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 such as **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**.
* Demonstrated leadership in managing projects across the software development lifecycle **(SDLC)**, particularly in the development of monitoring and notification tools using **Python**.
* 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 NoSQL** platforms, ensuring data integrity and performance.
* Deep knowledge of **AWS** services like EC2, S3, RDS, SQS, SNS, Auto Scaling, ELB, and Lambda services.
* Demonstrated proficiency in programming languages such as **Python** and **Java,** 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 with **HTML, CSS, JavaScript, ReactJS**, enhancing UI and interactivity.
* Experienced in using **PyDev, PyCharm** for debugging and worked with **Sublime Text 2, Vim Editors** for coding.
* Skilled in using **Docker** for containerization and **Kubernetes** for orchestration, ensuring scalable, high-performance, and reliable deployments across cloud environments **AWS, Azure, GCP.**
* Proficient in handling microservices using **FastAPI, Kafka,** **Django**, improving API performance, system modularity.
* Experienced with **PySpark** for large-scale data processing and analysis, optimizing data workflows and enhancing the efficiency of **ETL** processes.
* 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, 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, Ansible, Docker, Kubernetes, Terraform |
| **Methodologies** | Agile, Waterfall |
| **Cloud services** | AWS (S3, EC2, Boto3, Amazon EMR, VPC, SG, EKS, IAM, CloudFormation, Sage Maker, Lambda), Azure, Google Cloud Platform (GCP). |

