

# Sai siva reddy Maddula

✉ maddulasaisivareddy@gmail.com ☎ 3614881962 📍 Tampa, FL 🔗 <https://o1mssr.online>

🌐 [linkedin.com/in/sai-siva-reddy-m-964504224](https://www.linkedin.com/in/sai-siva-reddy-m-964504224) 🐙 [github.com/Mpr9640](https://github.com/Mpr9640)

## Skills

---

Programming Languages: Python, C++, HTML, CSS, JavaScript, TypeScript.

Libraries & Frameworks: React.js, Next.js, NumPy, Pandas, TensorFlow, Keras, FastAPI, Scikit-learn, Spring Boot, Hibernate, Angular, Vue.js.

Databases & Tools: PostgreSQL, Docker, Git, GitHub, GitLab, Colab, Blender, Chrome Extension, Google Forms, Jupyter (implied by Colab).

Concepts & Methodologies: Machine Learning, Deep Learning (implied by TensorFlow/Keras), Linear Algebra, Data Structures & Algorithms (DSA), Computer Networks, DFA, NFA, Regex, 3D Modeling, UI/UX (UI), RESTful APIs, Microservices, CI/CD, Unit Testing, Front-end Testing (Jest, Cypress), Back-end Testing (JUnit, Mockito), Teamwork, Responsible AI Practices.

## Projects

---

### o1 Application Autofill Extension,Phase 1(Part of Ongoing Project)

*Tech Stack: HTML, CSS, JavaScript, JSX, React, FastAPI, Python, Alembic, PostgreSQL, Chrome Extension*

- Developed a full-stack application with user authentication (account creation, email confirmation, login) and data storage using FastAPI and PostgreSQL.
- Built a Chrome Extension that detects job application forms, highlights a popup icon, and auto-fills fields by intelligently mapping backend user data to input elements.
- Implemented secure data fetching using tokens, with robust form parsing across varied layouts and input types.
- Managed database schemas and migrations using Alembic.

### Personal Portfolio Website, My Description

*Tech Stack: HTML, CSS, JavaScript*

- Designed and developed a dynamic, responsive personal portfolio to showcase skills, projects, and accomplishments.
- Implemented a modern UI with HTML for structure, CSS for styling, and JavaScript for interactivity, including animations, form validation, and dynamic content rendering.
- Ensured cross-browser compatibility and optimized mobile responsiveness for consistent user experience across devices.
- Demonstrated strong front-end development capabilities and provided an engaging platform for potential employers and clients to explore professional work.

### Fairness Evaluation using TensorFlow Fairness Indicators – ACS Dataset

*Tech Stack: TensorFlow, TensorFlow Model Analysis (TFMA), Fairness Indicators, Python, Pandas*

- Conducted a fairness audit on a machine learning model trained on the **ACS (American Community Survey)** dataset to detect and mitigate **bias across demographic groups** (e.g., gender, race, income level).
- Used **TensorFlow Fairness Indicators** to evaluate model performance metrics like **accuracy**, **false positive rate**, and **equal opportunity** across sensitive attributes.
- Integrated **TFMA (TensorFlow Model Analysis)** to compute and visualize **per-slice metrics** for better interpretability and transparency.
- Identified disparities in model performance among subgroups and adjusted data preprocessing, threshold tuning, and sample reweighting to reduce bias.
- Presented results through **interactive fairness dashboards**, highlighting areas where the model failed to maintain equitable performance.
- Demonstrated the use of **responsible AI practices** to ensure fairness and accountability in predictive models built on real-world census data.

### Transfer Learning & Multiclass Classification – Fashion MNIST Dataset

*Tech Stack: TensorFlow, Keras, NumPy, Matplotlib, Scikit-learn*

- Implemented **transfer learning** by training a base convolutional neural network (CNN) on 5 classes from the Fashion MNIST dataset and reusing learned features for a new target task involving the remaining 5 classes.
- Applied **data augmentation** techniques (e.g., horizontal flip, width/height shift) using ImageDataGenerator to increase training data diversity and reduce overfitting.
- Selected a **limited dataset** for the target task (10 samples per class) to simulate few-shot learning and validated model generalization on unseen classes.
- **Froze initial CNN layers** and fine-tuned the final dense layers for the target classification task, showcasing effective feature transfer.
- Measured and compared **training/validation accuracy and loss** before and after augmentation to highlight its impact on performance.

### Data Augmentation & Performance Evaluation – MNIST Dataset

*Tech Stack: TensorFlow, NumPy, Matplotlib, Scikit-learn*

- Applied **data augmentation** techniques to the MNIST handwritten digit dataset to increase dataset diversity and improve model robustness.
- Augmentation strategies included **random rotation**, **shifts**, **zoom**, and **flipping**, implemented using ImageDataGenerator from TensorFlow.

