ABHISHEK MADIRAJU

San Mateo, CA| (415)-910-6010 | abhishekmadiraju@gmail.com |www.github.com/AbhishekMadiraju

EDUCATION

New Jersey Institute of Technology, Newark — Sep 2016 – Dec 2017
Master of Science in Computer Science — **GPA: 3.56/4.0**
Courses: Internet and Higher Layer Protocols, Security and Privacy in Computer Systems, Database Management Systems, Cognitive Computing, Operating Systems, Current Topics in Data Science, Cloud Computing, Java Programming

Texas State University, San Marcos — Jan 2016 – May 2016
Master of Science in Computer Science — **GPA: 3.5/4.0**
Courses: Analysis and Design of Algorithms, Formal Languages, Programming C++

Visvesvaraya Technological University, Belgaum, India — Aug 2011 – Jun 2015
Bachelor of Engineering in Computer Science — **GPA: 3.0/4.0**

TECHNICAL KNOWLEDGE

Programming Languages:	Java, Python, SQL, C, C++
Professional Software:	Eclipse, Visual Studio Code, Code Blocks, Wireshark
Web Programming:	HTML, CSS, JavaScript, Node.js, MongoDB, Express
Tools:	ffmpeg, lex, yacc, gdb, Microsoft Windows, UNIX, Linux, Hadoop

PROJECTS

Yelpcamp – Campground Review Web Application — Nov 2017 – Dec 2017
- Developed a web application which enables users to view and review(comment) campgrounds. The users can add their own campgrounds with its image after they login.
- Used Node.js, Express and MongoDB on the backend, and HTML, CSS, Bootstrap, and JavaScript on the frontend.

Online Membership System — Nov 2017 – Dec 2017
- Developed an Online Membership System using JAVA, which provides access to user and admin based on a username and password with different functionalities for each of them. User can modify their own data and search for other users whereas the admin can add, view, remove and modify any of the users.
- Used JDBC to provide database functionality and Java Swings for the Client user interface. Made use of Sockets and Threads for the backend server.

Big Data Analysis on AWS — Sep 2017 – Oct 2017
- Developed a Java MapReduce application that takes an input of Wikipedia pages dedicated to US states, to compute occurrences of certain keywords in each page, outputs the number of states for which each of these words is dominant and identifies those that have the same ranking for these words.
- Tested the program on a 4-node cluster built using AWS EC2 instances running a custom Hadoop AMI stored in a S3 bucket.

Stock Options Advisor — Feb 2017 – Apr 2017
- Built a stock options trading advisor, which helps identify the critical factors which maximize profit.
- Implemented a Backus-Naur form grammar using yacc/bison to parse HTML data from 300 HTML files obtained from NASDAQ to extract the option chain. Used this data to determine the call and put value for an appropriate strike price.
- Implemented the A-priori algorithm on the above strike prices to verify the assumed factors affecting the profit.

Packet Radio Streamer — Nov 2016 – Nov 2016
- Developed a UDP client using C with Berkeley sockets API in a Linux environment, which fragments a given audio file in any format, and sends it to a remote network player like VLC player at regular intervals based on the bitrate.
- Experimented with ffmpeg to understand the various factors such as streaming rate, codec, container, protocol, bit rate and file type, which affect the quality of the stream.

Proxy Server — Sep 2016 – Nov 2016
- Developed a proxy server using C with Berkeley sockets API in a Linux environment. The server handles HTTP GET requests and can successfully download files from FTP servers.
- Obtained a deep understanding of the TCP/IP client server model.

Sentimental Analysis of Twitter Posts — Jan 2015 – May 2015
- Performed text analysis on twitter posts using Naïve Bayes Classifier and Maximum Entropy algorithms. The program deduces the sentiment of a tweet as positive or negative and displays the location of the tweeter on Google Maps.
- Implemented synonym mapping, which maps a word in a tweet to its synonym in the feature list, and abbreviation mapping, which maps abbreviations to their expanded meaning in the feature list. This made the learning more dynamic and helped achieve a significant increase in accuracy compared to conventional methods.