**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.
* Expanded functionality using **Flask** framework, improving the modularity and maintainability of web applications.
* Developed full-stack web applications using **Python** **(Django, Flask),** for backend logic and **React.js**, **TypeScript** for building responsive, dynamic user interfaces and cross-platform functionality.
* Utilized **GitHub** for code versioning, ensuring smooth collaboration and **reliable deployment** to staging and production environments, implemented **PyTest, Selenium** frameworks, ensuring code quality and reliability.
* Created reusable components in **React.js**, **TypeScript** for component design using **Redux** for global state management.
* Developed serverless applications in **AWS Lambda**, wrote Python functions to handle API requests, stream processing.
* Created reusable Python modules to define and manage state machine definitions for **Step Functions** in JSON format.
* Optimized long-running workflows with AWS **Step Function** for error-handling and to ensure reliability of systems.
* Orchestrated Python **Lambda functions** using **Step** Functions for tasks like data transformation and validation.
* Enhanced observability by integrating CloudWatch Logs, creating scripts to analyze Step Functions execution history.
* 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** or other AWS services, improving operational efficiency.
* Optimized data pipelines to **ETL** data into **Snowflake**, ensuring efficient data flow and processing for analytics.
* Managed **Snowflake** environments by developing data warehousing solutions, including schema design, query optimization, and performance tuning.
* Implemented real-time data streaming pipelines using **Apache Kafka** and **confluent-kafka** and **kafka-python**.
* Developed applications to produce and consume **Kafka**, enabling seamless data flow between distributed systems.
* Created **Kafka** consumers to process event streams and store data into databases like PostgreSQL, MongoDB.
* Utilized **Kafka** Streams to transform data in-motion, ensuring timely analytics and event-driven decision-making.
* Integrated **Kafka** with Python microservices for scalable, decoupled communication in event-driven architectures.
* Implemented **Kafka** monitoring using tools like Prometheus and Grafana to track message logs and system health.
* Used **Django** for backend **API** development, **React** libraries for UI to create robust Single-Page Applications (SPAs).
* Designed RESTful APIs using Python frameworks **Flask**, **Django** to communicate with **React**-based user interfaces.
* Enhanced applications using **Flask** and integrate with Redis, used Flask’s architecture to add error logging functionality.
* Optimized database queries in Flask apps using **SQL Alchemy** and connection pooling strategies, integrated third-party payment gateways into Flask applications.
* Integrated and fine-tuned Large Language Models, LLMs on AWS to develop AI-driven Python applications.
* Built pipelines for data preprocessing, inferencing of LLMs using Python libraries such as Transformers and PyTorch.
* Designed data visualization tools in **Flask** using Matplotlib, deployed Flask applications on AWS Elastic Beanstalk.
* Implemented data pipelines using Apache **Airflow** for scheduling and automating workflows in Python applications.
* Integrated **Airflow** with **AWS S3** and configured Airflow’s task monitoring tools for fault tolerance in critical pipelines.
* Created **RESTful APIs** with Python, and AWS API Gateway, enabling seamless integration with applications.
* Worked on creating GitHub action pipelines to deploy code through **Docker** containers into **AWS EC2** instances and frontend static files to **AWS S3** Buckets.
* Developed custom NLP solutions using LLM APIs for tasks such as text summarization, and content generation.
* Created data-driven web interfaces by embedding Python logic into **Mako** templates for real-time content updates.
* Optimized **Mako** templates to ensure proper rendering and compatibility across different environments.
* Used **Mako** with Python backends to generate dynamic PDF reports and email templates for automated workflows.
* Incorporated **Mako** templates into REST API responses for hybrid rendering solutions in Python projects.
* Implemented RESTful APIs using **AWS API Gateway** and integrated Python **Lambda and Step** functions to serve as the backend in ensuring data consistency and accuracy.
* Handled RESTful APIs using **FastAPI**, implemented backend services with FastAPI and integrating with databases PostgreSQL, MySQL, or MongoDB.
* 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.
* 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 in real-time environments using tools like AWS SageMaker, FastAPI, and Flask.
* Collaborated with data engineers to integrate **ML models** into data pipelines using Python and cloud platforms.
* Used Python to develop Explainable AI (XAI) components, ensuring transparency in machine learning predictions.
* Developed Python scripts for managing **Kafka** configurations, including topic creation, partitioning, and replication.
* Automated data ingestion workflows into **Redshift** using Python scripts and AWS services like **Lambda and Glue.**
* Optimized routing algorithms using **Apache** **Spark** to minimize transportation costs and delivery times, used Apache **Spark Streaming** to process and analyze real-time shipment tracking data.
* Deployed containerized applications on AWS **EKS** for scalable microservices architecture and managed querying relational data on **RDS** and **Aurora**, ensuring efficient data retrieval and updates.
* 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 mechanism for session, token validation while developing scripts.
* Used Python to preprocess data for ingestion into **Spring Boot** applications, enabling real-time data processing.
* Worked on creating python utilities in **AWS Lambda** for parsing data, generating reports and other functionalities.
* Built asynchronous task processing with AWS Lambda, **AWS** **SQS, Dynamo DB** to handle background jobs.
* Deployed containerized applications in **Docker** and orchestrate with AWS **ECS, EKS** to ensure scalable microservices.
* 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.
* Actively participated in **Stand**-**up** **meetings**, weekly assessments, daily triage meetings, worked with business analysts to uphold application quality, implemented **Agile** processes and **JIRA** issue management to track **sprint** cycles.

