in.linkedin.com/in/nakulgulati
github.com/nakulgulati

Nakul Gulati

nakulgulati.com

(+91) 978 2477 088
nakgulati@gmail.com

Experience

Software Engineer II **Amazon** **April 2017 - Present**

- Working on backend systems that power product pages on Amazon across the globe

Software Engineer **Baxi: The Bike Taxi** **August 2016 - March 2017**

- Currently working on the backend server (NodeJs + Express); used by the user facing apps and internal tools
- Implemented a logging mechanism to generate functional level logs. The logger intercepts function calls and logs its arguments removing the need to log manually
- Extended the backend system to accommodate iOS specific use cases
- Built the iOS hybrid (cordova) application from ground up

Research Assistant **LiveLabs@SIS, SMU, Singapore** **June - August 2015**

- Proposed and prototyped a system to detect occupancy and hogging of an area in real-time
- The IR sensor prototype achieved 80% occupancy detection accuracy
- Work published in the International Workshop on Internet of Things towards Applications, ACM Sensys 2015

Student Developer **Google Summer of Code 2015** **May - August 2015**

Organization: Pidgin, Finch and libpurple

- Reversed engineered and implemented Google+ Hangouts protocol for libpurple 3.0
- It enables users to use Pidgin (a popular open source IM client) for Google+ Hangouts
- Protocol to be merged in libpurple3 (currently in development)

Software Development Intern **Pyoopil Educational Technologies** **June - July 2014**

- Implemented 2 out of 5 REST modules for the Pyoopil Dynamic Learning Environment backend API engine; backend engine implemented in CakePHP and MySQL
- Setup a continuous integration pipeline to automate the code deploy process with Amazon Web Services

Publications

"Real-time Detection of Seat Occupancy & Hogging" by Nguyen Huy Hoang Huy, Nakul Gulati, Lee Youngki and Rajesh Krishna Balan, **International Workshop on Internet of Things towards Applications, ACM Sensys 2015**

Projects [C, Python, Java, PHP, SQL, JavaScript, GNU/Linux]

- **Sentiment Analysis** [Machine Learning, Natural Language Processing, scikit-learn, Python]
  - Predicting polarity (positive/negative) of textual data; the data is modelled into bigram probabilities
  - The data is then classified using a Naive Bayesian classifier and Stochastic Gradient Descent
- **Stock Recommender** [Python, Association Rule Mining, Content Based Recommendation]
  - The key idea for recommendation was relevance then rank. Association Rule Mining was used to filter and Content Based recommendation to further refine and rank.
  - Improvements can be made by using deep learning techniques and factoring in Sentiment Analysis.
- **Graph Engine** [Java, PHP, MySQL]
  - Engine to store graph like data in relational schema; provides API to generate first-level entity association graphs
- **Connect Four AI Bot** [Java]
  - Implemented Minimax algorithm in Java to play the game of Connect Four
  - The algorithm determines optimum move by 'looking' four moves ahead