Yennam Santhosh Kamal Murthy

Dayton City, OH | 774-701-9207 | yennamsanthosh82@gmail.com

EDUCATION – Campbellsville University, Louisville, Kentucky
Master of Science, Computer Science

Under Graduation
MVSR Engineering College, India   **June 2021**
Bachelor of Engineering, Computer science and engineering

SKILLS

- Strong Knowledge on Data Structures and algorithms, System design.
- **Frontend Technologies:** Type Script, React JS, Next JS, Android Development, Vue JS, Angular JS, HTML, JQuery, CSS, Bootstrap, Material UI, Tailwind, Webpack.
- **Backend Technologies:** Spring Boot, Node JS, Golang (basics), Flask, PHP, Laravel.
- **Programming Languages:** Java, JavaScript, C language, C++, Kotlin, Python.
- **Database:** MySQL, Postgres, MongoDB, Elastic Search, Redis, Cassandra**.**
- **IDE's:** VS code, IntelliJ, Android Studio.
- **Operating System:** Windows, Linux, macOS.
- **Version Control:** git, GitHub, Gitlab, Bit bucket.
- **Cloud:** AWS.
- **Containerization Tools:** Docker, Kubernetes.
- **Testing Libraries:** Junit, Mockito, Jest, Cypress, JMeter, Roboelectric.
- **Others:** Micro Services, OAuth Authentication, Rest API, Graph-QL, Web sockets, Computer Networks, Database Internals, Shell Scripting, Spring Security, In memory databases, Kafka, RabbitMQ, Postman, Swagger, Splunk, Graylog, Kibana, Rancher, Grafana, Firebase, Web Servers such as Ngnix and Apache.
- **My GitHub Repository**: Link

WORK EXPERIENCE

Zemoso Technologies Pvt Ltd, Hyderabad, India   **Aug 2021- Jan 2023**
Software Developer

- Worked on developing a web UI for a Privacy API solution using ReactJS and GraphQL.
- Developed a middleware (Backend for Frontend) for seamless communication between the ReactJS frontend (using GraphQL) and the NodeJs backend.
- Established seamless communication between the Node JS layer and a Golang backend via GRPC.
- Integrated Okta for authentication across multiple environments, including development, staging, and production, ensuring a secure and streamlined authentication process.
- Provided clients with the application, allowing them to access it through customized subdomains such as client1.mywebsite.com and client2.mywebsite.com, enhancing user experience and brand identity.
- Developed and maintained smoke and regression test cases, scheduled to run daily in the morning. Test reports were automatically deployed to an AWS S3 bucket, and the results, along with S3 test report links, were communicated to Slack channels. This proactive testing approach helped identify and address issues promptly, ensuring the robustness and reliability of the application.
- frameworks for the web UI, middleware, and SDK components, ensuring high code quality and reliability.

- Developed SDKs for same Privacy API Solution, targeting Android, Golang, Java, and Node.js platforms.
- Developed custom widgets for Credit Card, CVV, Expiry date, year, month, and name in client-side SDKs. These widgets included default validations and a customer component that allowed clients to provide multiple regex for validation across all components.
- Ensured that the Android, iOS, and JavaScript widgets complied with PII and PCI standards. The design ensured that when clients used our library to render our components, extracting text from protected information like CVV, card number, and expiry date was restricted.
- Implemented additional security measures by exposing keyboard listeners for these text fields. In the development environment, the values entered by the customer would be accessible through these listeners. However, this functionality would be restricted in the production environment, adding an extra layer of protection to sensitive information.
- Implemented unit testing and integration testing using Jest, Mockito, Junit, Jmeter, Roboelectric testing.
- Developed a CI/CD pipeline for all SDK projects using GitHub actions.
- For the CI/CD process, GitHub Actions were employed to deploy artifacts to GitHub Packages. Test cases and code coverage were executed when pull requests were raised. Merging code to the release branch triggered artifact deployment to the dev environment. Subsequently merging the release branch to the master branch led to deployment in the production environment.
- Utilized logging tools like graylog in Node.js and Apollo GraphQL for effective logging and debugging in application development.
- Collaborated with cross-functional teams to gather requirements, provide technical expertise, and deliver solutions that meet client needs.
- Integrated Codecov into the development workflow to measure test coverage when opening pull requests on GitHub.
- Conducted code reviews, identified, and resolved issues, and implemented best practices to ensure high-quality codebase and maintainable software.
- Participated in a hackathon, leveraging a client application while working for the company.

