

# Pritesh T.

## Sr. Python Developer

I have a wealth of experience in managing software development teams and completing high-quality projects as a Team Lead with over 7 years of experience. I can clearly explain complicated technical concepts to stakeholders who are both technical and non-technical thanks to my great technical talents and leadership qualities. Using a variety of technologies and programming languages, including but not limited to Python. I have substantial expertise in designing, developing, and implementing enterprise-level software solutions. I am knowledgeable about DevOps procedures, cloud computing platforms like AWS and Azure, and Agile approaches.

## PROFESSIONAL SUMMARY

---

- I have accumulated over 7 years of experience in all stages of the life cycle, including Design, Development, Implementation, and testing of server-client, Mobile, and web-based applications.
- My expertise includes application development, software development, and design using Python, Django, React.js, Algorithms and Data Structures, Software Design in HTML and DHTML, and databases like Oracle PL/SQL, and Postgresql.
- I am particularly skilled in the usage of Django and Flask, high-level Python Web frameworks for developing Web applications, and have experience in building frameworks and automating complex workflows using Python for Test Automation.
- I have experience with design, code, debug operations, reporting, data analysis, and web applications utilizing Python.
- I have extensive exposure to LAMP (Linux, Apache, MySQL, and Python) and WAMP architectures and excellent experience using various Python-integrated IDEs such as Sublime Text, SPyder, PyStudio, PyCharm, PyDev, and NetBeans.
- Furthermore, I am proficient in working with web services using protocols SOAP and REST, as well as various Python packages such as NumPy, SQLAlchemy, Pyside, SciPy, and PyTables.
- I am experienced in Software Development Life Cycle, data engineering, database designs, agile methodologies, coding, and testing of enterprise applications using IDEs such as Jupiter Notebook, PyCharm, Spyder, and Visual Studio.

## TECHNICAL SKILLS

---

**Programming Languages:** Python 2.x / 3.x, JavaScript, Bootstrap, HTML5, SQL, HTML, CSS3, XML

**Frameworks:** Django, Flask, Fast API, OAuth 2, JSON Web Tokens (JWT), Swagger

**Libraries/APIs:** Django REST API, Stripe, Selenium, Scrapy, Beautiful Soup, Tenants, Channels, Numpy, Scipy, Pyjamas, Stripe API, Facebook API, GitHub API, Twitter API, Google Places API, Google Play API

**Tools:** Firebase Cloud Messaging (FCM), Google Analytics, Pycharm, Shell, Jenkins, Jira, Visual Studio App Center, NetBeans, Tableau, JMP Pro, Travis CI, Spyder, Pycharm, Pyscripter, Pystudio, Atom, Git, AWS Lambda.

**Paradigms:** Automation, REST, Clean Architecture, Scalable

- I have expertise in full life cycle application development and good experience in Unit testing and test-driven development (TDD) and Behavior-driven Development.
- My operating system experience includes Windows, Linux, UNIX, and Mac, and I have strong development experience on Linux-based operating systems such as Ubuntu, Red Hat, and CentOS.
- I have interfaced with the product and other executives to determine strategy and drive technical direction.
- My technical skills include strong defect analysis, debugging, and problem-solving skills with an excellent understanding of problem-analysis methodologies, techniques, and tools.
- I have hands-on experience with an in-depth level of understanding in the strategy and practical implementation of AWS cloud-specific technologies including Elastic Compute Cloud (EC2), Simple Storage Services (S3), Route 53, Cloud Formation, Elastic IPs, Virtual Private Cloud (VPC), RDS, and Cloud Watch.
- I am also experienced in automating, configuring, and deploying instances on Cloud environments and On-Premise Data centers, as well as in the areas of DevOps, CI/CD Pipeline, Build, and release management, AWS/Azure, and Linux/Windows Administration.
- I have experience in Bash and Python Scripting with a focus on DevOps tools, CI/CD, and AWS Cloud Architecture and hands-on Engineering.
- I am experienced with controlling access to files and directories with Linux file system permissions and have extensively used Veritas Cluster Server 4.x/5.x for High Availability Environment.
- With a proven track record of leading successful software projects, I am able to guide and mentor team members to achieve their goals.
- I am committed to delivering high-quality products on time and within budget while maintaining a customer-centric focus.

Application, Agile, Functional Programming, Microservices, Continuous Delivery (CD), Continuous Integration (CI), Event-driven Architecture, Test-driven Development (TDD), Web Architecture, Serverless Architecture, Continuous Deployment, API Documentation, Machine Learning, Deep Learning, Neural Networks, Data Analysis, Data Scientist, Information Retrieval, Robot Framework, Image Processing

**Platforms:** AWS Lambda, Firebase, Google Cloud Platform (GCP), Twilio, Docker, Amazon Web Services (AWS), Unix, Heroku

**Storage:** JSON, PostgreSQL, Databases, NoSQL, Redis, MySQL, Cloud Firestore, Google Cloud, MongoDB

**Industry Expertise:** Fintech, Banking & Finance, Web Development, Education, Real Estate, Automobile, E-Commerce Digital

**Methodologies:** Agile, Scrum, TDD, BDD, Waterfall

# PROJECTS

---

## Coin Switch Kuber

The project mainly focused on writing high-quality code by maintaining scalability and extensibility in mind. The main functionalities of the product are to buy/sell crypto assets, maintain a wallet and provide lean user experience while maintaining high throughput capabilities from backend APIs. The main responsibilities involved integrating different payment gateways for the deposits to reduce downtime. Apart from that, solving the day-to-day bugs that arise in the app and working with the notifications flow for different use cases. Creating banners and developing the cryptophytes API, optimizing the coin listing API which shows the list of coins.

