JASH MERCHANT

jjm9801@nyu.edu | LinkedIn | New York

EDUCATION

New York University
M.S. in Computer Science, **GPA:** 3.6/4.0

Relevant Courses: Cloud Computing and Big Data, Software Engineering, Principles of Database Systems, Java, Design and Analysis of Algorithms, Operating Systems, Information Security & Privacy, Programming for Data Science

Awards and Honors: NYU Merit Scholarship ($7000/year)

Gujarat Technological University
B.E. in Computer Engineering, **CGPA:** 9/10, Oracle Certified Java Programmer (Passed SE-6 with 98%)

Activities and Societies: Member of IEEE SOU Branch, Google's Developer Student Club

SKILLS

C, Java, HTML, CSS, JavaScript, jQuery, Bootstrap, React.js, Node.js, Django, Flask, SQL, NoSQL, Git, AWS – S3, Lambda, API Gateway, DynamoDB, ElasticSearch, SES, EventBridge, Cloudwatch, Lex, Cloudshell, Rekognition, SageMaker, CloudFormation;

PROFESSIONAL EXPERIENCE

Green Action Studio, New York **Jan '23 – May '23 (5 months)**
Software Engineer Intern

1. Developed private charging feature from scratch utilizing serverless architecture using AWS CDK. Curated comprehensive unit tests using Vitest library ensuring 100% coverage as well as conducted integration testing for the entire feature.
2. Created numerous Kafka topics, forward compatible event schemas and change stream handlers to listen to MongoDB events and to forward them to Kafka in real-time.
3. Employed Kafka MSK to create producers and consumers and also migrated specific consumers to subscribe to new topics, replacing old ones.
4. Implemented dependency injection to streamline code structure as well as unit and integration testing processes.
5. Led the development of comprehensive API documentation for all service APIs utilizing the OpenAPI specification in Swagger.

Pikkal & Co – B2B Podcast Agency, Singapore **Feb '21 – March '21 (2 months)**
Full Stack Developer Intern

1. Worked in teams following Agile Development Practices.
2. Aided in redesigning and optimization of the company website using React to increase web traffic by 10%.
3. Used Google's PageSpeed Insights API to analyze performance and visualized it on the Geckoboard.
4. Developed web application using Django for sentiment analysis and integrated it with AWS Comprehend.

Harvard University **January '21 (1 month)**
CS50 Mentor

1. Assisted in answering questions from attendees via Zoom and EdStem during and after Harvard University's online CS50W (Web Programming with Python & JavaScript) seminars.

PROJECTS

HomeFix ***(Rotational Scrum Master)***
Technologies: Python, Django, Bootstrap5, JavaScript, Stripe API, Google Maps API, SQLite, GitHub, Travis CI/CD, Flake8, Heroku

1. HomeFix is a full stack, mobile responsive social application where users can exchange household services.
2. Programmed more than 70% of the application's frontend.
3. Integrated Stripe checkout portal and added custom discount coupons for users. Moreover, devised transaction system for users to pay/earn coins for services.
4. Collaborated in curating unit tests for the application to ensure more than 85% coverage.

Intelligent Search Album
Technologies: AWS (Lambda, API Gateway, Elastic Search, S3, Rekognition, Lex, CodePipeline, CloudFormation), Python, JavaScript

1. Developed a photo album web application using AWS services such as Lex, ElasticSearch, and Rekognition to enable natural language search for people, objects, actions, landmarks and more using both voice and text.
2. Automatically indexed photos using a Lambda function that detects labels in the image and stores relevant metadata in an ElasticSearch index.
3. Created a second Lambda function and an Amazon Lex bot to handle search queries and query the ElasticSearch index for relevant photos.

Flack
Technologies: Python, Flask, Socket.io, Local Storage, Bootstrap4, JavaScript

1. Flack is an online messaging service made using Flask and socket.io, similar in spirit with popular application like Slack.
2. Users will be able to sign in with a display name, create channels (i.e. chatrooms) to communicate in, as well as see and join existing channels. Once a channel is selected, users will be able to send and receive messages with one another in real-time.

For more projects and live previews, please visit ***https://github.com/jashmerchant***