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 3.75 *Expected Graduation May 2026*

Technical Skills

HTML, CSS, JavaScript, Python, PHP, MySQL, Java, Kotlin, Shell Scripts, C++, React, React Native, Node JS, Express JS, Drupal, Jetpack Compose, Git, GitHub, Docker, Linux, Android Studio, AWS, TypeScript, Flask, Firebase, Mongo DB, JEST, Figma, Jupyter, Agile, REST API, JEST, Google Colab, MATLAB, Tensorflow, Scikit Learn, Numpy, Pandas, Seaborn

Experience

Student Web Developer Jan 2023 - Present
Michigan State University

- **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).
- **Developed** a custom module in **PHP** using **Drupal Hooks** to customize the output of the **XML feed** of the upcoming events that would help developers of other technical teams on rendering the free slots available in the various event venues at MSU, 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; Assisted the team in testing the upgrade to Drupal 10 locally; **researched** and **documented** the errors I encountered along with fixes, easing the upgrade process on the production cluster.

Projects

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.

Leadership and Initiatives

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

- **Collaborated** in a team of 4 on a startup idea to **build** an AI based task automation software.
- **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 written by the other team members for database operations.
- **Collaborated** with the team to develop the machine learning model to efficiently score and schedule tasks in python, and setup the Mongo DB trigger using Amazon EventBridge to connect to an AWS lambda function containing the model code, whose CI/CD is set up with GitHub Actions workflow and basic shell script.
- **Hosted** our API written in Node JS on AWS EC2 Servers, and **automated CI/CD workflows with GitHub Actions** through **shell scripts** to automate pulling in changes committed to Version Control on the clones of the API repository present in the EC2 Server for hosting.