ADITYA DEVENDRA PANDEY

1615 S Cooper St, Arlington, TX 76010
(313)-775-9172
adi.pandeycs@gmail.com

http://adipandey.com/
https://www.linkedin.com/in/adi-pandey
https://github.com/addherbs

Academic Background

Master of Science - Computer Science - GPA: 3.42 / 4.0 — August 2016 to May 2018
University of Texas, Arlington

Bachelor of Engineering - Computer Engineering - GPA: 7.08 / 10.0 — August 2012 to May 2016
University of Mumbai, Xavier Institute of Engineering - Mumbai, India

Computing Skills

Programming Languages: Python, C, Java, JavaScript, PHP, C++, Assembly, Arduino
Statistical Language Tools: R Programming, MATLAB
Web and Application Frameworks: Flask, Node.js, Android Studio, Hadoop, ReactJS
Software: Microsoft Office, Microsoft Windows, Linux
Database: MongoDB, SQL**,** Minibase, AWS RDS, Oracle
Cloud Computing Services: EC2, S3, RDS, EB, ElastiCache, IBM Bluemix
Basic Knowledge: Firebase, Ruby

Professional Experience

Cloudstrats Technologies, Mumbai, India — June 2016 to August 2016

- Developed an Android application "XFix Services", a product of Cloudstrats Technologies which provides the interface between the customers and the services that the company provides.
- Developed a website for the company product "XFix Services".
- Was part of the database Migration project from Gmail to Office365.
- Developed an understanding on Oracle database and further analyzed large and complex datasets to complete projects provided by the company.
- Worked on two projects of developing websites for the clients.

Projects

TTL-Privacy (Time-to-Live) using Node.js (JavaScript) Link — Sep 2017 to Dec 2017

- Developed an encrypted messaging web app which maintains the integrity and confidentiality of the message.
- Messages aren't deleted after read immediately but have a TTL when composed. The message will be deleted even if it is not read after the allotted time.

Sort Texas Temperature using Hadoop Map/Reduce (Java) Link — Nov 2016 to Dec 2016

- Used the Hadoop Map/Reduce Paradigm on a dataset which consists of weather data gathered on an hourly basis for Texas State from year 2005 to 2011.
- Implemented a custom sort algorithm (with Partitioner and Sampler) to sort the temperature data and implemented a sampling technique to determine the non-uniform temperature ranges and sorted data accordingly.

Concurrency Control Transaction Manager using C++ Link — Oct 2016 to Nov 2016

- Implemented a transaction Manager that manages concurrency control using strict 2-phase locking protocol with shared locks for read operations and exclusive locks for write operations.
- Also, implemented a pool of mutexes, semaphores and condition variables to ensure the operations belonging to the same transactions are executed in proper order.

B+ Tree Implementation using Java Link — Sep 2016 to Oct 2016

- Developed a complete B+ Tree data structure in the Minibase database system.
- Java Programming was used as a tool to hardcode and synchronize the Minibase Libraries into the project.

Server Controlled Multi-Client Chat Messenger using Swing (Java) Link — March 2017 to May 2017

- Made a desktop client that can chat with multiple clients privately and can also broadcast messages.
- The server is the intermediate source. It can handle clients by commands like, 'clients', 'kick', 'raw', 'quit'.

Lyrics Finder using Flask (Python) Link — May 2017 to June 2017

- Implemented a general Lyrics Finder server, in which we can query an English as well as Hindi, Marathi songs.
- It also shows the related songs for a query.

Final Year Project: Mobile Anti-Theft System (Android Application) — Aug 2015 to April 2016

- An Android client application project based on GPS satellite tracking system.
- It had the following key features - Switch to General mode from Silent mode, Raising an Alarm, receiving notification on SIM card change, Content Deletion, Activation of MIC, Getting GPS location.

Give Weather Forecast using Flask (Python) Link — Feb 2017 to March 2017

- It takes input as latitude and longitude of a region and generates the temperature, wind speed, direction, cloud amount, precipitation and water states of that region.
- Flask was used for backend and beautiful soup was used for parsing the data from the SOAP request for a website.