

---

**EXPERIENCE**

---

**• Expertrec***Software Engineer*

Bangalore

*Jun 2019 - Dec 2019*

- Implemented cross browser snapshot testing (taking screenshots of a web-page on different OS and browsers then comparing them against old ones) using Protractor. Workflow was : using vagrant open an OS, then open one of the browser, inside it open testing web-page, take screenshots of each page, save them in the local server, run a pipeline in Jenkins, where we compare the production website screenshots with the testing ones and if all are same merge.
- Redesigned and Implemented part of frontend e-commerce dashboard and added a new analytics page.

*Typescript, Angular(5, 8), Docker, Jenkins, Vagrant*

---

**PROJECTS**

---

**• Bolowiki.com**

Created a website where users can search (either via voice or by typing) in multiple languages for some Wikipedia article and then convert that article to speech. Authenticated users get an additional feature of converting translated text between some languages, to speech. Website is hosted on a virtual ubuntu server using Unicorn as a python server and NGINX used as a reverse proxy, protected by ufw firewall.

*Python, Javascript, MySQL, Flask, Bootstrap, BS4, Unicorn, NGINX*[Related code](#) and [blogpost](#)**• Key-Value service**

Implemented strongly consistent fault-tolerant key value service by using Raft consensus algorithm under MIT 6.824 course. Operations supported by service are get(key), append(key, arg) and put(key, value), it maintains a simple database of key/value pairs where key and value are strings. Client talks to the service through a Clerk then clerks based on the type of request pass the request to key-value server whose associated Raft is leader. Service re-sends failed requests caused by network and server failures to raft after a timeout also detects any duplicate requests. Communication between clerk, key-value server, raft is done via RPC, threads are used to manage concurrent tasks and timeouts are used to detect failed requests.

*GO, RPC, Distributed systems*[Related code](#) and [blogpost](#)**• Raft Consensus Algorithm**

Implemented Raft, a replicated state machine consensus algorithm that gives strongly consistent fault tolerant behaviour to the service using it. Implemented leader election, log replication and state persistence to deal with network and server failures. Optimised the basic algorithm by implementing accelerated log backtracking for bringing stale followers up to date quickly.

*GO, RPC, Distributed systems*[Related code](#) and [blogpost](#)**• General Purpose Computer**

Built a general purpose computer under a course named [NandToTetris](#), on the hardware side built logic gates, registers, RAM, ROM, CPU. memory mapped I/O using hardware description language (HDL). On the software side implemented assembler, virtual machine and compiler for a simple language called Hack in python. Hack supports loops, conditionals, methods, constructors and classes. Then using Hack implemented a mini-OS which handles input from the keyboard, shows output on screen, has memory, math library, string and array types.

*Python*[Related code](#) and [blogpost](#)**• Map Reduce**

Implemented MapReduce library based on the paper with same name under MIT 6.824 course. Workers ask for a job, Master assigns them a Reduce job if all the Map jobs are finished along with the data file and map/reduce function. Communication between Master and Workers is done via RPC, threads/go routines are used to handle concurrent tasks.

*GO, RPC, Distributed systems*[Related code](#) and [blogpost](#)

---

**PROGRAMMING SKILLS**

---

**• Languages :** Python, GO, Javascript, Typescript, SQL, Rust, Dart, C**• Frameworks :** Flask, Angular, Bootstrap, Flutter**Others :** BS4, Nginx, Unicorn, Docker, Jenkins**• My setup :** i3gaps, polybar, vim, zsh/fish, alacritty/terminator[My config](#)

---

**EDUCATION**

---

**• Jaipur National University***Bachelor of Science**June 2014 – Dec. 2017*