

# Andrew Chang-DeWitt

📞 (812) 266-0282

✉️ [hire-me@andrew-chang-dewitt.dev](mailto:hire-me@andrew-chang-dewitt.dev)

www [andrew-chang-dewitt.dev](http://andrew-chang-dewitt.dev)

🐙 [github.com/andrew-chang-dewitt](https://github.com/andrew-chang-dewitt)

## Education

### Illinois Institute of Technology, Chicago, IL

Bachelor of Science, Computer Science

August 2024 - December 2025

Completing coursework on CS & SWE fundamentals, including systems programming, programming language theory, compiler construction, & database organization w/ special interests in embedded software as well as parallel & distributed systems & objects.

### Wilbur Wright College, Chicago, IL

Associate in Engineering Science,  
Computer Science

August 2022 - May 2024

GPA: 3.8/4.0

Studied Python, C, OOP w/ C++ & Java; Networking (OSI, TCP/IP, UDP); Discrete Computational Structures; Calculus; Diff. Eq.; & Linear Algebra. Participated in dual enrollment program with IIT, meeting high academic standards to maintain eligibility & completing courses on Data Structures and Algorithms as well as Computer Organization & Assembly Programming at IIT.

## Experience

### exploreCSR @ Northwestern University

Racket AI LLM Software Testing Research Problem Solving

*Research Assistant*

March 2024 - ongoing

- Worked with Dr. Christos Dimoulas @ Northwestern & his team to assist with programming language research
- Assisted in the beginning stages of creating an LLM-based tool capable of generating test suites capable of capturing a high degree of possible bugs for a given software source
- Researched fuzzing techniques to use in tandem w/ LLM test generation to hopefully improve the generated tests

### Tinder (Match Group)

React Redux Redux-saga TypeScript NodeJS git GitHub Project Planning Feature Design

*Web Engineering Intern*

June 2022 - January 2023

- Led implementation planning for & worked with a cross-disciplinary team including engineering, product, & design to develop a new feature shipped on Tinder's web app in late Q3 of 2022, impacting tens of millions of users
- Participated in the code-review & release process as well as regular, informal Scrum ceremonies
- Assisted in migrating JavaScript typed in Flow to TypeScript across the application
- Improved the Web experience for tens of millions of users by identifying & shipping fixes for a11y & i18n bugs

### Indiana University-Purdue University at Indianapolis

.NET MVC Razor C# Python JavaScript TypeScript NodeJS Vue HTML CSS

*Assistant Web Master*

August 2021 - May 2022

- Updated outdated .NET Razor views from Bootstrap/JQuery to Indiana University's modern css framework, improving page load times
- Implemented additional features in HTML, CSS, & vanilla JS (or Vue, where applicable)
- Built dev tools in Python & TypeScript to automate data & file collection & formatting

# State of Indiana, Department of Family Resources

Leadership

Coaching & mentorship

Communication

Project planning

## State Eligibility Consultant

State of Indiana, Department of Family Resources, June 2019 - August 2020

## Team Lead

Knowledge Services, May 2018 - June 2019

## Eligibility Specialist

Conduent, Knowledge Services, August 2016 - May 2018

- Assisted clients in solving problems in the cases/applications, correcting their benefit eligibility & advised them on complicated policies & processes
- Trained 12 new employees & worked with a team of experienced employees to improve their customer service, communication, & problem-solving skills by designing & hosting group training sessions in coordination with 1:1 mentoring
- Collaborated with local office & regional management to improve business processes for efficiency & accuracy, saving 90+ minutes/day when generating reports & assigning case loads

# Projects

## Smoke

Python

FastAPI

IOT devices

Analog input processing

Asynchronous programming

OOP

 [github.com/andrew-chang-dewitt/smoke](https://github.com/andrew-chang-dewitt/smoke)

An IOT fan controller & thermometer using a Raspberry Pi Zero, an MCP3008 for reading thermistors as analog inputs, & a level-shifter for controlling a 12V PWM fan from the RPi's 3.3V GPIO pins. Reads temperatures & responds accordingly to maintain a target temperature in a charcoal smoker. Currently implementing features to share status/temps/history & allow controlling the device via a local WiFi web server.

## Hoops

Rust

WASM

Leptos

Python

FastAPI

PostgreSQL

Unit testing

FP

OOP

 [github.com/andrew-chang-dewitt/hoops-app](https://github.com/andrew-chang-dewitt/hoops-app)

 [github.com/andrew-chang-dewitt/hoops-api](https://github.com/andrew-chang-dewitt/hoops-api)

A budgeting application to empower my wife & I to better organize our financial future; inspired by a favorite (now defunct) fintech product. Backend built with a RESTful API written in Python with FastAPI & PostgreSQL, with parts as needed in Rust with Axum. Intended to be consumed with frontend currently being written as an isomorphic web application using SSR & hydrating on the client with WASM using Rust & Leptos.

## JSpec

Java

Unit testing

OOP

Data structures & algorithms

Library API design

CLI tooling

 [github.com/andrew-chang-dewitt/jspec](https://github.com/andrew-chang-dewitt/jspec)

A Java testing library, test runner, & cli built as a class final project. Building this was a fun way to dive deeper into Java by utilizing Java's powerful introspection & reflection capabilities to traverse objects & customize compilation output. JSpec was also a great excuse to practice data structures & algorithms by building a custom implementation of doubly linked lists & n-ary trees, complete with traversal algorithms & collection manipulation methods such as ``reduce()`, `map()`, & `forEach()```; used for storing test result state before rendering.

## DB Wrapper & AMQP Worker

Python

PostgreSQL

AMQP 0-9-1

Unit testing

OOP

Type systems & static typing

Library API design

Microservices

Data modeling & validation

 [github.com/cheese-drawer/lib-python-db-wrapper](https://github.com/cheese-drawer/lib-python-db-wrapper)

 [github.com/cheese-drawer/lib-python-amqp-worker](https://github.com/cheese-drawer/lib-python-amqp-worker)

A pair of async, typesafe libraries intended to be used in a microservice-based & Docker-orchestrated implementation of Hoops (which has since pivoted to a monolithic backend). First is an ORM & db connection/query library driven by Psycopg2 with runtime type validation in an intuitive API. Second is a library for building services that communicate over AMQP with a simple, Flask-like API.