

# Abhishek Addagatla

abhi.add0503@gmail.com | (630)-301-4776 | [abhishekaddagatla.me](http://abhishekaddagatla.me) | [github](https://github.com/abhishekaddagatla) | [linkedin](https://www.linkedin.com/in/abhishekaddagatla)

## EDUCATION

### Purdue University - West Lafayette, IN

Bachelors of Science in Computer Science (Machine Learning Track), Minor in Mathematics

Aug 2021 - Dec 2024

**Relevant Coursework:** Data Structures & Algorithms, Data Mining and Machine Learning, Introduction to Artificial Intelligence, Information Systems (DBS), Software Testing, Systems Programming, Computer Architecture, Programming in C

## SKILLS

**Coding Languages:** Python, HTML/CSS/JavaScript, Java, C#, C++, UNIX Terminal, Bash, TypeScript, React.js

**Technologies/Environment:** Shell Scripting, Linux, TensorFlow, Pandas, PyTorch, API Programming, Git, Agile/Scrum

## EXPERIENCE

### Software Engineer Intern

May 2024 - July 2024

#### Spot Inc. | React, Typescript, .NET, C#, SQL

- Developed an internal service to integrate third-party APIs for retrieving CO2 emission data from carriers, enabling customers to track and manage their carbon footprint.
- Applied Agile development practices using Jira to efficiently track project progress and bugfixes.
- Utilized Postman for rigorous API testing with 1,000+ concurrent requests, achieving less than 0.7% error rate.
- Implemented background task processing with Hangfire and Cron to offload the monthly recurring updates.
- Implemented Redis for caching, improving application performance and reducing overall API calls and database load.
- Leveraged Docker for containerization deployments on Azure, improving scalability and reliability.
- Improved code maintainability using dependency injection, promoting loose coupling in company code.
- Leveraged Azure DevOps to automate CI/CD build and deployment processes, reducing manual effort.
- Developed responsive user interfaces (UI) with React and Material-UI, enhancing user experience (UX).
- Applied CQRS code standards with .NET to separate read/write operations, enhancing system scalability.

### Full Stack Developer

May 2023 - Dec 2024

#### Purdue Center for Career Opportunities (CCO) | ASP.NET MVC, Bootstrap, C#, SQL, Ajax, JQuery, React

- File Uploader
  - Rebuilt the CCO's file uploader to be more efficient and remade the UI with Tailwind CSS to provide an engaging user experience.
  - Designed and implemented RESTful APIs with SQL and C# and Ajax + JQuery to manipulate data in JSON.
  - Website performance improved by 80%, resulting in a significant increase in daily traffic (10,000+ visits per day).
- Alumni Caller System
  - Purdue's CCO performs phone alumni reach-outs to check in on their post graduation life and career advancement. The center's feedback capture tool was rudimentary.
  - Built a visual analytics dashboard for tracking performance, improving team decision-making with data-driven insights.
  - The new tool handled 5,000+ records and was 90% faster with an intuitive and frictionless UI.

### IT Network Operations Engineer

May 2023 - Aug 2023

#### Purdue IT | Python, AWS Route 53, Bluecat, Selenium, LibreNMS API, Crontab, Splunk

- Successfully managed and implemented a project to migrate 2,500+ network devices to a new network management protocol by creating a python script using the LibreNMS API. Automated the migration project using Ansible playbooks.

## PROJECTS

### myMusicDB ([github](https://github.com/abhishekaddagatla/myMusicDB))

- A full-stack project where users can store and organize their music online in the cloud; using PostgreSQL for the database, HTML5/CSS/JS for the front end, and implementing RESTful APIs with Express/Node.js
- Leveraged the Drizzle ORM to align code with database structures and optimized database access by adding Indexes and using prepared statements.

### Custom Shell Interpreter

- Engineered a custom shell interpreter written in C and C++ by employing Flex and Bison to parse and execute complex commands, and to manage input, output, and error redirection. Implemented shell functionalities such as signal handling, environment variables, wildcards, and introducing a line editor for typo corrections and command history navigation.

### AI Dad Joke Generator ([github](https://github.com/abhishekaddagatla/ai-dad-joke-generator))

- Created a frontend using Bootstrap, CSS, HTML, and JS that prompted users to provide the AI with a topic for the dad joke. Utilized the OpenAI API to create requests to the GPT app in the backend and translate the JSON data into a displayable format using Jquery in the front end. Deployed the project on a Python web server made with Flask