## Resume Summary

**Name:**Sunil Kumar
**Mobile No:** +91-8089106942
**Email id:** sunilkumar.gec56@gmail.com
**Current Location:** Cochin/ Kochi/ Ernakulam
**Preferred Location:**Anywhere in South India
**DOB:** 22/Apr/1988
**Gender:** Male
**Blog:** http://sunilkumarn.wordpress.com/
**Github: h**ttps://github.com/sunilkumarn

## Resume Details

**Title:** Web Developer(Ruby - Rails and Sinatra)
**Key Skills:** Ruby and its frameworks - Sinatra and Rails.
Good understanding of the MVC(skinny controllers-fat models) concept.
Database: Mysql
Nosql: Redis
Logging: Facebook scribe.
Big data managment: Hive
Other Languages: Python, html, css, javascript( library: jquery), node.js, awk, shell.
Platform: UNIX

## Professional Details

**Work Experience:** 2 yrs.
**Functional Area:** Web developement
**Area of Specialization (AOS Experience):** Web Development In Ruby
**Current Employer:** Mobme Wireless Solutions Pvt Ltd.
**Current Industry:** IT (Web development)
**Current Salary (Rs. P.A.) :** 3, 60,000p.a

## Qualification

**Highest Degree/Diploma:** BE/B.Tech
**Specialisation:** Computers
**Institute :** Calicut University

## Detailed Resume

Projects undertaken:

***Professional:***

**Ruby – Rails and Sinatra.**

Has undertaken or been a part of 6 web applications for leading telecoms in the country. As per the technical part, the project involves in depth understanding of Ruby frameworks like Sinatra and/or Rails. The MVC nature has been maintained throughout the Project and Yehuda Katz`s view of Skinny controllers fat models has been given thought and implementation. The restfulness has been adhered to.

Scribe has been quite used for logging. The tool developed by facebook and made open source was used quite extensively in all the projects that was undertaken, both for logging the visits as well as the events that happened in the application. The track record of each and every end user is taken care of.

Have satisifed the billing requirement logics of mutilpe clients. These works were done in Ruby as well. Have in tandem with clients like IBM to satisfy the billing requirements of Telecom companies.

Have used AMQP, Redis as and where deferring logic came up. The alternatives were switched based on the complexity of the task. Redis lists, though a bit primitive with would be the first choice because of the low setup effort but AMQP was the defintive choice wherever serious queing systems were needed and also because introspection was easier through their interface.

ORM that were used in all the projects was Active record: Active record migrations, named scopes etc have been dealt with pretty number of times.

**Node.js**
Recently created an api that was benchmarked at 500TPS. Since Ruby couldn't provide the necessary TPS was required, the application was ported into Node.js, an event-driven server side Javascript framework. Use of Nosql database Redis was also made to make this happen. The api requests were cached into redis and all other processes(including DB lookups and scribbing) were deferred. This helped in attaining the required TPS.

***Academical :***

**Implementation of a simple evaluator for Scheme in Python**
The metacircular evaluator for the Scheme language is explained in the classic CS text Structure and Interpretation of Computer Programs . A simple version of this evaluator was written in Python.

**Analysis Of TinyPython Virtual Machine v1.1**
TinyPython is a minimalist implementation of Python in 64K written by Phil Hassey. The working of the TinyPy VM was traced with ctags/gdb. The representation of a list in the VM was studied. The objective was to gain experience working with real code and also to understand the way an interpreter actually works.

**An AVL tree implementation for Python in C**
An AVL tree data structure was implemented in C; this was interfaced to Python using SWIG.

**Network simulation using the NS2 tool**
The Network simulation tool NS2 was used to model different network topologies.

**A minimal LOGO-like system in Python**
Py-LOGO is a very simple program to create geometrical shapes on a Tkinter canvas . It was implemented using Python, with abilities to move a given distance, turn a given angle clockwise or anti-clockwise, show or hide a moving figure of arbitrary shape, decide between pen down and pen up modes.

**Study of the introspection mechanisms available in Python**
The working of the built-in unit testing framework in Python, the unittest module, was studied. A toy version of the same was implemented by writing a small Python script. The objective was to study the utility of the introspection mechanisms available in Python.+