A blue hexagon with white text

Description automatically generatedVeda Kola

[veda.vnk@gmail.com](mailto:veda.vnk@gmail.com)

+1 646-809-5182

https://www.linkedin.com/in/vedavarun

PROFESSIONAL SUMMARY

* AWS Certified **Python Developer** with a passion for **full-stack development,** and **web application deployment,** dedicated in building robust, scalable applications using frameworks and libraries **Django, Flask,** and **Pandas.**
* Proficient in generating business-specific datasets, conducting large-scale **data analysis** using **Python Pandas**, and engineering Python routines to optimize **user experience**.
* Collaborated effectively on **Python** applications for risk management, contributing to the development of **Django-based** web applications and implementing robust client-side validation with **JavaScript**.
* Exhibited expertise in constructing dynamic data tables, leveraging **PyQt** for enhanced user interface, and conducting thorough testing of **RESTful APIs** within **Agile** methodologies.
* Made significant contributions to web application development through **Python scripting** for **data processing** and **visualization**, following Test-Driven Development (**TDD**) practices diligently.
* Conducted **API** testing using tools like **Selenium, Postman**, ensuring the reliability of **RESTful web services**.
* Maintained databases across **MySQL, MongoDB,** and **Graph QL** platforms, ensuring data integrity and performance.
* Deep knowledge of **AWS** services EC2, S3, EKS, CDK, RDS, SQS, SNS, Auto Scaling, DynamoDB, Lambda services.
* Demonstrated proficiency in programming languages such as **Python** and **Ruby,** along with frameworks like **Django**.
* Proficient in development tools **PyUnit, Git, Jenkins** for efficient project management and version control.
* Experienced in **version controls** with **Git, SVN** and **GitHub,** and handled microservices architecture with **Docker, Kubernetes,** maintained infrastructure with **Terraform,** dealt with CI/CD pipeline orchestration using **Jenkins.**
* Experienced in front-end development **HTML, CSS, JavaScript, ReactJS**, **TypeScript** enhancing UI and interactivity.
* Experienced in using **PyDev, PyCharm** for debugging and worked with **Sublime Text 2, Vim Editors** for coding.
* Skilled in **Shell** scripting, usage of **Docker** for containerization and **Kubernetes** for orchestration, ensuring scalable, high-performance, and reliable deployments across cloud environments (AWS).
* Proficient in handling microservices using **FastAPI** and **Django**, improving API performance and system modularity.
* Experienced with **PySpark** for large-scale data processing and analysis, optimizing data workflows and enhancing the efficiency of **ETL** processes, handled **ETL** migrations into AWS **RedShift**.
* Demonstrated leadership in managing projects across the software development lifecycle **(SDLC)**, particularly in the development of monitoring and notification tools using **Python**.
* Familiar with **Agile** methodologies, actively participated in sprint cycles, and iterative development processes.
* Worked in diverse Operating Systems like **Windows, Linux**, and **UNIX**, adapting to deployment scenarios.

TECHNICAL SKILLS

|  |  |
| --- | --- |
| **Programming Languages** | Python, Ruby, C++, Java/J2EE, JavaScript |
| **Web Technologies** | HTML5, CSS3, XML, Bootstrap3, AJAX, Dom, Springboot, jQuery, GoLang, Angular, Next, Vue.js, Bootstrap |
| **Databases** | SQL, Graph QL, MySQL, Oracle DB, MongoDB, Cassandra |
| **Frameworks** | Django, Flask |
| **Tools & IDE** | Visual Studio, Eclipse, PyTest, PyCharm, Xcode, IntelliJ |
| **Python Libraries** | NumPy, Pandas, Matplotlib, SciPy, TensorFlow |
| **Operating Systems** | Windows, Linux, Mac OS |
| **Automation tools** | Jenkins, Selenium, Ansible, Docker, Shell, Kubernetes, Terraform |
| **Methodologies** | Agile, Waterfall |
| **Cloud services** | AWS (S3, EC2, Boto3, Amazon EMR, RDS, CDK, VPC, SG, EKS, IAM, CloudFormation, Sage Maker, Lambda), Azure, Google Cloud Platform |

