

Akshar Patel

(669) 241 – 8220 * aksharpatel144@gmail.com * <https://akshar2401.github.io>

EDUCATION

Bachelor of Science, Computer Science

Graduated: Jan 2021

Minor, Mathematics & Statistics

GPA: 3.57

CALIFORNIA STATE UNIVERSITY SACRAMENTO

- **Selected Coursework:** Compiler Construction, Statistical Computing (R), Data Visualization, Data Mining, Algorithms, Database Systems, Software Engineering, Advanced Algorithms, Artificial Intelligence, Parallel Programming with GPUs, Cloud Computing.

SKILLS

- **Programming Languages:** Java, C, Python, R, C++, C#, Rust
- **Web Development:** HTML, CSS, JavaScript, TypeScript, Bootstrap, Angular, React, Django, NodeJS, ASP.NET Core, .NET Core, Redux
- **Data Science Libraries:** Numpy, Pandas, Scikit-Learn, Tensorflow, Keras.
- **Compiler Construction:** Lexical Analysis, Parsers, Semantic Analysis, Abstract Syntax Tree, Code Generation, Optimizations, Bison, Flex
- **Databases:** My SQL, Postgre SQL, SQL Server, MongoDB
- **Tools:** Git, Docker, AWS, Azure, Azure DevOps, JIRA, Selenium Web Driver, Visual Studio Code, IntelliJ, PyCharm, Eclipse, Visual Studio

RELEVANT WORK EXPERIENCE (More on: <https://www.linkedin.com/in/akshar-patel-378071122/>)

Software Engineer

Feb 2022 - Current

Microsoft, Atlanta GA

- Designing and implementing a **Language Server**, and an **Editor** with features like **intellisense**, **signature help**, **syntax highlighting**, and much more to allow usage of low code language PowerFx in Power Apps Canvas App Designer using **Visual Studio Code** based **Monaco-Editor Npm Package**, **Language Server Protocol**, **C#**, **React**, **TypeScript**, **Redux**, **SignalR**, **WebSockets**.
- Developing class library and visual studio extension to generate PowerApps Canvas App from Swagger/OpenAPI definition of Rest Apis using **C#**, **OpenAPI.NET**, **Visual Studio SDK**, thus **increasing monthly active users of Microsoft PowerApps Express Design feature by 60%**.
- Contributing to the development of [open-source Test Engine](#) that allows authoring of test cases using **Microsoft's low code programming language PowerFx** for different kinds of apps supported by Microsoft Power Apps using **C#**, **Playwright**, **JavaScript**.
- Contributing to the development of [Microsoft's open-source low code programming language PowerFx](#) by fixing outstanding bugs and contributing to features like lexical analysis, parser, semantic analysis, and IR translation using **C#** and **.NET Core**.
- **Hackathon:** Prototyped a Visual Studio Code Extension and Language Server to add support for low code language PowerFx and allow editing of Power Apps Canvas App source code in Visual Studio Code using **NodeJS**, **TypeScript**, **C#**, **Language Server Protocol**, **SignalR**.

Software Engineer II

Jan 2022 - Feb 2022

Butterfly Network Inc, Virtual

- **Fullstack Development:** Developed a token exchanging system to generate and exchange tokens for different identity providers using **Python**, **Flask**, **Flask-RESTy**, **SqlAlchemy**, **React**, **GraphQL**, and **AWS**.

Software Engineer

Jan 2021 - Oct 2021

Intel Corporation, Folsom CA

- **Backend Development:** Implemented and owned REST APIs for many aspects of managing Intel firmware configurations using **C#**, **.NET core**, **ASP.NET Core**, **MongoDB**.
- **C Header File Parser:** Developed a parser using **C#**, **CppAst** to compile and extract Enum constructs from uploaded header files and map Enum members to configurations with name same as Enum names, **thus effectively replacing manual entries of Enum members with automatic importing**
- **Unit Testing:** Achieved more than **75% code coverage** by implementing effective unit tests using **Xunit**, **AutoFixture**, **Moq**.

Compiler Research Assistant Volunteer

Jan 2021 - Dec 2021

California State University Sacramento, Sacramento CA

- **Instruction Scheduling:** Implementing Parallel Ant Colony Optimization Algorithm to find optimal schedule for GPU target-based instructions with minimum register pressure cost using **C++**, **CUDA**, and **LLVM**.
- **Benchmarks:** Compiling and running floating point calculations and machine learning benchmarks with Parallel Ant Colony Optimization Instruction Scheduling algorithm enabled to gather and analyze its performance data.

Software Undergraduate Intern

July 2019 - Dec 2020

Intel Corporation, Folsom CA

- **Web Development:** Developed a configurator web app to manage firmware configurations using **Angular**, **TypeScript**, **JavaScript**, **Kendo UI**, **Bootstrap**, **HTML**, **CSS**, **Python**, **Django**, and **SQL Server**.
- **Compiler Construction:** Developed context free grammar and parser using **Python**, **Ply** for firmware configurations expressions of different data types. Implemented semantic analyzer for type checking. Built C language code generator to generate firmware build files from configurations.
- **Git Workflow Integration:** Integrated Git Workflow to manage configurations with configurator web app using **Python**, **Django**, **GitPython**, **Git**, **Bitbucket APIs**, thus resulting in **full automation of workflow and increase efficiency in managing changes to configurations by 90%**.
- **Algorithm Design:** Modeled different types of dependencies among different firmware configurations as **Multi-Graph data structure** and implemented optimized iterative Depth First Search to traverse the Multi-Graph to resolve dependencies.

PROJECTS (More on: <https://github.com/akshar2401>)

Mini C Compiler: (C, Bison, Flex, Python)

- Developed Mini C compiler with scanner, LR (1) Parser, Semantic Analyzer, AST Tree Generator, Code Generator and Local Register Allocator.

SAVIS: (Node Js, D3.js, Chart.js, JavaScript, Electron)

- A statistical educational tool to help intuitively understand statistical hypothesis testing and confidence intervals using simulations and visualizations.

Programming Language Detection: (Python, TensorFlow, Keras, Scikit-Learn, Numpy, Pandas)

- Designed and implemented a CNN model with 92% accuracy to detect the programming language based on the images of the code snippet.

Histogram: (CUDA, C++, Parallel GPU Programming)

- Implemented an efficient Histogram algorithm using privatization technique for an input of array of integers. 4096 Histogram bins use unsigned 32-bit counters that are saturated at 127.