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

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

Client: Healthesystems, Tampa - FL

***Roles and Responsibilities:***

* Responsible for gathering requirements, **system analysis**, **design**, **development**, **testing** and **deployment**.
* Developed monitoring and notification tools using **Python,** generated **Python** **Django** forms to record online data.
* Enhanced **UI/UX** design using **React.js** components, **CSS** frameworks Bootstrap, Material-UI and custom styling to ensure a consistent and visually appealing user experience.
* Developed secure and scalable **RESTful APIs** using Python and hosted them on **GCP’s** **Google Cloud Run** and **App Engine** to manage traffic, security policies, and access controls effectively.
* Lead the integration of frontend and backend systems, ensuring smooth communication between **React** components and Python-based APIs via **AJAX, Axios,** and **Web Sockets**.
* Used Google Identity Platform to implement OAuth2.0 authentication in Flask applications, managed Flask applications integrated with Google BigQuery for advanced analytics and reporting.
* Configured Flask to interact with Google Cloud Storage for dynamic file uploads, enhanced application performance by integrating Flask with Google Cloud CDN, reducing latency for global users.
* Used **Flask** to develop real-time data streaming solutions powered by Google Pub/Sub for event-driven architectures.
* Developed **Mako** templates for consistent UI, integrated Mako templates with Flask to render server-side HTML.
* Optimized template rendering performance by caching frequently accessed components using **Mako’s** built-in features.
* Designed partitioning strategies for **Kafka** topics for efficient load distribution and high availability of applications.
* Configured Kafka brokers, zookeepers, and clusters using Python automation for streamlined deployment and scaling.
* Designed Python ETL processes using **Kafka** as a central message broker for large-scale data ingestion.
* Deployed **Airflow** on **Kubernetes** clusters to scale pipeline executions efficiently using containerized tasks.
* Implemented secure **Airflow** environments by configuring RBAC and encrypted connections for data privacy.
* Worked with **LLMs** to extract insights from unstructured data, integrating capabilities for analytics and reporting.
* Designed context-aware **LLM** workflows for python applications, improving user query resolution and task automation.
* Integrated Python applications with LLM APIs like **OpenAI API** and Hugging Face, delivering scalable AI solutions.
* Built robust **ETL** pipelines using **GCP** services **BigQuery, Cloud Storage,** **Dataflow** for seamless data processing.
* Automated data ingestion workflows from various sources to BigQuery using Python scripts and **GCP** Functions.
* Implemented **Pub/Sub** messaging systems to enable real-time data streaming for Python-based applications.
* Handled Python libraries **Google-Cloud-Storage**, **Google-Cloud-Bigquery in GCP** to interact with microservices.
* Designed and managed **GCP IAM** policies to ensure secure access to Python applications and **GCP** resources.
* Migrated on-premises applications to **GCP**, rewriting workflows in Python to optimize for cloud scalability.
* Used **FastAPI** to build data ingestion pipelines, processing large datasets and storing them in GCP data lakes.
* 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.
* Developed hybrid solutions combining **Spring Boot** for server-side logic and Python for data science, **AI** capabilities.
* Integrated **FastAPI** with task queues like RabbitMQ for asynchronous job processing and task scheduling.
* Built Python-based alerts using **GCP’s** **Cloud Monitoring (Stackdriver)** APIs to track application performance.
* Automated infrastructure in **Terraform** to set up **GCP** resources like **Compute Engine** and **Kubernetes clusters**.
* Integrated **Machine Learning (ML)** models into Python applications using **Vertex AI** and **AI Platform** on **GCP**.
* Implemented data encryption and security best practices in Python applications interacting with **GCP** services.
* Optimized **Spark** jobs for scalability to handle medical records, and research datasets while maintaining data privacy.
* Developed streaming APIs with **FastAPI** for real-time updates, deployed **FastAPI** applications to **Kubernetes.**
* Integrated **RabbitMQ** with Python-based microservices for reliable **message delivery** and task orchestration.
* Deployed event-driven architectures using **RabbitMQ** with Python for decoupled and scalable system design.
* Implemented advanced RabbitMQ features in Python, such as priority queues, message TTL, and dead-letter exchanges.
* Monitored **RabbitMQ** message flow and system health to integrate with management plugins and logging tools.
* Utilized **GCP** with **Databricks** to manage large datasets, optimizing Python to interact seamlessly for best data retrieval.
* Worked in Python libraries like **Pandas** and **PySpark** for complex data transformations before loading into **Snowflake.**
* Developed entire **frontend** with **React JS** and **backend** **modules** using **Python** on the **Django** **Web** **Framework.**
* Created web services API layer in **Django** to supply data, built all **database** **mapping** classes using **Django** **models.**
* Integrated **Spark MLlib** to build predictive models for identifying risks and improving customer outcomes.
* Applied data modeling techniques in **Snowflake** to create efficient snowflake schema designs.
* Employed **PySpark** to handle big data processing tasks, improving efficiency and scalability of **ETL** processes.
* Developed complex web, **middleware**, and **backend** systems in **Python, Java** and **SQL** on a **Linux** server.
* Managed code checking using **Git** and **GitHub,** wrote validation scripts in **SQL, Mongo DB** to validate data loading.