**PROFESSIONAL EXPERIENCE**

Role: Sr. AWS Python Developer ***November 2022 – Present***

Client: Uline, Pleasant Prairie - WI

***Roles and Responsibilities:***

* Implemented **Django** views and templates to create **dynamic web interfaces**, enhancing user experience, developed **Python** batch processors to automate data handling tasks.
* Developed full-stack web applications using **Python** **(Django, Flask),** for backend logic and **JavaScript, React.js**, **TypeScript** for dynamic user interfaces and integrating with C++ modules.
* Created reusable components in **JavaScript, React.js**, **TypeScript** for component design using **Redux.**
* Automated data processing workflows using Python, C++ for performance-critical components and built Python scripts to interface with C++ tools, developed Python utilities for testing, debugging, and profiling C++ applications.
* Developed serverless applications using **AWS Lambda**, writing Python functions to handle event-driven processes such as API requests, file uploads, and stream processing.
* Developed custom datasets in **PyTorch** to preprocess large-scale structured and unstructured data for model training.
* Automated **Databricks** cluster management and job scheduling using Python scripts and AWS **Lambda** functions.
* Integrated **Databricks** with AWS services like S3, Glue, Athena, and Redshift for seamless data ecosystem interaction.
* Utilized Python to build and optimize Databricks notebooks for interactive data exploration and analysis.
* Managed **ETL** workflows using AWS **Glue** automating ETL processes to streamline data pipelines and integrated AWS Glue with Amazon S3 and Redshift, for efficient data storage for large datasets.
* Processed large datasets using Pandas and NumPy, combining results with C++-based simulations or algorithms.
* Integrated Python with C++ models to visualize computational results using libraries Matplotlib.
* Optimized data processing tasks by leveraging **AWS Glue's** serverless architecture, implemented data cataloging solutions with AWS Glue, enhancing data discovery and governance across multiple data sources.
* Created reusable Python modules to define and manage ML models for **Step Functions** in JSON format.
* Optimized long-running workflows with AWS **Step Function** for error-handling and to ensure reliability of systems.
* Integrated Python-based ML models with **Step Functions** for seamless deployment and real-time workflows.
* Automated build and deployment processes for **Java** applications using tools like **Maven**, **Gradle**, and **Jenkins**.
* Configured **AWS Glue** jobs for incremental data ingestion, enabling real-time data processing and integrated AWS Glue with AWS Lake Formation to enhance data security and access control in a data lake environment.
* Scheduled and orchestrated **AWS Glue** workflows using AWS Step Functions and CloudWatch events for automation.
* Set up and maintain CI/CD pipelines with **AWS CodePipeline, CodeBuild**, and **Jenkins** to streamline release process.
* Automated workflows by writing Python scripts to trigger **AWS** **Lambda functions** based on events in S3, **DynamoDB**, AWS **RDS**, improving operational efficiency.
* Developed data pipelines in AWS **Databricks** using Python for efficient data ingestion, transformation, and analysis.
* Optimized ETL workflows on **Databricks** to process large-scale datasets from AWS S3, RDS, and Redshift.
* Leveraged **PySpark** on Databricks to perform distributed data processing, enabling faster computation on datasets.
* Implemented Delta Lake solutions on **Databricks,** ensuring reliable and scalable data storage with ACID compliance.
* Handled RESTful APIs using **FastAPI**, for better request handling, implemented backend services with FastAPI and integrating with databases PostgreSQL, MySQL, or MongoDB.
* Extended Python apps with C++ modules for time-sensitive operations like image processing, numerical simulations.
* Refactored legacy C++ codebases to support Python bindings, enabling wider accessibility and usage.
* Developed end-to-end Machine Learning pipelines using Python libraries **Scikit-learn, TensorFlow,** and **PyTorch**.
