Ashwin Hariharan

• (213) 880-1316 • hari542@usc.edu • https://in.linkedin.com/in/ashwin-hariharan-4a7b17101 • https://github.com/ashwinusc93

An aspiring, passionate software engineer with a strong background in application and web development. My areas of interest include application and web development, databases, Cloud Computing, Big Data Analytics and Networking.

Currently a Masters student in Computer Science at University of Southern California, looking for internships in one of the above areas.

EDUCATIONAL BACKGROUND

Masters in Computer Science (2015 – 2017 expected): University of Southern California

Bachelor of Engineering in Computer Science: RV College of Engineering, Bangalore, India

TECHNICAL PROFICIENCIES

Programming Languages	C, C++, Java, Android, C#, Python
Operating Systems	Windows, Linux , Android
Database Systems	MySQL, Oracle 11g, Spatial Databases, MongoDB, JDBC + Swing.
Web Technologies	HTML/CSS, JavaScript, jQuery, AJAX, JSON, Bootstrap, HTML5, CSS3, PHP, XML, ASP.NET, REST applications deployed on AWS Elastic Beanstalk, Firebug
Cloud Computing	Windows Azure – Table Storage, Event Hub, Web Apps and cloud services, AWS Elastic Beanstalk
Networking	TCP/IP(IPv4 and IPv6), UDP, Socket Programming, wget, dig, whois

Hackathons and Events – Development of an Event Organisation app at HackSC using Facebook to register for the event and generating a QR Code on coming to the event. Using the Facebook info and Qr Code, find the location of the bus used for the events

Awards & Certifications

●2nd prize for the project "Development of Wireless Smart Energy Management System" in an exhibition judged by the Indian Institute of Science

● Part of the publicity wing of the organizing team of Hack59 hackathon in 8th mile fest

● Facilitator of Android Student Club organized by Google

● Member of the Rotaract Club of RVCE

Certifications:

- jQuery Essential Training - Lynda.com, License 7DC454 - http://www.lynda.com/ViewCertificate/7DC454D0F3694C82A5E718DE7C0BC941?utm_source=linkedin&utm_medium=sharing&utm_campaign=certificate

- Bootstrap 3 Essential Training - Lynda.com, License A72F04 - http://www.lynda.com/ViewCertificate/A72F04045115416DB3E2C74C2F57BD7F?utm_source=linkedin&utm_medium=sharing&utm_campaign=certificate

PROFFESIONAL EXPERIENCE

Software Intern, DromeBox Labs (06/2016 - Present)

- A project involving Volumetric Scene Capture
- The overall goal of Project UTU is to generate a toolkit for a near real-time multi-spectral volumetric point cloud capture / data pipeline optimization / data meshing (for 3d visualization) / mesh shading (for 3d visualization) / and pattern matching protocols for extrapolating object data from the data field.
- I am in the Database Creation and Management and UI teams of the project. I am using MongoDB as the backend Database to store RGBD data, Spectroscopic Data and 2D depth data coming from 3D cameras taking images from different angles and unifying the images into a single data set

- I am also working with OpenCV in Python to open the image obtained, perform difference mapping and analysis of the image and to put all the gathered data into a numpy array

➢ **Software Intern, Lam Research India 05/2014 – 08/2014**

- Developed a failure mode effects analysis system.
- Objective- Remove pre-production defects and failures through online team collaboration
- Worked on the Risk Catalog module of the project using Silverlight and ADO.NET backend

➢ **Software Engineer Intern, Zeesense Systems Pvt Ltd 05/2013 – 08/2013**

- Developed an Android application called CityFinder.
- Find places of interest from the current location within the range of 70km
- Current Location is computed using Google Maps API
- After current location is found, JSON parsing is done to parse data given by Places API based on the current location and range, and the places of interest are rendered on the map along with the information
- The application is published in the Google Play Store

PROJECT EXPERIENCE

Title: Development of a Wireless Smart Energy Management System

The aim of the project was to have automation of the usage of devices by building a rules engine which would behave in accordance with the energy data of the devices stored in the cloud.

Title: Querying and generation of stock market information using Markit on Demand Lookup and Quote API's (Web application as well as an Android App) - Perform HTML5 validation of user input, GET or POST to transfer data to the Web Server

PHP script retrieves the data and sends it to the Markit on demand restful Web Service
Use the Lookup web service in the Markit on Demand API to find the stock symbol corresponding to the user input and parse the data sent in XML and render it in the PHP form
If the lookup service returns the data, then Quote web service is called to get the company stock information and JSON parsing is done to display the stock data.
AJAX is used to perform asynchronous transactions to the PHP script on AWS to fetch the stock info and perform JS client side processing of data
Bootstrap is used for responsive UI design.Stock charts displayed using Yahoo Charts API.
Share stock info using Facebook Connect API and store favourites in browser local storage.
Historical charts information using Highcarts Highstock API. Display stock news using Google News Feed API

Title: Java based information system using Oracle Spatial Database(Database Systems, USC)

- Designed a Java Swing interface allowing the user to select and query features from spatial database objects in Oracle
- Executed dynamically generated queries via JDBC, processed returned data and displayed features selected on a map

Title: Med+Plus

An Android app using which the user can search for the medicine by locality and the distance range from his current location

This app used a MySQL database with PHP used to interact between the Android app and the MySQL database

Title: Single server-multi client chat application based on Wake-On-LAN

* A server has a C++ program written to build a magic packet. The magic packet is sent to the clients whose IP's are specified in the program

* The clients are booted after the Wake-on-LAN capable hardware receives the magic packet. When the system boots up, a shell file containing a chat application is executed and a socket based communication between the server and the clients takes place