The cryptophytes API mainly gives upcoming rewards and features for a user while maintaining a content length very short. It is like the InShorts app which provides a summary of articles in a structured manner. Majorly this API was integrated to boost user interaction and spread knowledge among users about crypto investments.

### Key Challenges Faced

Adding the payment gateways along with existing gateways without affecting the system, The payments distribution are mostly capped due to payment gateway-related restriction, unavailability and/or policy changes so there are several measures that have to be implemented.

Maintaining the throughput and the concurrency aspects.

There was a whole cycle of code quality and deployment tools that had to be passed before the production push. It was not just about technical understanding as sometimes the legal guidelines, finance terms, and conditions have to be understood before working on major components.

**Technologies:** Python, Flask, Postgresql, Kibana, Grafana, Twilio, Textlocal, AWS EC2, Bitbucket, AWS Codepipeline, Jira, Gitlab, Redash, New Relic, Retool, Appzotic, Elasticsearch, Logstash

## Scissors & Scotch

The project mainly focused on booking appointments online for a particular barber shop and checking for the availability of a particular barber. Customers can book an appointment in the available slots and can pay online for the services they want. Creating the staging and pre-production environment for enabling a proper testing flow and optimizing the collection queries involved in the APIs.

### Key Challenges Faced

Adding and maintaining the staging and the pre-production servers.

Maintaining the accuracy of logging and alerting tools in order to provide a negligible delay in alerts.

**Technology:** Python, Flask, MongoDB, ElasticSearch, Logstash, Kibana, Logtail, AWS Codepipeline, AWS

## PRASAR: Video Conferencing Server

- PRASAR (Video conferencing Server) is a web application-based visual communication session between two or more users regardless of their location, featuring audio and video content transmission in real-time.
- It saves your time and money on business vacations, cuts event management costs, etc.
- Scheduling of meetings/conferences in different periods - daily, weekly, monthly, and yearly.
- Notifying meeting guests/users with email and i-calendar(ICS files)
- Manage ongoing portal to handle ongoing conferences for different features

**Technology:** AJAX, CSS, HTML, RestAPI, Javascript, JQuery, Python, Django, XML, Scheduling, ICS, MariaDB, SQLite3

## Teach Guide - Teaching Guide: Digital Teaching Platform

- The main requirement was to gather the data of online teaching platforms, what courses they offer, how many and which authors are there on each e-learning platform, and also the metadata for each course like the categories, course details, co-authors, schedule, etc.
- We did this by storing the sitemap of all these sites in the cloud in a customized JSON format and then triggering an SQL stored procedure that compares the already existing data with the new files to update the existing courses if there are any changes or add new data.
- Creating a common architecture that generalizes scrapping all sitemaps and gathers common data from different sites like Udemy, Udacity, Coursera, etc.
- Upload large content of scraped data to Azure cloud and dump the data in the warehouse in Azure.
- SQL procedures to check and update only non-existing data in the warehouse

**Technology:** Python | Scrapy | Microsoft Azure Cloud

## Physician Search & Review System: Healthcare

- This web application was developed to allow patients to search (for doctors and rate and add reviews for them).
- The system allows Physicians to generate unique PMS for their patients to add reviews for them.
- The patients can use the pins assigned to them and provide their feedback for specific physicians by visiting the site and clicking on their profiles.
- Profiles for physicians were auto-created on the site using an automated cron job.
- Search addresses of physicians/clinics in desired areas.
- Patients can review their physicians 0 Patients can upload images

**Technology:** Django 2.1.1, Python 2.7, GIT, JQuery, Google API, GeoDjango, PostgreSQL, Radius/Location Search

## MakeMyDocs: Education Website

- MakeMyDocs is an educational website platform where the primary goal is to create a custom document based on user requirements with different payment modes.
- The platform provides users with a simple UI interface where users can simply drag and drop templates to create documents quickly and easily.
- You don't have to be an IT guru to use this tool—no technical knowledge or special required.
- The application provides easy creation of thousands of secure documents within minutes via CSV document with payment modes for users or organizations.
- The application platform provides other features like - adding/modifying/deleting the uploaded templates, and custom templates to generate multi/bulk documents with the same template.
- No minimums plan/No hidden fees, You pay for what you create and then download and use that document over and over again, etc.
- The application provides a client-server system with a robust and optimized backend with AWS lambda.

**Technology:** Django, Python, Rest API, JavaScript, jQuery, Angular, HTML, Payment Gateway - Stripe API (SEPA & Credit Card), AWS S3

## Tradcrew

To develop a platform that converts natural language to SQL queries, the client wants a platform where we can add databases as data sources, and for retrieving the data instead of using complex queries in SQL we can simply ask questions in plain English and the platform will provide all answers needed. The problem is widely popular in the Natural Language Processing domain as a 'Text-to-Code' conversion We Created the Hybrid approach to tackle the situation where we are using statistical as well as deep learning to solve it

### Key Points:

- Natural language processing to solve the Text-to-code problem.
- Deep learning-based as well as statistical models aka Hybrid approach
- Google Bigquery for warehousing the data sources
- APIs for easy Interaction with the platform
- Google 1 AM authentication in GCP

**Technology:** Google Cloud Platform, Deep learning, Python 3.6, Ln2SQL