* Implemented data preprocessing and feature engineering steps to optimize **ML model** performance and accuracy.
* Deployed **Machine Learning** models and also expertise in prompt engineering for real-time data gathering and other environment tools like AWS SageMaker, FastAPI, and Flask.
* Integrated and fine-tuned Large Language Models, LLMs on AWS to develop AI-driven Python applications.
* Developed custom NLP solutions using LLM APIs for tasks such as text summarization, and content generation.
* Built pipelines for data preprocessing, inferencing of LLMs using Python libraries such as Transformers and PyTorch.
* Developed microservices architectures using **Java** frameworks like Spring Boot, enabling dynamic scalability.
* Created **Java**-based solutions for complex business logic, integrating APIs, third-party services, and legacy systems.
* Maintained robust RESTful, SOAP web services in **Java** for seamless communication between distributed systems.
* Optimized data pipelines to **ETL** data into **Snowflake**, ensuring efficient data flow and processing for analytics.
* Managed **Snowflake** environments by developing data warehousing solutions, schema design, performance tuning.
* Optimized Glue job performance by tuning worker configurations, **PySpark** code, designed end-to-end data ingestion workflows from **AWS S3** to Amazon Redshift using AWS Glue.
* Built scripts to interact with Snowflake using **Snowflake** Connector to execute queries and manage data operations.
* Deployed infrastructure in AWS CDK for automation, integrated **AWS CDK** with CI/CD enhancing deployments.
* Developed monitoring scripts with **Shell** for system metrics, and for ETL operations for data analysis and visualization.
* Used **Django** for backend **API** development, JavaScript and React libraries to create Single-Page Applications (SPAs).
* Integrated JavaScript, **React** applications with Python microservices to deliver dynamic and interactive web solutions.
* Created **RESTful APIs** with Python, and AWS API Gateway, enabling seamless integration with applications.
* Developed data pipelines for streaming data ingestion into **Snowflake** using Kafka services and AWS Kinesis.
* Automated data ingestion workflows into **Redshift** using Python scripts and AWS services like **Lambda** and **Glue.**
* Enforced data security best practices by implementing RBAC and data encryption within **Snowflake** and AWS services.
* Managed batch processing workflows in **AWS Batch,** Lambda, **DynamoDB** for large-scale data processing tasks.
* Optimized routing algorithms using **PySpark** to minimize transportation costs and delivery times, used Apache **Spark Streaming** to process and analyze real-time shipment tracking data.
* Utilized Python to test and debug Step Functions workflows locally using the **AWS SDK** and Step Functions Local.
* Automated database maintenance tasks like backups, scaling, and monitoring for **AWS RDS** and **Aurora** using Python.
* Deployed ML models, containerized applications on AWS **EKS** for scalable microservices architecture and managed querying relational data on **RDS** and **Aurora**, ensuring efficient data retrieval and updates.
* Implemented Python-based ETL pipelines to extract data from **Aurora,** transform it, and load it into analytics systems.
* Managed datasets using Panda data frames and **MySQL**, queried **MySQL** database queries from **Python** using **Python** -**MySQL** connector **MySQL dB** package to retrieve information.
* Designed Python utilities to optimize **Spring Boot** application configurations and streamline performance tuning.
* Implemented **Spring Boot’s OAuth2** authentication mechanisms for session, token validation while developing scripts.
* Used Python to preprocess data for ingestion into **Spring Boot** applications, enabling real-time data processing.
* Built asynchronous task processing with AWS Lambda, **AWS** **SQS, Dynamo DB** to handle background jobs.
* Utilized **PySpark** and **Dynamo DB** for processing large datasets, optimizing data workflows and processing times.
* Authored validation scripts in **SQL, GraphQL, DynamoDB, PostgreSQL** to ensure data integrity in loading processes.

