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 of AWS.
* 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).
* 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), 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.
* Involved in complete **SDLC** process and developed monitoring and notification tools using **Python**.
* 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**, reducing bug detection time by 25%.
* 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.
* 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.
* 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.
* 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 operations and maintenance support for **AWS cloud resources** which include launching, maintaining, and troubleshooting **EC2** instances, **S3** **buckets**, Virtual Private Clouds (**VPC**), Elastic Load Balancers (**ELB**) and **RDS.**
* 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.
* Built asynchronous task processing with AWS Lambda and **AWS** **SQS** to handle background jobs.
* Deployed and managed containerized applications using **Docker** and orchestrate them with AWS services like Amazon **ECS** and **EKS** to ensure scalable microservices architecture.
* Implement real-time data synchronization between different healthcare systems (EHR, PMIS, payer systems) to ensure accurate, up-to-date information is available for billing and claims processing.
* 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, 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**.
* Participated in the complete Software Development Life Cycle (**SDLC**).
* 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 (e.g., Bootstrap, Material-UI), and custom styling to ensure a consistent and visually appealing user experience.
* Implemented cloud-based solutions using AWS services to build efficient systems for healthcare RCM applications.
* Implemented state management solutions in **React** using **Redux**, Context API, and custom hooks, improving code maintainability and performance for large-scale applications.
* Maintain Python-based applications to automate key processes within the **RCM workflow**, improving billing, claims management, and payment posting for healthcare providers.
* Lead the integration of frontend and backend systems, ensuring smooth communication between **React** components and Python-based APIs via **AJAX, Axios,** and **Web Sockets**.
* Implement **ETL** pipelines to integrate data from multiple healthcare systems (**EHR, EMR**), ensuring seamless data flow for billing and claims management processes.
* Created business unit-specific datasets using **Python** **scripts** from **database** **tables.**
* Implement AWS security best practices to ensure that sensitive data is handled in compliance with HIPAA regulations.
* Created and integrate RESTful APIs to connect third-party RCM software with internal systems.
* Created testing scripts using **PyUnit**, the **Python** **unit** **test** **framework**, for all Python applications.
* Worked on Python libraries like **Pandas, 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.**
* Integrated **Lambda** with AWS CloudWatch for real-time logging, monitoring, and alerting, proactively addressing performance issues and errors in serverless applications.
* Develop data pipelines and analytical tools to track Key Performance ndicators (KPIs) for revenue cycle health, helping healthcare organizations optimize revenue, reduce denials, and improve collections.
* Implemented event-driven architecture by leveraging **AWS Lambda** to process events from AWS services like S3 (file uploads), Kinesis (streaming data), and DynamoDB Streams.
* Built and maintained **Lambda-backed microservices** for tasks such as image processing, data aggregation, and notifications, reducing operational complexity and infrastructure costs.
* Implement logging and monitoring solutions using **AWS CloudWatch**, and other observability tools to ensure application health and proactive incident management.
* Configured AWS and Snowflake for cross-region data replication to improve data availability and redundancy.
* Created web services API layer in **Django** to supply data, built all **database** **mapping** classes using **Django** **models.**
* Designed and built **RESTful APIs** using **FastAPI** to streamline data access and integration.
* 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 **AJAX**-**driven** applications by invoking web services/API and **parsing** the **JSON** response.
* 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, Amazon Web Services (AWS), 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** environments for job cloning, and improved **website** **visibility** through **search** **engine** **optimization**.
* Implemented **client**-**side** **validation** using **JavaScript** and created **PyUnit** test cases for **unit** **testing**.
* Constructed data tables using **PyQt** to display customer and policy information, facilitating the manipulation of customer records.
* 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.
* 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.
* Managed datasets using Panda data frames and **MySQL**, queried **MySQL** database queries from **Python** using **Python** -**MySQL** connector **MySQL dB** package to retrieve information.
* Built real-time features such as chat systems, live data updates, and notifications using **WebSockets** in combination with Python and **React**.
* Reduced infrastructure costs by adopting serverless architecture with **AWS Lambda**.
* Migrated traditional **monolithic** backend services to serverless architecture using **AWS Lambda**, improving scalability, reducing operational overhead, and enhancing application reliability.
* Created microservices with **FastAPI** to improve system modularity and performance.
* Developed frontend applications using **HTML**, **CSS**, **jQuery**, **JSON**, and **JavaScript**.
* Tested **RESTful** **APIs** including **Twilio**, **Facebook**, and **Twitter** **integrations**.
* Designed DBMS system using **MySQL**, **Mongo DB** and engaged in **Agile** Methodologies and **SCRUM** processes.
* Developed a fully automated continuous integration system using **Git**, **MySQL**, **Jenkins**, and custom tools developed 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.
* Containerized applications using **Docker**, ensuring consistent and efficient deployment.

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**) 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 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.
* 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 on **Python** frameworks.
* Generated Python Django forms to record data of online users and used **PyTest** for writing test cases. Increased the speed of pre-existing search indexes through Django **ORM** optimizations.
* 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.
* 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.
* Monitored AWS **EC2** container instances via Datadog API, enhancing cache data, assigned permissions, policies, and roles through **AWS Identity and Access Management (IAM).**
* Implemented **AWS Elastic Container Service (ECS)** scheduler for streamlined Docker-based cloud deployment.
* Created an Amazon **VPC** with public-facing and private-facing subnets for servers and backend systems.

Environment: Django, ECS, EC2, Dynamo DB, S3, Jenkins, Cassandra, PySpark, Python, SSIS, SQL.

EDUCATION

Bachelor of Technology in Computer Science.

**CERTIFICATIONS**

AWS Certified Developer Associate