GPU.
- *Github link*: Repo
- *Data link*: NCBI Proteins
- *Tech Stack*: **Python, Tensorflow, BiopPython, Transformer, LIME, Amazon EC2, Git, Github.**
- *Contributions*:
  - Crawled raw peptide data from NCBI.
  - Split peptides > 50 amino acid into 10–50 amino acid subsequences.
  - Extract features based on peptide sequences for model input.
  - Trained a Transformer-based model on 3 datasets (GenBank, non-GenBank, combined), all achieving accuracy close to 100%.
  - Used Amazon EC2 (GPU virtual machine) for efficient training.
  - Used LIME to explain model predictions and highlight contributing features. Managed source code with Git and GitHub.