Environment: Python, Java, PyUnit, Pandas, AWS, HTML, AJAX, CSS, JavaScript, Django, Jenkins, JSON, REST, Git, JIRA, SQL, Graph QL, Mongo DB, Agile, windows and Linux.

Role: Sr. Python Developer  ***December 2020 – October 2022***

Client: Healthesystems, Tampa - FL

***Roles and Responsibilities:***

* Developed monitoring and notification tools using **Python,** generated **Python** **Django** forms to record online data.
* Enhanced **UI/UX** design using **JavaScript, React.js**, **TypeScript** components, **CSS** frameworks Bootstrap, Material-UI and custom styling to ensure consistent and visually appealing user experience.
* Implemented state management solutions in **JavaScript, React** using **Redux**, **TypeScript,** Context API, and custom hooks, improving code maintainability and performance for large-scale applications.
* Developed applications using Python and C++ for deployment in production environments, for database interactions.
* Lead the integration of frontend and backend systems, ensuring smooth communication between **React** components and Python-based APIs via **AJAX, Axios,** and **Web Sockets**.
* Designed Python extensions in C++ to enhance the functionality, improving execution speed for compute-heavy tasks.
* Migrated C++ applications to Python/C++ hybrid systems, reducing development time while maintaining performance.
* Integrated with APIs to pull data to **Databricks**, managed continuous integration of Databricks with automated testing.
* Used **FastAPI** to build data ingestion pipelines, processing large datasets and storing them in AWS S3 buckets.
* Implemented API versioning and ensured backward compatibility for long-term FastAPI projects and wrote unit and integration tests for **FastAPI** endpoints to ensure reliability and robustness.
* Deployed user defined functions within **Snowflake** to enable advanced data transformations directly in the database.
* Integrated **FastAPI** with task queues like RabbitMQ for asynchronous job processing and task scheduling.
* Managed cloud infrastructure using **Infrastructure as Code** tools like **AWS CloudFormation** or **Terraform** for predictable and version-controlled ML deployments.
* Integrated **Databricks CLI** with Python scripts for real-time monitoring, logging for system reliability.
* Developed streaming APIs with **FastAPI**, deployed **FastAPI** applications, ML models to platforms like **Kubernetes.**
* Developed complex **FastAPI** applications for handling custom authentication, logging, and request validation.
* Built APIs using GraphQL with **FastAPI** to provide flexible and efficient querying capabilities for clients.
* Automated API testing for **FastAPI** endpoints using tools Postman and Python libraries **Pytest.**
* Implemented rate limiting and throttling in FastAPI to protect APIs from abuse and ensure fair resource allocation.
* Developed monitoring solutions to track performance for **RDS** and **Aurora** databases, created Python scripts to integrate **AWS Lambda** Functions with cloud-native services such as S3, **DynamoDB**, and SNS.
* Managed Python REST APIs on AWS **Lambda** using **API Gateway**, connected to **Aurora** for backend processing.
* Automated schema migrations and data synchronization for Aurora and RDS databases.
* Automated ML model training, evaluation, and deployment processes using AWS SageMaker and Python SDKs.
* Built NLP solutions using **AWS Comprehend** for entity recognition, and language detection in Python applications.
* Implemented image and video analysis features using Amazon Rekognition APIs and Python integrations.
* Integrated Amazon Polly for text-to-speech conversion, developed data extraction workflows using **AWS Textract.**
* Developed Infrastructure as Code solutions with AWS **CDK**, ensuring consistent deployments across environments.
* Automated data pipeline deployments using **AWS Glue** and Python, ensuring consistent and reliable data workflows.
* Integrated **Step Functions** with API Gateway using Python **Lambda** functions to expose APIs for external access.
* Designed asynchronous workflows in Step Functions to handle large-scale batch processing tasks.
* Integrated monitoring and alerting solutions into **AWS CDK** stacks using CloudWatch Alarms and AWS Config Rules, migrated existing cloud configurations to **AWS CDK**.
* Enhanced **LLM** performance through prompt engineering and testing within Python-driven AI workflows.
* Automated data ingestion pipelines for LLM training datasets ensuring data quality and format compliance.
* Implemented data-driven workflows by using **Step Functions** input-output data manipulation and data transformations.
* Managed containerized Python and React applications using **Kubernetes** for automated deployments and maintained Kubernetes clusters using **AWS EKS** in compliance with HIPAA regulations.
* Integrated **Shell scripts** in Python to handle unused system resources like disk space, memory for optimal performance.
* Collaborated with cross-functional teams to design and implement cloud architectures using **AWS CDK**, aligning infrastructure with business requirements.
* Integrated Python scripts with low-level C++ APIs for sensor interfacing and embedded systems and used C++ for real-time, low-latency tasks while Python provided high-level control and monitoring.
* Worked with **LLMs** to extract insights from unstructured data, integrating capabilities for analytics and reporting.
* Integrated Python applications with LLM APIs like **OpenAI API** and Hugging Face, delivering scalable AI solutions.
* Developed data pipelines using **Apache Spark** on **Amazon EMR** for analyzing large volumes of customers data.
* Optimized **Spark** jobs for scalability to handle medical records, and research datasets while maintaining data privacy.
* Utilized **AWS EKS, Lambda** with **Dynamo DB** for serverless container management, scaling healthcare applications.
* Configured **Kubernetes** pods, handled serverless workflows with **AWS** **Step Functions** for automating data pipelines.
* Developed hybrid solutions combining **Spring Boot** for server-side logic and Python for data science, **AI** capabilities.
* Integrated Spark with **TensorFlow** and **PyTorch** for deep learning models to classify detect issues in health systems.
* Utilized **PySpark** to handle big data processing tasks, improving efficiency and scalability of **ETL** processes.

