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**.
* 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**.
* Demonstrated leadership in managing projects across the software development lifecycle **(SDLC)**, particularly in the development of monitoring and notification tools using **Python**.
* Proficient in handling microservices using **FastAPI** and **Django**, improving API performance and system modularity.
* Experienced with **PySpark**, **Collibra ADI** 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, Collibra ADI |
| **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 |

**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 the overall user experience, developed **Python** batch processors to automate data handling tasks.
* Expanded functionality using **Flask framework**, improving the modularity and maintainability of web applications.
* Generated business unit-specific **datasets** using **Python** scripts from **database** **tables**, created **DB** **views** and **stored** **procedures** to retrieve **datasets** based on timestamps.
* Developed full-stack web applications using Python **(Django/Flask)** for backend logic and **React.js** for building responsive, dynamic user interfaces.
* Utilized **GitHub** for code versioning, ensuring smooth collaboration and **reliable deployment** to staging and production environments, implemented **PyTest** for **automated testing**.
* Design serverless architecture using **AWS Lambda** to process inventory updates in real-time, used **AWS DynamoDB** and **Amazon RDS** to store inventory data with high availability.
* Implement a notification system using **Amazon SQS** to alert warehouses when stock reaches critical levels.
* Created reusable components in **React.js** for component design using tools like **Redux** for global state management.
* Developed serverless applications using **AWS Lambda**, writing Python functions to handle event-driven processes such as API requests, file uploads, and stream processing.
* Integrated APIs like **payment gateways, logins** and Python **Pandas** for large-scale **data** **analysis**.
* 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, or other AWS services, improving operational efficiency.
* Designed Python scripts for bulk asset creation, updates, deletion with **Collibra ADI**, optimizing governance processes.
* Implemented robust ETL pipelines to preprocess data for ingestion into Collibra using the **ADI** framework.
* Created validation scripts in Python to verify the integrity and compliance of data ingested via **Collibra** **ADI.**
* Created **RESTful APIs** with Python, AWS service 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.
* Implemented RESTful APIs using **AWS API Gateway** and integrated Python **Lambda** functions to serve as the backend for web and mobile applications.
* Develop a backend using AWS **API Gateway** to expose shipment tracking endpoints, stored and analyzed shipment data using AWS **Redshift** for real-time analytics.
* Developed Python scripts to automate data governance workflows in **Collibra,** for efficient metadata management.
* Integrated **Collibra ADI** with external systems in Python and **Collibra REST APIs**, enabling seamless data exchange.
* Built custom **connectors** using Python to integrate **Collibra ADI** with databases, data lakes, and analytics platforms.
* Developed **data quality** validation pipelines with Python to enforce governance policies and update in **Collibra** ADIs.
* Enforced data security best practices by implementing RBAC and data encryption within **Snowflake** and AWS services.
* Managed batch processing workflows in **AWS Batch,** Lambda, or **Step** Functions for large-scale data processing tasks.
* Handled for **AWS** services like maintaining and troubleshooting **EC2** instances, **S3** **buckets**, **VPC, Load Balancers.**
* 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.
* Worked on creating python utilities in **AWS Lambda** for parsing data, generating reports and other functionalities.
* Developed and maintained microservices using **FastAPI** to enhance **API** performance and reduce latency.
* Deployed and managed containerized applications using **Docker** and orchestrate them with AWS services like Amazon **ECS** and **EKS** to ensure scalable microservices architecture.
* Utilized **PySpark** for processing and analyzing large datasets, optimizing data workflows and processing times.
* Authored validation scripts in **SQL, MongoDB** to ensure data integrity during 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, Collibra, 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 **Azure API Management** to manage traffic, security policies, and access controls effectively.
* 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.
* 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.
* Integrated robust data security protocols using **Azure Key Vault** and **Azure Databricks Secrets** to protect sensitive information within Python applications.
* Designed Python systems to monitor ADI workflows, logging errors, providing real-time updates for governance teams.
