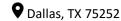# HARSH BHAVSAR

(469) 407-0782   harshbhavsar21   harshbhavsar2110@gmail.com   Dallas, TX 75252

SUMMARY:

Software Engineer with 1+ years of experience specializing in middleware for micro web services, working with technologies like Java 8/9, Spring Boot, gRPC, Cognitive Science in Agile Scrum environment with a graduate degree in Intelligent Systems.

EDUCATION:

- Master of Science in Computer Science, The University of Texas at Dallas, Richardson, Texas **May 2017**
- Bachelor of Engineering in Information Technology, Gujarat Technological University, India **May 2014**

  Relevant Courses: Artificial Intelligence, Big Data Analytics, Machine Learning, Natural Language Processing, Design and Analysis of Algorithms, Advanced Computer Network, Object Oriented Analysis & Design

TECHNICAL SKILLS:

Programming Languages	:	Java, C, C++, Python
Web Technologies	:	AngularJS, React.js, Redux, HTML5, CSS3
Frameworks	:	Spring Boot, gRPC, OAuth2, Hadoop, Spark, Kafka, Pig, Redis, RabbitMQ
Cloud Services + Databases	:	AWS, Google Cloud, OpenStack, Couchbase, Oracle DB, MySQL
Hardware Technologies	:	Wireless Sensor Network, IoT, Embedded Systems
Tools and Applications	:	Sumo Logic, SOLR, JIRA, Jenkins, GIT, Google Analytics

PROFESSIONAL EXPERIENCE:

Software Engineer at Copart Inc., Dallas, TX **May 2017 – Present**

Back Office Billing & Payments – Copart Back Office portal for accounting.
- Part of Back Office project Architecture Design while working on Bank Reconciliation and Document Service modules.
- Coordinated with business, finance team, SAP specialists and other team members for architectural design.

Software Engineer Intern at Copart Inc., Dallas, TX **January 2016 – May 2017**

Image & Document Service – Micro web services for image and document processing.
- Involved in development of efficient micro web services using cloud and messaging services. Added error handling and retry mechanism to upload documents automatically & manually. Secured it using OAuth2 authentication.
- It processes more than 1 million images and documents per day and the failure rate is lies between 0.025% to 0.05%.

Intelligent reader for image scanning – A desktop application to scan and upload documents.
- Improved document scanning process using bar code to scan and read data and spring boot inspired multi-threaded architecture to achieve highly efficient asynchronous behavior.
- Implementation of this structure resulted in 60% faster performance and 25% fewer errors.

Other Projects: Document Scanning and Imaging, Copart-Auth (OAuth2)

Software Developer at Orbit IT Solutions, India **June 2014 – November 2014**
- Involved in building **Quality Management System** project while designing, maintaining and developing the interaction of RESTful web services in Spring MVC with desktop application in JavaFX + Eclipse RCP in CentOS environment.

Software Engineer Intern at Indian Railways (CRIS), India **June 2013 – July 2013**
- Worked on **Freight Operation Information System's** middleware component using Oracle Tuxedo & Web logic, while maintaining and tracking the live status of freight trains.

PATENT & PUBLICATIONS:

- Published **Patent** named as "Multifunctional Gateway Sensor Node" with Patent Number **3327/MUM/2013**. **October 2013**
- Published **IEEE International Paper** on "Multifunctional Gateway Sensor Node" with **ISBN: 978-1-4799-3826-1**. **April 2014**
- Published **AICTE National Paper** on "The Assessment of Micro Weather Station" with **ISSN: 2349-9079**. **January 2014**

ACADEMIC PROJECTS:

Political Actor Recommendation System **January 2017 – May 2017**
- Developed a Big Data application which identifies a political actor from live news streaming and feeds, tweets and recommends actor's role using Spark Streaming, Elastic Search, NLP and Machine Learning algorithms.

English Grammar to ASL Grammar Algorithm **May 2016 – December 2016**
- Developed an algorithm using Natural Language Processing and Machine Learning for converting English grammar into American Sign Language grammar. It also corrects English grammar before parsing it to the ASL.