Environment: Python, PyUnit, Pandas, Shell, HTML, AJAX, CSS, Java, JavaScript, Django, Jenkins, JSON, REST, Git, JIRA, SQL, Agile, Windows, and Linux.

Role: Python Developer  ***May 2019 - December 2020***

Client: Cigna, Bloomfield - CT

***Roles and Responsibilities:***

* Collaborated with a team of developers on **Python** applications focusing on RISK management.
* Developed **Django**-**based** web applications for insurance premium calculations.
* Utilized **Python's** **OS** module in **UNIX** for job cloning, improved **website** **visibility** with **search** **engine** **optimization**.
* Implemented **client**-**side** **validation** using **JavaScript** and created **PyUnit** test cases for **unit** **testing**.
* Integrated RESTful APIs, **GraphQL** endpoints with **AWS API Gateway** to interface with frontend **React** application.
* Ensured communication between client-side **React** applications and backend using **data validation** techniques.
* Built data validation frameworks in Python to ensure data integrity during Redshift ingestion processes.
* Integrated **S3 with Redshift** for efficient data storage and retrieval, leveraging COPY commands to load large datasets.
* Automated **Redshift** cluster management tasks like resizing, backup scheduling, and monitoring with **Python** scripts.
* Enhanced **ETL** performance by leveraging parallel processing and optimizing Python scripts for batch data processing.
* Developed unit tests for Python **ETL scripts** to ensure robust and reliable data workflows in **Redshift.**
* Constructed data tables using **PyQt** to display customer and policy information, facilitating the manipulation of customer records, handled test automation with **Selenium.**
* Configured centralized logging, auditing in ELK Stack to support traceability and HIPAA-compliant logging practices.
* Utilized **Python**-**based** **GUI** components for frontend functionalities such as selection criteria.
* Improved security by implementing secure authentication methods, including **OAuth, JWT**, and **Two-Factor Authentication (2FA)**, ensuring safe user access across **Python** and **React**, **TypeScript** layers.
* Built **ETL pipelines** using Apache Spark and **AWS Glue** to extract, transform, and load data into **AWS RDS**.
* Supported system integration by connecting **Core Java backend** systems with Python-based data analytics pipelines.
* Worked in Python libraries like **Pandas** and **PySpark** for complex data transformations before loading into **Snowflake.**
* Managed containerized Python and React applications using **Kubernetes** on **AWS EKS** for secure deployments.
* Implemented event-driven architecture by integrating Spring Boot services with Python-based event processors.
* Employed the **Unit** **Test** **Python** **library** for testing various **Python** programs and code blocks using **Selenium.**
* Built features like chat systems, live updates, notifications using **WebSockets** in combination with Python and **React**.
* Created microservices with **FastAPI** to improve system modularity and performance, developed frontend applications using **HTML**, **CSS**, **jQuery**, **JSON**, and **JavaScript**.
* Designed DBMS system using **MySQL**, **Graph QL** and engaged in **Agile** Methodologies and **SCRUM** processes.
* Developed a fully automated continuous integration system with **Git**, **Jenkins**, and developed custom tools in **Python**.
* Hands-on with **Selenium** automation, containerized applications using **Docker**, ensuring consistent deployments.

Environment: Python, Django, XML, Shell, Java, Apache, CSS, MySQL, JSON, HTML, JavaScript, Shell Scripts, agile, Restful, UNIX and Windows.