* Integrated **Collibra ADI** with enterprise systems (e.g., Snowflake, Redshift, and Tableau) using Python for seamless metadata exchange.
* Customized **ADI** connectors using Python to meet unique organizational requirements for metadata ingestion.
* Developed Python-based testing frameworks to validate **Collibra ADI** configurations, workflows during deployments.
* Enhanced **Collibra's** metadata ingestion processes with Python scripts for ETL operations from various data sources.
* Automated reporting processes in **Collibra** using Python to generate insights on data governance KPIs.
* Leveraged Python to manage and update **Collibra** asset templates, enabling consistent data governance all throughout.
* Optimized Python-based **integrations** with Collibra to ensure reliable performance and scalability.
* Designed Python systems to manage **Collibra** permissions and roles programmatically, ensuring secure access control.
* Containerized Python applications using **Docker** and deployed them to **Azure Kubernetes Service (AKS)** for scalable microservices architecture and improved application management.
* Integrated Python with external APIs to pull data into **Databricks**, using **Azure Functions** for handling data.
* Integrated Python scripts in **Azure Logic Apps** to create complex workflows and automate data processing for services.
* Worked on **Azure Blob Storage** and **Azure File Storage** for large-scale storage solutions, implementing Python scripts for efficient data storage and retrieval operations.
* Implemented event-driven architectures with **Azure Event Grid** to enable seamless communication between services.
* Optimized **Spark** jobs for scalability to handle medical records, and research datasets while maintaining data privacy.
* Utilized **Azure Data Lake Storage** with **Databricks** to manage large datasets, optimizing Python code to interact seamlessly with ADLS for efficient 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.
* Employed Apache **Spark** with big data tools to analyze medical images stored in a distributed file system like **HDFS**.
* Managed code versioning and CI/CD pipelines using **Azure DevOps**, ensuring continuous integration of Databricks projects with automated testing and validation.
* Orchestrated **Collibra** tasks and workflows using Python with scheduling tools **Apache Airflow, AWS Step** Functions.
* Conducted **API testing** for Collibra using Python libraries such as Postman and Requests, ensuring robust integrations.
* Deployed Python-based functions on **Azure Functions** for executing serverless computing tasks that integrated with **Databricks** workflows for extended capabilities.
* 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 **Azure Monitor**, **Databricks CLI** with Python scripts for real-time monitoring, logging for system reliability.
* 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, 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.
* 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 **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** layers.
* Created Python-based automation scripts to manage and scale **Azure** resources dynamically, using **Azure Resource Manager** (ARM) templates and **Azure SDKs**.
* Created data ingestion and processing workflows using Python scripts within **Azure Data Factory** and **Azure Functions** for serverless data orchestration.
* 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.
* Configured Python applications for **high availability** using **Azure Load Balancer** and **Availability Sets**, ensuring minimal downtime and fault tolerance.
* Orchestrated complex data ingestion in **ETL** pipelines using **Azure Data Factory** to automate data movement.
* Configured applications with **Azure Load Balancer**, **Azure Traffic Manager** to manage high-availability and traffic.
* Implemented disaster recovery plans by scripting automated backups using **Azure Backup** and **Azure Site Recovery**.
* Managed datasets using Panda data frames and **MySQL**, queried **MySQL** database queries from **Python** using **Python** -**MySQL** connector **MySQL dB** package to retrieve information.
* Managed applications in **Azure Kubernetes Services**, ensuring better orchestration for containerized workloads.
* 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**.
* Built Python scripts in **Azure Data Lake Analytics** and combined with **U-SQL** for data processing and analysis.
* Optimized **PySpark** code and Databricks cluster configurations for performance, achieving reduced execution times and cost savings through effective partitioning and caching strategies.
* Configured and managed Python-based **Databricks jobs** to automate the execution of ETL tasks and machine learning workflows, integrating with **Azure Data Factory** for orchestration.
* 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, 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