

# Vaishnavi Rai

Vaishnavi29rai@gmail.com | [vaish290 · GitHub](#) | <https://www.linkedin.com/in/vaishnavi-rai-a074ba194/>

## EDUCATION

**Indiana University Bloomington**  
Master's In Computer Science

2025  
GPA:3.6

## SKILLS

**Languages:** Python, JavaScript, TypeScript, PHP  
**Frontend:** React, React Native (Expo), HTML/CSS, Tailwind, Bootstrap  
**Backend:** Node.js, Express.js, Flask, REST APIs  
**Cloud & DevOps:** AWS (DynamoDB, Lambda basics), Google Cloud (Cloud Functions, Storage), Docker, Kubernetes, CI/CD  
**Databases:** MySQL, PostgreSQL, MongoDB, DynamoDB  
**Tools:** Git/GitHub, Elasticsearch, Kibana, Splunk, AppDynamics, Dynatrace, ServiceNow, Jira, Power BI, Tableau  
**Libraries:** Pandas, NumPy, scikit-learn, Matplotlib, Seaborn  
**Concepts:** Data Structures, Algorithm, System Design (HLD/LLD), Design Patterns

## EXPERIENCE

**Heartland Community Network**

USA

*Software Developer*

2025 - Present

- Built modular React Native (Expo) components using **JavaScript/TypeScript**, delivering pixel-perfect Figma-matched screens with shared design tokens and reusable UI patterns.
- Developed onboarding, authentication, and profile flows with **React Navigation**, dynamic forms, validation, and clean conditional UI.
- Integrated components with **Node.js/Express APIs**, ensuring reliable data handling, error states, and optimized rendering performance across iOS/Android.

**QualityKiosk**

India

*Site Reliability Engineer*

2022 - 2023

- Handled **100GB+** of logs every day, setting up **automated pipelines** in **Elasticsearch** that made data more reliable and cut manual work by 60%.
- Built **interactive Kibana dashboards** and worked with server teams to fine-tune queries, which sped up indexing, reduced troubleshooting time by 40%, and helped resolve issues faster.

**QualityKiosk**

India

*Performance Assurance*

2021 - 2022

- Led client-specific R&D initiatives using advanced data extraction techniques in **Splunk**, improving system integration efficiency by 30%.
- Resolved critical system issues, reducing downtime by 50% and increasing client satisfaction by 40%, while ensuring technical feasibility across operational features.

## PROJECTS

**Master Of Jokes**

2024

A full-stack joke-sharing platform where users can post, rate, and moderate jokes with secure access control.

- Built the web app using **Flask**, **SQLAlchemy**, **HTML/CSS/JavaScript**, enabling user accounts, submissions, ratings, and moderator workflows.
- Designed **HLD/LLD** architecture with **Python decorators**, **session management**, and **route guards** to implement fine-grained role-based access.
- Packaged the application using **Docker** and deployed on **Render** for reliable, production-ready hosting.

**Multi-Format File Conversion Engine On Google Cloud**

2024

A serverless file-processing system that converts documents (PDF, DOCX, etc.) via cloud functions.

- Implemented backend logic using **Flask**, **Google Cloud Functions**, and **Cloud Storage** to securely handle uploads, conversions, and signed-URL downloads.
- Developed a user-friendly, responsive frontend using **HTML/CSS** for file upload and format selection.
- Ensured safe file handling with **temporary storage**, unique filenames, and Python-based validations and error handling.

**Speech Emotion Recognition**

2020

A machine learning model that classifies emotions (happy, sad, angry, neutral) from speech audio recordings.

- Built emotion classifier models using **Python**, **TensorFlow**, and **Multilayer Perceptrons (MLPs)**.
- Enhanced audio quality with **noise reduction**, **MFCC feature extraction**, and data preprocessing techniques.
- Achieved ~85% accuracy across multiple labeled speech datasets.

**Public Transit Booking System**

2019

A basic booking system that supports route discovery, user accounts, and ticket management.

- Developed responsive web pages using **HTML/CSS and JavaScript**, ensuring clean UI behavior and smooth data flow with **MySQL**.
- Implemented route search, account creation, and booking logic with **SQL queries + JavaScript-driven UI** in a local environment.