Aditya Archunan Anand

517-329-2260 | anandad1@msu.edu | linkedin.com/in/aanandadi/ | github.com/AdityaAA2004

Education

Michigan State University East Lansing, MI
B.S Computer Science – GPA 3.8 *Expected Graduation May 2026*

Technical Skills

Languages: HTML, CSS, JavaScript, Python, PHP, MySQL, Java, Kotlin, C++, TypeScript, Shell Scripts

Frameworks & Libraries: React, React Native, Node.js, Express.js, Flask, Drupal, Jetpack Compose, TensorFlow, Scikit-learn, NumPy, Pandas, Seaborn, JEST

Tools & Technologies: AWS, Azure, Firebase, MongoDB, Git, GitHub, Docker, Linux, Fast API, Android Studio, Figma, Jupyter, Google Colab, MATLAB

Experience

Application Developer | *Angular, Java, MySQL, Git, Bitbucket, Apache Tomcat, Oracle Database, Spring Boot, Agile*
Delta Dental of Michigan December 2024 -Present

- **Collaborated** with fellow interns and full-time employees on the Group Admin Audit feature project, following the **Agile workflow**. **Developed** and **maintained** multiple audit processes for various aspects of insurance contracts within Roosevelt Solutions, a platform used nationwide by over **22 million users**.
- **Proposed** API design MD Docs containing the URL endpoint, request parameters, model response, error conditions, security coverage, and speed and performance details. **Developed** efficient, accurate, and complex **SQL** queries to retrieve the audits from the database. **Developed** the API endpoint using **Spring Boot** with a **Test-Driven Development** approach.
- **Collaborated** with other full-time developers and **developed** the API endpoint using **Spring Boot** with a **Test-Driven Development** approach and developed the UI in **Angular** for the admin users to access these audits whenever needed.
- **Designed** and **implemented** a reusable **AI agent** using **Azure OpenAI and Azure Document Intelligence** services exposed through a **Fast API** application to extract and data from enrollment forms and documents of different formats, to automate eligibility determination and streamline surrounding administrative workflows.
- **Designed and implemented** an intuitive and elegant UI component using **Angular** in Roosevelt Solutions for admin users to interact with the AI agent in a seamless manner.

Student Web Developer | *PHP, MySQL, PostgreSQL, Devilbox, Object-Oriented Programming, Docker, Git, Gitlab, Linux*
Michigan State University Jan 2023 - Present

- **Developed** solutions and fixes using **PHP, MySQL, and Devilbox** to resolve 20+ technical issues in websites and assisted the team in 2 long-term projects under the Residential and Hospitality (RHS) Department using Drupal Content Management System (CMS) and Symfony framework.
- **Developed** a custom module in **PHP** using **Drupal Hooks** with **Object-Oriented Principles** to customize the output of the **XML feed** of the upcoming events at multiple venues in MSU and added it as a **git submodule** in our **Gitlab** repository and used it on different websites.

- **Collaborated** with a team of 6 developers in a project to rewrite the MSU Sports camps site from its old stack to the Drupal stack for better stability, performance, and better experience for more than 2000 end-users and site admins; **maintained** the sites by promptly solving bugs arising due to the refactoring.

PROJECTS

Targetist | *React Native, Expo, TypeScript, AWS EC2, Amazon EventBridge, AWS Amplify, AWS Lambda, Python, Shell Script, CI/CD, MongoDB, Node JS, Express JS, Git, GitHub, Agile development*

- **Collaborated** in a team of 4 on a startup idea to **build** an AI based task automation software strictly following the **Agile development cycle** and pitched the idea in several startup competitions to fellow students and Venture Capitalists along with product demos.
- **Developed** the User Interface of the mobile application using **MongoDB**, **Node**, **React Native** and **Expo Command Line tools** by implementing authentication using **AWS Amplify**, and calling the API for database operations and **published** the app on iOS App Store.
- **Collaborated and developed** the API to manage DB operations using Node.js & TypeScript incorporating best practices of OOP principle, following the MVC model, coupled with server-side authentication, authorization, and validation, with Test-Driven Development approach using Jest to write unit and integration tests.
- **Collaborated** with the team to develop the scheduling engine to efficiently score and schedule tasks in python, and setup a highly available, durable, and scalable architecture with **AWS services** (**Amazon SQS, Amazon EventBridge, AWS Lambda and Amazon API Gateway**), connected to the Mongo DB Triggers, and enabled **CI/CD automation** with GitHub actions workflow and **shell scripts.**

CrashEye Crash Detection System | *Mongo DB, Express JS, Node JS, React, Electron JS, AWS EC2, Git, GitHub*

- **Collaborated** with a team of 3 to develop an Al based software crash detection system using **MERN stack** (frontend) and **Python** (backend) which empowers developers to swiftly identify and categorize the type of software crash that occurred in users' system eliminating the need for manual report submissions and significantly streaming the development process.
- **Developed** the frontend using the **MERN stack** and executed it in the desktop using **electron JS**, by developing the authentication mechanism and the data extraction on an **Express JS** and **Node JS REST API** and building the client-side using **React JS** and using the **Axios** library to make different requests to the API and **hosted** it on **AWS EC 2** virtual server.

Amazon Clone | *Next JS, NextAuth, Redux, Stripe Payments API, Google Cloud, Webhooks, Firebase, Git, GitHub*

- **Built** a clone of the Amazon website using **Next.js** on the frontend, added products to the website using **Fakestore API**, **authenticated** users through the Google Authentication Provider from **NextAuth.js**
- **Created** a **Redux store** to contain the items added by the user to the basket, and used the library to keep the basket stored in the user's local cache to make it accessible any time they want, added **Stripe API** to the site to handle payments and order storage using **webhooks**.
- Used **Google Cloud** Functions to create an Application Default Credential using a service account key to access the Firebase database and display a user's previous orders.

ACHIEVEMENTS

- **International Tuition Grant (MSU) – 2022 – 2026**

- **AWS Certified Developer Associate**