Environment: Python, GCP, PyUnit, Pandas, 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:***

* Developed **Django**-**based** web applications for insurance premium calculations, implemented **client**-**side** **validation** using **JavaScript** and created **PyUnit** test cases for **unit** **testing**.
* Ensured seamless communication between client-side **React** application and backend services using secure protocols and **data validation** techniques.
* Created Python automation scripts for managing **GCP** resources, including provisioning, scaling, and cost monitoring.
* Enabled continuous integration and deployment pipelines (CI/CD) for Python projects on **GCP** using **Cloud Build.**
* Collaborated with data scientists to preprocess and analyze data on **Dataproc** using Python for large-scale computations.
* Implemented robust unit tests within **Databricks notebooks** using **PyTest**, ensuring the reliability of data pipelines.
* Used **Kafka** Connect with Python transformations to facilitate data integration between Kafka and external databases.
* Designed event sourcing patterns using **Kafka** to maintain a reliable record of state changes and user actions.
* Deployed Python-based stream processing applications leveraging **Kafka** and frameworks like Faust or Apache Flink.
* Constructed data tables using **PyQt** to display customer information, facilitating the manipulation of customer records.
* Configured centralized logging using **ELK Stack** to support traceability and HIPAA-compliant logging practices.
* Utilized **GCP’s Google Cloud Scheduler** to automate periodic execution of scripts for data aggregation and reporting.
* Built serverless APIs using **GCP** Cloud Endpoints, enabling seamless integration with third-party services.
* 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.
* Deployed containerized Python applications using **Google Kubernetes Engine** (GKE) for better resource management.
* Integrated Python applications with **Google Sheets API** to dynamically generate and update reports on **GCP**.
* Developed custom logging by exporting log data from GCP Cloud Logging to **BigQuery** for deep analysis.
* Improved security by implementing secure authentication methods, including **OAuth, JWT**, and **Two-Factor Authentication (2FA)**, ensuring safe user access across **Python** and **React** layers.
* Designed event-driven architectures by integrating Python with **GCP** services like Cloud Functions and **Eventarc**.
* Used **Google Cloud Vision API** to develop image recognition applications for automated document processing.
* Orchestrated multi-step workflows using **Google Cloud Workflows** to automate complex business processes.
* Handled application-level code to interact with APIs and web services using **JSON**.
* Worked on Google Kubernetes Engine (GKE) to orchestrate and scale Flask-based microservices.
* Built multi-tenant Flask applications on GCP with RBAC to manage user-specific workflows securely.
* Employed the **Unit** **Test** **Python** **library** for testing various **Python** programs and code blocks.
* Used Google Translate API with Python to localize applications and provide multilingual support.
* Processed streaming data from **IoT** devices by connecting to Python applications running on **GCP** Cloud Dataflow.
* Configured, managed custom **VPCs** for Python-based workloads in GCP to ensure secure and isolated environments.
* Integrated **GCP’s Cloud AI APIs** like **Natural Language, Speech-to-Text** to deliver intelligent features.
* Built GCP Cloud Composer **Apache Airflow** workflows for managing data pipeline orchestration across **GCP** services.
* Handled microservices with **FastAPI** to improve performance, developed frontend using **HTML**, **CSS, JavaScript**.
* Optimized **PySpark** code and Databricks cluster configurations for performance, achieving reduced execution times and cost savings through effective partitioning and caching strategies.
* Designed DBMS system using **MySQL**, **Mongo DB** and engaged in **Agile** Methodologies and **SCRUM** processes.

Environment: Python, Django, GCP, XML, 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)** for system development, handled unit tests in Python Test frameworks.
* 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 various request methods (**GET, POST, PUT, DELETE**) on each URL 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.
* 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 UI components for accelerating development.
* 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** 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 **PostgreSQL**, 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: 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.
* 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 **PostgreSQL** for data storage and retrieval.
* Configured and deployed applications on on-premises servers, using **Apache** or **Nginx** for web server management.

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

EDUCATION

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

**CERTIFICATIONS**

AWS Certified Developer Associate