Headline:	**Junaid Soomro | Full stack backend heavy engineer** **Python - Nodejs - Reactjs**
Mobile:	080-7841-1087
Personal Email:	Js.darkld@gmail.com
LinkedIn:	https://www.linkedin.com/in/junaid-soomro-063a6a179/
Portfolio:	https://portfolio-414ad.web.app
GitHub:	https://www.github.com/junaid-soomro/ (Most repositories on my GitHub are private - confidential)

Language Skills: English – Fluent Level
Japanese – Basic

Education: Sep 2014 - Dec 2018
Iqra University, Pakistan
Bachelors in Computer Science(Software Engineering)

Qualifications:

- Academic IELTS 7.0

Technical Skills:

- Flask
- Typescript
- AWS
- Openstack
- Python Django
- Kubernetes
- MYSQL and MongoDB
- NodeJs with Express
- Azure Devops
- Bash scripting
- Azure service bus integrations.
- Dynamics365
- Google suite API integration (gmail, gchat, gdrive)
- Docker
- Ubuntu
- Java
- Reactjs/Nextjs
- HTML, CSS

Professional Summary

Senior Software Engineer with **7** years of experience in IT, specializing in building **performant** and **scalable** server side applications using **NodeJs** and **Python**. From designing apps and development to deployment on **Azure/AWS** virtual machines powered by **CI/CD** and **Kubernetes** is what i am hands on with.

Professional Experience

July 2024 – Present Bebit Inc/ Japan

Product Name: Usergram Position: Backend Engineer

Usergram helps its consumers to analyze user behavior for better UX and achieve their revenue or sales goals. The product also utilizes AI to get a much enhanced user behavior analysis based on the data provided.

Technologies: AWS Athena, AWS Cassandra, AWS EKS, Python, Flask, Linux, LLM integration

- **User intermediate journey:** A service which pulls on user activity history from the database and passes it on to AI to understand why users didn't reach the end goal for e.g. checkout or booked an event etc. The service was built using **Python** and **Flask**.

- **Usergram contribution:** Contributed to some small features in usergram product which involved **AWS Athena queries**, **AWS glue jobs**, PR reviews etc.

- **Terraform AWS:** All of the AWS resources are managed via **Terraform** repository. Recently I created **lambda functions** and managed **user permissions**.

Mar 2022 – June 2024 Cognni/Shieldox, USA

Product Name: Cognni - Data classification and Labeling
Position: Backend Engineer

The product manages somewhere between 20 to 30 customers averaging with 500 users in each organization. The product classifies and labels activities of users. The project defined below are standalone services that were supporting the main product and it was a full solo effort.
Technologies: Typescript, AWS, NodeJs, Python, Flask, Azure CI/CD, Service bus and Kubernetes, Linux, MySQL, Mongo, Redis

- **Project AI Model Training:** The goal was to create a file classification model by training it with various **datasets** and **hyperparameters** to achieve maximum accuracy. The model used for training was **BERT-small**.

- **Project Tokenizer:** The responsibility of this project was to classify personal and critical information from information/content that is shared by the users that exist in the customer's organization. Personal information can be SSN, credit card etc. This service receives activities of all the users of an organization and will label them if it found any personal information in the content.

- **Project Gsuite:** Our product was limited to Microsoft only that's when we brought in Google. The project integrates and scans for **GDrive**, **GChat** and **Gmail** activities in the customer's organization, and similarly to Microsoft, we started classifying and labeling information from customers who own Google enterprise.

- **Project Risky Users:** As the title itself suggests. Once the product classifies and labels the information. This information is then further analyzed by this project to see and mark any risky users. Risky users are those who do forbidden actions for e.g sharing a file to their personal email or an external organization etc. Emails were also sent out to these users informing them about their activity with a refraining notice

My roles and responsibilities:

- API design and development of backend applications built using **NodeJs** and **Python**.
- **MongoDB** integration with the service and database query optimization to reduce time, save network costs and test for scalability.
- Partially implemented **SOLID** principles when creating server applications which hosted several endpoints.
- Built standalone automated server side health monitoring scripts using **Flask** that had API's for returning their current status.
- Devops work includes; **Azure CI/CD** for a **Nodejs** service, configuring service busses interaction in **Azure** with the backend service and service deployments in **Azure Kubernetes** environment.
- Feature knowledge transfer meetings, project demonstrations, and code reviews.

Nov 2019 –Mar 2022 iVolve Technologies, Pakistan

Product Name: Cloud7/QCloud Position: Full Stack Engineer

Cloud7 is a cloud orchestration portal backed up by Openstack API's just like AWS. Just like any cloud portal it had cloud resources management screens as well as billing. We didn't had many customers back then because this was a work in progress. We normally had 10 to 50 users.

Technologies: HTML, CSS, Sass, Openstack/GCP, Typescript, Reactjs, Nodejs, Python Django, Docker, Kubernetes, NTOP, Killbill billing, MongoDB

- **MFA Feature:** Integrated **OpenStack** MFA functionality in our product with TOTP.

- **Egress Traffic Counter Research:** I researched on how we can track **egress/outgoing** traffic from the virtual machines that were hosted on **Openstack**. Came up with a solution called **NTOP.** I then started integrating **NTOP** with our product to track and bill customers outgoing traffic from vms.
- **PROJECT MS-BILLING:** This project handled invoicing, subscriptions, usage reporting, pricing and SKU management of our cloud portal. Partial implementation of payment gateways were also made.

My roles and responsibilities:

- Created backend applications using **ExpressJS** which were talking to **SQL** and **MongoDB** databases.
- Followed **MVC** design pattern to create API's on **NodeJS**.
- Integrated **Openstack** API's with **Python Django** backend server.
- Code reviews.
- Feature demonstration to customers.
- Automation scripting.
- Deploying services to production servers running **Kubernetes** on **AWS**.