

# Junyoung Lee

✉ [leejy31415@outlook.com](mailto:leejy31415@outlook.com)

🔗 [leejy12](https://leejy.net)

🌐 [Junyoung Lee](https://leejy.net)

🌐 <https://leejy.net>

## EDUCATION

### Texas A&M University

M.S. in Computer Science

Aug 2025 – Present

College Station, TX

### Korea Advanced Institute of Science and Technology (KAIST)

B.S. in Computer Science

Feb 2019 – Aug 2024

Daejeon, Korea

## EXPERIENCE

### Elice

Software Engineer

Oct 2024 – Jun 2025

Seoul, Korea

- Engineered a backend billing system with Python, FastAPI, and PostgreSQL; integrated payment providers like PortOne, Toss, and Stripe to automate card registration, recurring subscriptions, and on-demand charges.
- Designed and implemented a metric collection system using PostgreSQL and TimescaleDB to track and aggregate cloud resource usage (CPU, GPU, disk) per customer.

### Nuvilab

Software Engineer Intern (KAIST Co-op program)

Aug 2023 – Feb 2024

Seoul, Korea

- Developed a food tray scanning program using Tauri framework, Intel RealSense, and OpenCV libraries.
- Introduced a CI/CD process of the scanner application with AWS S3 and Bash scripts.
- Created a custom Ubuntu ISO with CUBIC to automatically manage hundreds of scanner devices.
- Managed the backend server written using Python FastAPI framework and ORM for MySQL database.

### Imprimed Korea, Inc

Software Engineer Intern

Jan 2023 – Feb 2023

Seoul, Korea

- Developed a Python tkinter program to analyze AML cells' ex-vivo sensitivity data.
- Developed a prototype website using C# and ASP.NET Core that suggests treatment type for multiple myeloma based on patient survivability prediction AI model.

## TECHNOLOGIES

### Languages

Python, C++, C, SQL, Bash, TypeScript, JavaScript

### Frameworks & Databases

FastAPI, PostgreSQL, SQLAlchemy, MySQL, TimescaleDB, Redis, React

### Infrastructure & Tools

Git, Docker, CI/CD, Nginx, Linux, Azure Static Web App

## PROJECTS

### MLB ScoRHEgami

2025

- Developed a real-time Twitter bot using Python and PostgreSQL to identify and post unique MLB box score combinations ("ScoRHEgamis") as games unfold.
- Built a companion web app using React + TypeScript, hosted on Azure Static Web Apps (<https://mlbscorhegami.com>).
- Inspired by NFL Scorigami; tracks rare Run-Hit-Error combinations across all MLB games.

## OPEN-SOURCE CONTRIBUTIONS

### Windows Terminal (C++, C++/WinRT, XAML)

2022

- Fixed an interaction bug in the settings page of Microsoft's Terminal application. ([PR](#))
- Enhanced the Settings UI by exposing a previously JSON-only configuration as a user-editable option, using XAML and the MVVM architecture. ([PR](#))