Dixon Minnick

Software Engineer

415.797.7609
Dixon@Minnick.co

LinkedIn
Github

Up-to-date Resume
at Minnick.co

SKILLS

Github / SVN Subversion
HTML5 / CSS3
JavaScript / jQuery / jQuery Mobile
AngularJS (some)
MongoDB / PostgreSQL (some)
NodeJS / Express / Stormpath / EJS
Ruby on Rails (some)

C / C++
Java
Python (some)
ML (some)
Ruby (some)
Erlang (some)
Scheme (some)

EXPERIENCE

AcquireMedia, Burlington MA - *Software Developer*

MAY 2014 - AUGUST 2015

Responsibilities included: adding new features to and fixed bugs on NewsEdge.com web software platform using html, javascript, css, jQuery, AJAX and XML as part of a regular product development cycle. I acted as Lead Developer on a project to build a mobile-optimized platform containing a subset of features of NewsEdge using jQuery Mobile, JSON and a NodeJS server that I wrote; also Lead Developer to incorporate some of the aforementioned mobile features into iOS and Android native apps using Cordova.

EDUCATION

Tufts University, Medford MA - *BS Computer Science*

AUGUST 2011 - AUGUST 2015

Coursework: Web Programming, Web Engineering, Programming Languages, Object Oriented Programming for Graphical User Interfaces, Data Structures and Algorithms, Machine Structures and Assembly Language Programming, Discrete Mathematics, Introduction to Computer Science, Theory of Computation, Concurrent Programming, Introduction to Computer Security, Introduction to Algorithms, Computer Graphics

Tufts Men's Varsity Crew Team Tufts Alpine Ski Team Tufts Mountain Club

POJECTS

Lexicographic Word Ranker - (independent project)

An algorithm that runs in O(n) time to determine the Lexicographic Ranking of a word relative to any word constructed using the same letters. This is packaged in a web-page widget using AngularJS.

TicTacToe2 - (independent project)

A new twist on the classic game of tic tac toe, intended for 2 players, requiring a much greater level of strategy than the original game. Created using javascript and the HTML5 canvas. See the link below for the full game rules. This is an ongoing project. I plan to eventually enable a computer player.

fleetofthemalden.com - (independent project)

A training tool intended primarily for rowers on the Tufts Varsity Crew Team, allowing access to various erg workouts, training calculators and a calendar training plan. It is inspired by the resources available at tuftsoarsmen.org, but my site is mobile friendly and much easier to use. The site is an ongoing project.

Node Server - (adaptation of professional work)

An XML to JSON parsing server written in while working at Acquire Media, there was a point where we transitioned from XML server calls to JSON. I wrote this NodeJS server to to act as a proxy converting those server calls so the front-end engineers (myself included) could continue working using only JSON calls. This code is similar in structure to the server I wrote for Acquire Media. It uses NodeJS, Express, Requestify, xml2json and some custom Node packages I wrote myself.

Messagehub (Rails app) (iOS app) - (academic project) A simple Ruby on Rails project for pushing and viewing messages, with a corresponding iOS app that works with the API.

Frogger and Pokemon Frogger - (academic/independent project)

A working version of the classic Frogger game created using javascript as part of a Web Programming assignment, and a version (still under development) using pokemon sprites.

Concurrent Programming (Comp 50)

A simple game – For our semester project, my partner Evan Fincher and I designed a simple multiplayer game using a Java GUI coupled with distributed Erlang nodes (using Jinterface). We delivered a basic functioning prototype.

Machine Instructions and Assembly Language (Comp 40)

Using the C Programming Language, we wrote a number of programs including:

•An abstract data type for polymorphic 2 dimensional arrays, including a representation for 2D blocked arrays

•A program which used these 2D arrays to remove black pixels from the edges of images

•A suite of operations able to compress, decompress and perform basic transformations on images

•A Universal Machine able to run basic programs using a limited instruction set

•An Assembler capable of taking in more complex instructions and emitting instructions for the Universal Machine
We defused a "Binary Bomb" using a debugger when given only the Assembly language version of a program

Using Assembly Language we programmed a basic RPN calculator capable of addition, subtraction and division.