- Trained a baseline model without augmentation and recorded validation loss and accuracy metrics.
- Retrained the model on the augmented dataset and **compared performance**, observing a **decrease in validation loss** and improvement in generalization.
- Visualized and analyzed the impact of data augmentation on model training and validation using matplotlib.
- Demonstrated the effectiveness of augmentation in **reducing overfitting** and enhancing model performance on unseen data.

### Multiclass Classification – Fashion MNIST Dataset

*Tech Stack: TensorFlow, Scikit-learn, NumPy, Matplotlib*

- Developed a multiclass image classification model using **TensorFlow** to classify clothing items from the Fashion MNIST dataset into 10 categories.
- Preprocessed and normalized image data using NumPy, and split the dataset into training and test sets using train\_test\_split from Scikit-learn.
- Designed and trained a neural network with multiple layers, optimizing performance using appropriate activation functions and loss metrics for multiclass classification.
- Visualized training progress, accuracy, and sample predictions using matplotlib.pyplot to evaluate model effectiveness.
- Achieved strong classification performance and improved accuracy by fine-tuning model architecture and hyperparameters.

### ML Linear Regression – Chicago Taxi Dataset

*Tech Stack: Colab, Python, NumPy, Pandas, Keras, Plotly, Seaborn*

- Built a supervised learning model using **linear regression** to predict outcomes from the Chicago Taxi dataset.
- Used pandas and numpy for data preprocessing, feature selection, and identifying high-impact columns influencing target values.
- Refined the dataset by removing irrelevant features and trained the model using **Keras** with adjusted hyperparameters such as epochs, learning rate, and optimizers to improve accuracy.
- Visualized data trends, model training, and predictions using plotly.express  $\mathcal{E}$ , plotly.graph\_objects, and seaborn to provide interactive and static graphical insights.
- Performed model evaluation by predicting outcomes on new data and calculating overall accuracy to assess performance.

### Fire Flame Detector, Detects the fire and makes noise.

*Tech Stack: Arduino Uno, IR Flame Sensor, MQ-2/MQ-135 Smoke Sensor, Buzzer, LED, Resistors, Breadboard, Jumper Wires*

- Built a fire detection system using Arduino Uno that identifies flame and smoke using IR Flame Sensor and MQ-2/MQ-135 Smoke Sensor.
- Triggered immediate audio-visual alerts (buzzer sound and LED indication) when fire or smoke was detected.
- Designed and assembled the circuit on a breadboard with appropriate resistors and jumper wires, powered via USB/adaptor.
- Demonstrated real-time detection capabilities for fire safety applications using embedded hardware and sensor integration.

## Professional Experience

### CORTracker, Software Engineer

03/2025 – present

- Developing responsive and dynamic user interfaces using HTML, CSS, JavaScript, Typescript, React, Angular, or Vue.js.
- Building robust backend systems using Java, Spring Boot, Hibernate, and RESTful APIs for seamless data processing.
- Working with SQL (MySQL, PostgreSQL) to efficiently store and manage application data.
- Creating and integrating RESTful and SOAP APIs for smooth communication between frontend and backend systems.
- Designing and deploying micro services-based applications for modular and scalable system development.
- Using Git/GitHub/Gitlab for source code management and collaborating with cross-functional teams.
- Writing unit tests (JUnit, Mockito) for backend and Jest, Cypress for front end to ensure high code quality and performance.
- Deploying applications on cloud platforms like AWS, Azure, or Google Cloud and managing CI/CD pipelines.

### Datics Inc, Software Engineer (Internship -> Contract)

01/2024 – 03/2025

- Execute full software development life cycle (SDLC).
- Develop flowcharts, layouts and documentation to identify requirements and solutions.
- Write well-designed, testable code.
- Produce specifications and determine operational feasibility.
- Integrate software components into a fully functional software system.
- Develop software verification plans and quality assurance procedures.
- Document and maintain software functionality.
- Troubleshoot, debug and upgrade existing systems.
- Deploy programs and evaluate user feedback.

## Education

### Auburn University at Montgomery, Master of Science in Computer Science

08/2022 – 05/2024  
Montgomery, United States of America

- Graduated with a CGPA of 3.83.
- Developed and designed websites using HTML, CSS, JavaScript, and React focusing on responsive and user-friendly interfaces.
- Worked on neural network projects within machine learning, applying advanced algorithms to solve complex problems.
- Ran robust and efficient C++, and Python (OOPS) and Attended lab sessions to gain practical knowledge for emphasizing best practices and maintainability.
- Gained Knowledge in data structures and algorithms to efficiently organize and manage a large amount of data.
- Implemented machine learning code in Google Co-lab, enhancing data analysis and model training efficiency.
- Possess strong SQL skills for querying and managing databases.
- Completed projects on 3D Modeling Using the Blender Application.
- Experienced with NFA, DFA, and Regular Expressions.