Role: Python Developer ***December 2017 - April 2019***

Client: BBVA Compass Bank, Birmingham - AL

***Roles and Responsibilities:***

* Created a web application using Python scripting for data processing, utilizing **MySQL** for the database, and incorporating **HTML**, **CSS**, **jQuery**, and High Charts for data visualization on served pages.
* Employed a Test-Driven Development (**TDD**) approach to develop the application and implemented unit tests using the Python Unit Test framework.
* Developed views and templates using Python and **Django's** view controller to craft a user-friendly website interface.
* Conducted API testing using the **POSTMAN** tool, examining ETL requests on URLs for response and error handling.
* Debugged and troubleshooted web applications using **Git** as a **version control** tool to collaborate with team members.
* Worked in **MySQL**, **PostgreSQL** and **MongoDB** databases on simple queries and writing stored procedures for normalization and renormalization.
* **Migrated** data pipelines from legacy systems to **Redshift**, leveraging Python-based ETL tools for seamless transitions.
* Employed Apache **Spark** to analyze medical images stored in a distributed file system like **Hive** or **Amazon S3**.
* Created web services API layer in **Django** to supply data, built all **database** **mapping** classes using **Django** **models.**
* Managed data migrations and database schemas using Django’s ORM and manual **SQL scripting** to handle large data.
* Implemented **authentication**, **authorization** mechanisms using tools like **OAuth 2.0, JWT** in web applications to manage security and user access.
* Managed applications on AWS using services **EC2, S3, Lambda**, and **RDS** and implemented serverless architecture using **AWS Lambda**, **API Gateway**, and **DynamoDB** for efficient backend solutions.
* Configured **AWS CloudFront** for content delivery and S3 for static asset storage, deployed microservices using **Docker** and orchestrated with **AWS ECS**, **EKS** for scalable backend systems.
* Worked on **React** advanced features, such as **React hooks**, **context API**, and state management libraries like **Redux**, to build robust client-side applications and services.
* Used **React’s** component-based architecture to develop reusable and modular UI components, accelerating development and improving maintainability.
* Handled **file I/O operations** for applications that required processing and manipulation of CSV, JSON, XML files.
* Configured **AWS Elastic Load Balancer** (ELB) and **Auto Scaling** Groups to ensure seamless handling of high traffic loads and application uptime.
* Implemented **CloudFront** caching strategies and leveraged AWS S3 for serving frontend data and systems.
* Executed MySQL database queries from Python using **Python-MySQL** connector and MySQL database package.
* Branching and merging code lines in the **Subversion** and resolving all the conflicts while merging codes.
* Designed and maintained databases using Python, developing a Python-based **API** (**RESTful Web Service**) with **SQL Alchemy** and **Graph QL**, used Python **scripts** to update the content in **database** and manipulate files.
* Enhanced UI by incorporating navigations, customizable column views using Python-based **GUI** components.

Environment: React, Python, Pandas, AWS, API, Docker, Kubernetes, Postman, Selenium, MySQL, PostgreSQL, SQLite, Python, Django, Git, RESTful web service, POSTMAN, SOAP, HTML, CSS, jQuery, CRUD.

Role: Python Developer ***May 2015 – November 2017***

Client: Menlo Technologies, India

***Roles and Responsibilities:***

* Developed views and templates with **Django** view controller and template language to create a user-friendly website interface and build dynamic web pages and backend services on **Python** frameworks.
* Designed and implemented **RESTful APIs**, supporting data integration and communication between systems.
* Managed code versioning with **GitHub**, Bit Bucket and deployment to staging and production servers and implement MVC architecture in developing the web application with the help of **Django** framework.
* Developed **Python scripts** for task automation, including data extraction, file handling, and process automation.
* Implemented python-based web applications interacting with **MySQL**, implemented **PyTest** for automated testing.
* Developed and optimized **Django** ORM queries, reducing the number of database queries needed by 30%, significantly improving application performance.
* Created **Shell-Python** integrated workflows to monitor system logs and send notifications for unusual activity patterns.
* Designed modular **Shell** scripts to wrap Python scripts, providing end-users with simplified commands and options.
* Collaborated with cross-functional teams to create unified Shell scripts for triggering Python-based CI/CD pipelines.
* Implemented server-side logic and **database** interaction, managing data with **SQL Alchemy** and native Python libraries to build efficient and responsive applications.
* Created data analysis scripts with **Pandas and NumPy**, supporting data processing needs for business operations.
* Worked on **system administration** tasks, creating Python scripts for managing server operations, integrated applications with relational databases, using **SQLite, MySQL**, and **Graph QL** for data storage and retrieval.
* Configured and deployed applications on on-premises servers, using **Apache** and **Nginx** for web server management.

Environment: Django, PyTest, Shell, Jenkins, Cassandra, PySpark, Python, SSIS, SQL, Apache, Nginx, Pandas, NumPy, MySQL, PostgreSQL, Graph QL

EDUCATION

Bachelor of Technology in Computer Science from Cherabuddi Venkata Raghava College of Engineering, India.

**CERTIFICATIONS**

AWS Certified Developer Associate