

Elden Deguia

(409) 338-6521 | eldendeguia@outlook.com | [linkedin.com/in/elden-deguia](https://www.linkedin.com/in/elden-deguia) | github.com/ejd230001

EDUCATION

University of Texas at Dallas

Bachelor of Science in Computer Science - 4.0 GPA - Dean's List

Richardson, TX

Aug 2023 – Dec 2026

- **Coursework:** Data Structures & Algorithms, Computer Architecture, Operating Systems, Database Systems, Systems Programming in UNIX, Software Engineering, AI, Advanced Algorithm Design & Analysis, Computer Networks

TECHNICAL SKILLS

Languages: Python, JavaScript/Typescript, HTML/CSS, Java, C, C++, C#, SQL, Bash Scripting

Libraries/Frameworks: FastAPI, Flask, React, Express.js, TensorFlow, Pytorch

Developer Tools: Git/Github/Gitlab, CI/CD, Codex, AWS, Unix CLI, REST APIs, VS Code, Jupyter Notebook

EXPERIENCE

IDT&S GenAI Intern

The Boeing Company

May 2026 – August 2026

North Charleston, SC

- Developed components of a multimodal image-tagging pipeline leveraging OpenAI CLIP and GPT models to automate aircraft classification and metadata generation, reducing average tagging time from 5 minutes to 15 seconds per image.
- Built an internal dashboard that integrates with the GitLab API, providing project managers, maintainers, and developers with real-time visibility into active GenAI use cases and development progress, saving teams an estimated 5+ hours per week in manual tracking and status reporting.
- Supported the end-to-end deployment of a RAG use case through Boeing's established CI/CD pipelines, contributing to integration and testing and saving new team members approximately 3 hours of training each.

PROJECTS

AI Disaster Assessment | [Visit](#) | [GitHub](#) | *Python, FastAPI, OpenAI API, AWS*

Jan. 2026 – May 2026

- Lead a team of five in developing a full-stack disaster assessment platform that visualizes disaster imagery on an interactive map, overlaying AI-generated damage assessments onto individual properties at their real-world locations.
- Developed the **FastAPI** backend, implementing endpoints that retrieve disaster imagery and information from **AWS S3** and integrates with the **OpenAI API** for automated damage assessment.
- Developed a Python script that parses provided image metadata files to determine the real-world coordinates of disaster imagery.
- Contributed to the application's **RAG** pipeline, supporting document-based question answering.
- Led weekly team discussions to review progress and coordinate development priorities, while facilitating communication with the project sponsor to clarify requirements and share updates.

Medvisor | [Visit](#) | [GitHub](#) | *Python, React, Flask, Tensorflow, OpenCV*

Sept. 2024 – Dec. 2024

- Contributed to the development of Medvisor in UTD's AI Society, a full-stack AI-powered application for detecting and diagnosing spinal abnormalities from MRI scans.
- Preprocessed over 300 raw 3D MRI scans, extracting slices and converting them into PNGs for model training.
- Fine-tuned a pretrained **ResNet50** classification model using **TensorFlow** to predict spinal abnormalities and assisted in training a **U-Net** segmentation model for intervertebral disc localization.
- Contributed to the full-stack application by developing a **Flask** backend API along with several **React** components to deliver interactive model predictions.
- Awarded 1st at AIM Night, a society-wide showcase where projects were presented to judges for evaluation.

League of Trivia | [Visit](#) | [GitHub](#) | *React, Node.js, Express, PostgreSQL, AWS*

Oct. 2025 - Nov. 2025

- Developed League of Trivia, an interactive full-stack trivia game based on the game League of Legends, featuring multiple categories and difficulty levels.
- Built a responsive **React** frontend and an **Express** REST API that dynamically generates unique questions.
- Designed and implemented a **PostgreSQL** database containing structured information on characters, professional esports players, and in-game equipment, optimized for flexible and efficient question retrieval.
- Deployed the full application using **AWS S3 + CloudFront** for the frontend and **AWS EC2 + RDS** for the backend, creating a fully hosted, production-ready web experience.