Prashanth Rao

Email : psprao95@gmail.com
Mobile : +1-682-256-1683

Education

- **University of Texas at Dallas** — Richardson TX
  Masters, Computer Science; GPA: 3.81/4.0 — *Jan 2018 – Dec. 2019*
- **VIT University** — Vellore, India
  Bachelor of Technology, Electronics and Communication Engineering; GPA: 3.5/4.0 — *Jul 2013 – Jul 2017*

Technical Skills

- **Programming Languages:** : C++, SQL, Java, Python, C
- **Web Technologies**: HTML, CSS, JavaScript, PHP, jQuery, AJAX, XML, JSON, Web Services (REST and SOAP), Mean Stack
- **Databases**: MySQL, MAMP, MongoDB
- **Tools**: Eclipse, IntelliJ, Clion, Git, Atom, Slack, Latex

Coursework

Database Design, Advanced Design and Analysis of Algorithms, Web Programming Languages, Machine Learning, Operating Systems, Discrete Structures, Data Structures and Algorithms, Statistical Methods for Data Science*, Big Data Management and Analytics*, Computer Vision*
*Currently ongoing in Spring 2019

Projects

- **Contact-Manager**: A Java based GUI application that manages a list of contacts by interacting with an SQL database. Contact details may include name, multiple addresses, phone numbers and birthday. Functionalities include: 1. Searching for a contact based on any field, Adding a new contact, Modifying an existing contact, Deleting an existing contact,
- **Davis-Base**: A simple database engine that is based on the file-per-table variation of the SQLite file format. Supported functions: Create Database, Drop Database, Show Databases, Create Table, Drop Table, Show Tables, Select Star, Select-From-Where, Insert, Update, Delete
- **Client-Server-Datastore**: Client-server system that communicates over TCP-IP sockets. The functions of the server included storing, retrieving, deleting and listing named data sets. The server was responsible for storing this data in a persistent fashion for later retrieval. There were two modes of testing: 1. Against a datastore server class implemented by the team 2. Against the grader's datastore which was running on an AWS server
- **Task Executor Library**: Service that accepts instances of tasks and executes each task in one of the multiple threads maintained by a thread pool. The goals were: 1. To implement multithreaded synchronization when tasks are being executed by multiple threads. 2. To implement a FIFO queue for the tasks added by the user which is both thread-safe and blocking This project was implemented in Eclipse.
- **Easy-Movers**: A website for a moving company. Features include user signup, login, cart (remove item, checkout clear cart), order history , search filter (text and category) for user and admin privileges for product updating product information, removing product, or adding a new product.
- **Wine Dataset NN**: Implemented a neural network for classifying the wine data set. The wine data set contained examples each of which had 13 features. The project was implemented in python with the help of Tensorflow. Two neural networks were developed: 1. One for the original dataset, which used overfitting techniques 2. A smaller portion of the original dataset, which did not use overfitting techniques

Certifications

- 'Neural Networks and Deep Learning' and 'Neural Networks: HyperParameter Tuning, Optimization and Regularization' (Courses 1 and 2) of the Deep learning Specialization by *deeplearning.ai* on Coursera
- Python Fundamentals and Python Data Structures (Courses 1 and 2) of the Python Specialization by *University of Michigan* on Coursera

Links

- Github: www.github.com/psprao95 view here
- Linkedin : www.linkedin.com/in/psprao view here