**Prawin Thangaswamy**
prawin609@gmail.com
+1-548-333-1184

**LinkedIn:** https://www.linkedin.com/in/prawin-thangaswamy-75545468/
**GitHub**: https://github.com/Prawin609

## Professional Summary

Agile Full Stack Software Developer with more than 3 years of strong experience in design, development, testing and deployment of web-based and Enterprise Client-Server business applications using OOP, Java/J2EE, Angular and Kubernetes.

## Technical Skills

| | | |
|---|---|---|
| Programming Languages | : | Java, JavaScript |
| Java/J2EE Technologies | : | Spring Boot, Spring Data JPA, Spring Security Basics |
| Web Technologies. | : | HTML5, CSS3, JavaScript, Angular 11 |
| Operating Systems | : | Windows, Ubuntu |
| Databases | : | MySQL, MongoDB |
| Web/App Servers | : | WebLogic, Apache Tomcat |
| Web Services | : | RESTful Webservice |
| Version Control Tools | : | GitHub |
| Build Tools | : | Maven 3.0, Jenkins |
| IDE | : | Eclipse, Visual Studio Code |

## EDUCATION

**Graduate Certificate** in **Web Design and Development** **Jan '20 – Apr '21**
Conestoga College, Kitchener, Ontario

**Bachelor of Technology in Electronics and Instrumentation Engineering** **Aug '12 – Jun '16**
Sri Ramakrishna Engineering College, Coimbatore, India **GPA – 3.6/4.0**

## Professional Experience

**Employer: Infosys Limited**
**Client: Comcast**
**Role: Senior Systems Engineer**

**Project (vCMTS NGAN):** **Dec '16- Nov '19**

**Description:**

The project is development, implementation changes, deployment and testing of vCMTS – Virtual Cable Modem Termination System, A Continuous Integration/Continuous Deployment enabled application deployed in Kubernetes environment.

**Responsibilities:**

- Developed application on **Micro Service Architecture** using **Spring Boot** to integrate and communicate with the newly created and already existing applications.
- It is containerized as a **Docker** image built by Jenkins and tested under **Kubernetes Platform**.
- Performed **REST API** calls between the services for the manipulation and transferring the data between modules.
- Designed solution for alerts for proactive monitoring of the application to reduce the downtime for end customers.
- Developed the **Unit Test cases** for the module using **Mockito Framework**, Analysed the code quality and Unit Test coverage using **SonarQube**.
- Used **RabbitMQ** library for channel creation, exchanging and publishing the message based on queue to the other modules and checked the Kubernetes logs in **Kibana Dashboard**.
- Have the good amount of working experience in **Amazon Web Service**.
- Implemented some of the functionalities in Java 8 - **Lambda functions** and **Streams API**
- Rich experience in developing the respective module's frontend in **Angular**. Have good exposure to Angular routing, authentication using JWT, authorization using route guard, Angular forms validation.
- Periodically committed the code in **GitHub** and triggered **Jenkins** build to get the docker image in nexus repository.
- Familiar in getting the logs and monitor the heath of the deployed image using kubectl commands of Kubernetes along with ubuntu commands. Sometimes used Kafka for messaging service in other modules.

**Technical Project**

**Project (To-do Management Application):**

**Description:**

This Project is development of **Spring Boot** application as a server with **Angular 11** as a Front-End application.

- This application manages the users to track the daily tasks and update it based on the completion level.
- Developed **REST API** to expose the data to the front end from the server for a valid user.
- Used **Maven** as a build tool for this project.
- Implemented one of the security concepts, **JWT authentication** to allow the user to get the data securely.
- Implemented Login and Logout functionality, hided the few features if the user not logged in.
- Developed Single Page Application in Angular to manipulate the CRUD operations, **two-way data binding** in forms, **angular forms** data validation.
- Used **Spring Data JPA** framework for the CRUD operation using **MySQL** with **H2 database**.
- The page design in based on HTML 5, CSS 3 and Bootstrap for the interactive UI.
- Containerized the Server Application as Docker image and pushed it to the repository of Docker Hub.
- Deployment of the image to **Google Cloud Platform** with Kubernetes is currently in development.

**Environment:** Java 7.0, Spring Boot, Kubernetes, Docker, Rest Template, Maven, Web Services, GIT.

**Project (Party Store Application):**

**Description:**

This Project is developed in the **MEAN** (MongoDB, Express JS, Angular and Node JS) stack technology with more features for the ecommerce.

- Created collections for inventory details, users along with the subdocuments with **MongoDB Cloud**.
- Implemented the APIs for getting the data from the MongoDB document collection using the **mongoose library**.
- Routed the APIs from the controllers to the specific routing links for the client rendering process.
- Deployed the server-side application in Heroku along with pipeline integration of GitHub.
- Developed the client application in **Angular 11** with more functionalities.
- Designed the signup page for the user-registration to login the application with **JWT authentication**.
- Hided the details of other users if a specific user is logged-in using **ng-directives**.
- Implemented **RxJS switchMap** for viewing the specific item details from the inventory list.
- Filtered the data based on the specific category in the inventory list.
- Implemented CRUD operation in the reviews section of the specific product.
- Used Route Guard for the accessing the URL from the unauthorized user.

**Environment:** Node JS, Angular 11, MongoDB, Express JS, Rest Template, GIT, Heroku.

**Project (Spring Cloud - Microservices):**

**Description:**

This project is developed in the **Spring Boot, Spring Cloud, Spring Data JPA, Docker** and **Kubernetes**.

- Created more than 6 service application to talk to each other to attain the use case for microservice architecture.
- Created each service with the Spring Clod dependencies like **starter-config, starter-sleuth, sleuth-zipkin, spring-rabbit**.
- Registered every service in the **Eureka Naming Server** and attained load balancing using **Spring Cloud LoadBalancer**.
- Traced each server requests using **Zipkin UI, RabbitMQ and Sleuth**.
- Made all these microservices as a **docker image** and deployment to **Kubernetes** in **Google Cloud Platform** is currently in progress.