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** 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.
* 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 with **Vue.js**, developed **Python** batch processors to automate data handling tasks.
* Developed full-stack web applications using **Python** **(Django, Flask)** and Rubyfor backend logic and **JavaScript, React.js**, **TypeScript, Vue.js** for building responsive, dynamic user interfaces.
* Designed RESTful APIs using Flask and **Django** with scalable architecture for microservices-based applications.
* Worked on **Django,** Flask for modular backend development, ensuring efficient separation of concerns and scalability.
* Developed serverless applications using **AWS Lambda**, handled Python functions for event-driven processes such as API requests, file uploads, and stream processing.
* Integrated **Databricks** with AWS services like S3, Glue, Athena, and Redshift for seamless data ecosystem interaction.
* 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.
* Designed Python utilities to manage PySpark permissions and roles, orchestrated PySpark tasks and workflows with AWS Step Functions and optimized Python integrations with PySpark to ensure reliable performance and scalability.
* Utilized PySpark for processing and analyzing large datasets, optimizing data workflows and processing times.
* Integrated Swagger, **OpenAPI** for auto-documentation of APIs to enhance developer onboarding and usage.
* Created reusable Python modules and enhanced UI with **Vue.js** to manage ML models for **Step Functions** in JSON.
* Integrated Python-based ML models with **Step Functions** for seamless deployment and real-time workflows.
* 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.
* Automated workflows trigger **Lambda functions** based on **S3**, **DynamoDB**, AWS **RDS**, maintained Ruby on Rails codebase by adding features, fixing bugs, and optimizing existing code.
* Built task scheduling and asynchronous processing features in **Flask** applications using Celery and RabbitMQ.
* Secured Flask APIs by implementing input validation, handled **Vue.js** microservices, SQL injection prevention.
* Handled Flask-Mail for email notifications and automated messaging in web applications and integrated Flask applications with NoSQL databases like **MongoDB** using **PyMongo** for handling unstructured data.
* 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.
* Handled RESTful APIs using **FastAPI**, for better request handling, implemented backend services with FastAPI and integrating with databases PostgreSQL, MySQL, or MongoDB.
* Deployed **Machine Learning** models and expertise in prompt engineering for real-time data gathering and other environment tools like AWS SageMaker, FastAPI, and Flask.
* Developed microservices architectures using **Java** Spring Boot, and enhancing UI with React.js and Vue.js.
* 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.
* Managed **Snowflake** environments by developing data warehousing solutions, including schema design, query optimization, and 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.
* Used **Django** for backend **API** development, JavaScript, Vue.js, React libraries to create Single-Page Applications.
* Integrated **React, Vue.js** 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.
* Designed modular back-end services using **Django** REST framework, adhering to microservices architecture principles.
* Created highly available and scalable API gateways using Flask and deployed on **AWS Elastic Beanstalk**.
* Automated data ingestion workflows into **Redshift** using Python, Ruby scripts and AWS services **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.
* Enhanced database performance for large datasets by optimizing queries in PostgreSQL, MySQL, and MongoDB.
* Developed Python, Ruby scripts to integrate and manage **SQLAlchemy** ORM for seamless database interactions.
* 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.
* Managed datasets using Panda data frames and **MySQL**, queried **MySQL** database queries from **Python** using **Python** -**MySQL** connector **MySQL dB** package to retrieve information.
* Implemented efficient SQL queries, indexing strategies in MySQL to reduce data retrieval for high-traffic applications.
* Automated MySQL backups, disaster recovery, and integrated Django MySQL in SQLAlchemy for DBMS operations.
* Conducted data migrations and version control using Django’s migrations framework for MySQL databases.
* 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.
* 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.
* Handled validation scripts in **SQL, GraphQL, DynamoDB, PostgreSQL** to ensure data integrity in loading processes.

Environment: Python, Java, PyUnit, AWS, S3, EC2, VPC, Glue, EKS, 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**.
* Managed cloud infrastructure using **Infrastructure as Code** tools like **Terraform** for version-controlled deployments.
* Built robust **ETL** pipelines uisng **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.
* Configured **Cloud SQL** databases for Python applications requiring scalable backend storage.
* Built Python-based alerts using **GCP’s** **Cloud Monitoring (Stackdriver)** APIs to track application performance.
* Containerized Python applications using **Docker** and deployed them to **Kubernetes** for scalable microservices.
* 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.
* Utilized **GCP** with **Databricks** to manage large datasets, optimizing Python to interact seamlessly for best data retrieval.
* Handled scripts in **PyUnit**, **Python** test frameworks, developed **business** **logic** in **Python, Java** to implement planning.
* Worked in Python libraries like **Pandas** and **PySpark** for complex data transformations before loading into **Snowflake.**
* Managed **ConfigMaps** and **Secrets** to securely store clinical data without exposing sensitive information.
* Developed entire **frontend** with **React JS** and **backend** **modules** using **Python** on the **Django** **Web** **Framework.**
* Developed custom **Helm charts** for consistent deployment of complex healthcare applications, configured **Horizontal Pod Autoscalers (HPA)** to handle high-traffic periods, such as telehealth sessions or patient record retrieval.
* Handled Python scripts to automate data workflows in **PySpark**, ensuring metadata management and compliance.
* Integrated **Pyspark** with external systems andREST APIs, enabling data exchange and synchronization.
* Automated the creation and management of data pipelines and workflows using PySpark, Python scripts.
* Employed Apache **Spark** with big data tools to analyze medical images stored in a distributed file system like **HDFS**.
* Created web services API layer in **Django** to supply data, built all **database** **mapping** classes using **Django** **models.**
* Designed **RESTful APIs** using **FastAPI** to streamline data integration, handled every phase in **SDLC.**
* Integrated Spark with **TensorFlow** and **PyTorch** for deep learning models to classify detect issues in health systems.
* 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.
* 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.
* Built custom connectors using Python to integrate **PySpark** with databases, data lakes, and analytics platforms.
* Developed data quality validation pipelines with Python to enforce governance policies and update PySpark assets.
* Enhanced **PySpark** ingestion processes with Python scripts for extracting and loading metadata from sources.
* 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**.
* 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.
* Built real-time features such as chat systems, live data updates, using **WebSockets** with Python and **React**.
* 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.
* Developed a fully automated continuous integration system with **Git**, **Jenkins**, and developed custom tools in **Python**.
* Used **Python** **IDE’s** such as **PyCharm**, **sublime** **text** for developing the code and performing unit test using python modules like PyTest and unit test.
* Participated in the entire **project** **lifecycle** including Design, Development, Deployment, Testing, Implementation.

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