

Philip Vo

San Jose, CA | 408.386.6142 | philiptranbavo@gmail.com
[linkedin.com/in/philipvo](https://www.linkedin.com/in/philipvo) | github.com/PhilipVo | philipvo.github.io

SUMMARY

Full stack web developer in MEAN, Python, and iOS with a strong focus on back end development. Background in computer science and engineering with knowledge of both hardware and software. Looking to join a team of enthusiastic engineers and developers that challenge each other to produce great products.

SKILLS & TECHNOLOGIES

Languages	Front end	Back end	Databases	OS	Software	Hardware	Misc.
HTML5	Angular.js	Flask	MySQL	Windows	Git/GitHub	Stellaris	Bash
CSS3	React.js	Node.js	-Workbench	macOS	Eclipse	Arduino	Breadboard
JavaScript	jQuery	Express.js	-MAMP	Linux/Unix	Xcode		Security
C/C++	AJAX	Socket.IO		Ubuntu	VMware	Altera	VANET
Python	Bootstrap	JWT	MongoDB	Red Hat	OMNeT++	-SoCKit	REST
Swift	Skeleton	AWS EC2	-Mongoose	Fedora	µVision	-DE2	CRUD
Java		Nginx		Kali	MATLAB	-Arria II	MVC
SQL					OpenCL/GL	-Nios II	
Verilog					CUDA		TCP/IP
ARM					Quartus	Xilinx	-SSH
x86					Vivado	-Artix-7	-FTP/SFTP
Tcl					LaTeX	-ZedBoard	-HTTP/S
					PuTTY	-MicroBlaze	-Bluetooth

EDUCATION

Coding Dojo 2016
Resident Engineer Triple Black Belt (highest earning of achievement)

University of California, Davis 2013 - 2015
Master of Science, Electrical & Computer Engineering Researched VANETs and ITS
In progress Units completed: 64

University of California, Davis 2009 - 2013
Bachelor of Science, Computer Engineering GPA: 3.54

FULL STACK DEVELOPMENT

froxxi - froxxi.com Ronin Dev, 2016

- Social shopping site integrating ShopStyle API to allow users to share links to where they buy clothes.
- **Roles:** Delegated tasks amongst team of four members. Setup backend server, database, and front end technologies for full MEAN implementation. Secured site using JWT for authentication. Integrated ShopStyle API. Managed GitHub repository, merging conflicts and refactoring code.
- **Technologies:** MEAN (MongoDB, Angular), JSON Web Token, JQuery, Skeleton, ShopStyle API.

Stardate - github.com/PhilipVo/Meteorites Coding Dojo, 2016

- Future meteorites are tracked as events, allowing users to discuss and create meetup locations for stargazing.
- **Roles:** Solely wrote an entire web application using full MEAN stack. Integrated NASA's API to gather meteorite data and Facebook's login API to allow users to login to their dashboard. Set up MongoDB to store user information and events. Deployed on AWS servers.

- Technologies: MEAN (MongoDB, Angular), JQuery, Bootstrap, NASA API, Facebook login API.

Localator - github.com/PhilipVo/Localator-1

Coding Dojo, 2016

- Use iPhone's location to actively find distance from your friends for meetup purposes, such as at concerts.
- Roles: Directed and guided team of three members to implement and integrate various technologies into an intuitive iPhone app. Integrated MapKit and Core Location frameworks to map location of connected devices. Designed UI that responsively sends visual, audio, and tactile alerts to users based on distance.
- Technologies: Swift, OS X, Xcode 7.3.1, Core Location, MapKit, Socket.IO, AVFoundation, JavaScript.

Snippet - github.com/PhilipVo/iOS-Projects/tree/master/snippet

Coding Dojo, 2016

- Text editor on iPhone/iPad for users to write, save, and send code to each other through the use of a chat room.
- Roles: Worked in a team of three members. Integrated an intuitive keyboard through CocoaPods for improved typing. Coded a text editor that autocompletes brackets, braces, and parentheses for writing faster code. Worked on autoindent feature for text editor. Helped combine all features into a functional application.
- Technologies: Swift, JavaScript, OS X, Xcode 7.3.1, Socket.IO, CocoaPods.

ENGINEERING PROJECTS

VENTOS - [Vehicular NeTwork Open Simulator](#)

UC Davis, 2014 - 2015

- Simulator for studying Intelligent Transportation Systems (ITS) where vehicles utilize wireless communications.
- Roles: In a team of five students, individually researched ITS technologies and traffic controllers to implement various intelligent traffic signal controllers with state machines. Coded new module to include bicyclists in the simulator. Contributed in writing papers for submission in multiple conferences.
- Technologies: Ubuntu, C++, Python, OMNeT++, Veins, SUMO, MATLAB.

Robotic Systems - [Project report](#)

UC Davis, 2014

- Robotic system consisting of a moving robot to find and collect trash and a fixed base station to sort the trash.
- Roles: Worked in a team of four students. Planned idea behind how the components of the system will interact. Helped build a robotic system with legos, sensors, servos, and motors. Coded logic for robotic system to use sensors to search for trash and return to base station. Programmed base station to sort incoming objects based on color.
- Technologies: C, Keil µVision, Arduino, various sensors, servos, and motors.

NATCAR - [Project report](#)

UC Davis, 2013

- Miniature autonomous racecar capable of racing down a track through the use of an optical sensor.
- Roles: Led a team of four members to design a racecar consisting of various hardware and software components. Wired circuit board for use with motors, servos, sensors, and a microprocessor. Programmed logic for racecar to track and follow the racetrack.
- Technologies: C, Stellaris LM3S8962, various sensors, servos, motors, and other electrical components.

Dead Reckoning

UC Davis, 2013

- iOS application that uses accelerometer to try and find location based on distance traveled.
- Roles: Solely wrote an iOS application to use accelerometer to find the current location. Researched dead reckoning and the flaws of relying on just accelerometer to find location.
- Technologies: Objective-C, OS X, Xcode, Core Motion.

15

UC Davis, 2011 - 2012

- A die game called 15 was built on a breadboard using combinational and sequential logic.
- Roles: Solely mapped out the logic needed to wire the breadboard. Ran functional and timing simulations in Quartus for verification. Completely built out the logic for the dice game with a state machine diagram.
- Technologies: Verilog, Altera Quartus II, ICs, sequential and combinational logic.