INTERNSHIP
Zemoso Technologies Pvt Ltd, Hyderabad, India   **Nov 2020- July 2021**
Software Developer

- Played a key role in developing a Spring Boot application seamlessly connecting to a AWS RDS database.
- Leveraged Spring Security with multiple filter chains for crafting secure authentication mechanisms.
- Implemented a filter chain for JWT authentication and another for connecting to OIDC (using OAuth).
- Dual authentication approach allowing different APIs to undergo either JWT or OIDC authentication based on specific requirements.
- Demonstrated proficiency in using profiles for Spring Security, tailoring security configurations to different environments.
- Provided REST APIs and GraphQL using Spring Boot, showcasing versatility in catering to different communication paradigms.
- Well-versed in connecting to web sockets and utilizing Redis for accelerated data access.
- Possess in-depth knowledge of structuring projects with multiple modules, each having its own pom.xml file.
- Efficiently managed these modules through the root pom.xml, utilizing the <modules> tag to encompass various sub-modules.
- Modules include Spring Boot applications, AngularJS, and more.
- Notable achievement includes generating artifacts from a Spring Boot application, deploying them to JFrog.
- Incorporated deployed artifacts into a Docker image and subsequently deployed to Google App Engine.
- Showcased end-to-end proficiency in managing the entire software delivery lifecycle.

GRADUATION PROJECT
Full stack Project   **Jan 2023 - May 2023**

- Designed and developed a robust web application using Flask, incorporating Flask-WTF for seamless frontend form handling. Flask was employed to manage server-side logic and create API endpoints for backend functionality.
- Established a secure and efficient connection between the Flask backend and MySQL database using the SQLAlchemy library. This implementation ensured optimal data management and retrieval within the web application.
- Utilized Docker to containerize the application, streamlining deployment processes and enhancing scalability across various environments.
- Implemented GitHub Actions for continuous integration and deployment, automating the testing and deployment pipelines for the Flask web application.
- Deployed the web application on AWS EC2, leveraging SSH for secure remote access and configuration. Putty was employed as an additional layer of security to facilitate seamless remote interactions.
- Managed file transfers between the local development environment and the AWS EC2 instance using FileZilla, ensuring efficient deployment and timely updates to the application.

COMPETITION PROJECT
Distributed File System Enhancement   **Aug 2023 – Dec 2023**

- Spearheaded a team project during the master's program, focusing on optimizing a distributed file system for higher consistency and availability.
- Collaborated in a group setting, with responsibilities including the development of Flask-based client and backend applications, alongside a file server crafted in Go lang.
- Implemented Docker for seamless containerization, including the creation and deployment of images for client, backend, and the file server, streamlining the project environment.
- Addressed performance bottlenecks by incorporating load balancers and introducing Redis caching mechanisms.
- Elevated the throughput of the system by optimizing API calls, particularly in file creation and deletion processes, ensuring improved consistency and availability.
- Utilized strategic API call loops to simulate increased load, enabling comprehensive testing of system responsiveness.
- Achieved enhanced file management capabilities by facilitating effective communication between Flask backend and the Go lang file server.
- Demonstrated problem-solving skills by identifying and rectifying issues leading to low throughput, ensuring the project met high-performance benchmarks.

FREELANCE
Android Application Development   **March 2020 - Dec 2020**

- Successfully released the Rationwala application on the Google Play Store, taking on the role of the frontend developer.
- Utilized the Kotlin programming language to meticulously implement the frontend, ensuring a seamless and interactive user experience.

- Collaborated closely with UI/UX developers, translating design concepts into reality using XML for Android widgets, including Linear Layout, Grid Layout, Relative Layout, Tab Layout, Scroll View Buttons, Text View, Adapters, Spinners, and incorporating loading animations etc.,
- Implemented custom adapters for efficient display of items retrieved from databases, enhancing the overall user interface.
- Skillfully implemented push notifications functionality using Firebase Cloud Messaging, facilitating real-time communication with users, and boosting user engagement.
- Demonstrated expertise in implementing authentication mechanisms, including Google Sign-In, Firebase Authentication, and integration with third-party messaging services for OTP verification.
- Integrated the application seamlessly with Google Maps, leveraging the Google Maps API key to provide users with location-based services.
- Implemented RESTful services using the Retrofit library, ensuring efficient communication between the frontend and backend components of the application.
- Employed broadcast receivers to listen for system-wide events, enhancing the app's ability to respond to various triggers.