

HENRY RIVERA

(301) 541-3441 · resume@henryrivera.com
Washington DC Metro Area

Skills

Databases

MySQL	10.9 Years	Expert	SQL Server	6.0 Years	Intermediate to Advanced
PostgreSQL	9.7 Years	Intermediate to Advanced (Rusty)	SQLite	4.4 Years	Intermediate to Advanced

Languages

PHP	10.9 Years	Expert	Python	6.0 Years	Advanced (Rusty)
Perl	10.9 Years	Advanced to Expert	C#	2.4 Years	Intermediate to Advanced
JavaScript	10.9 Years	Expert	C++	8.8 Years	Intermediate to Advanced (Rusty)
XML	4.4 Years	Advanced	Java	2.2 Years	Intermediate
XSLT	4.4 Years	Advanced to Expert	LUA	7.2 Years	Advanced (Rusty)
HTML, XHTML, HTML5	10.9 Years	Expert	CSS, CSS2, CSS3	10.9 Years	Expert

Operating Systems

Windows Desktop	10.9 Years	Advanced to Expert	Linux / Debian	10.9 Years	Advanced to Expert
Windows Server	10.9 Years	Advanced to Expert	Linux, Ubuntu	6.0 Years	Advanced to Expert
FreeBSD	10.9 Years	Intermediate (Rusty)	Linux, Redhat	10.9 Years	Advanced to Expert
			Linux, CentOS	6.0 Years	Advanced to Expert

Libraries, Packages, Applications, and Technologies

Vim	Wordpress	jQuery	Memcache
Eclipse	Visual Studio	ASP.NET MVC	Doxygen
Apache	Postfix	SOA	Adobe Illustrator
IIS	BIND	Django	Adobe Photoshop
Tomcat	Exchange	IPTables	Courier

Employment History

Technotix, Inc.

Technical Warlord

July 2011 to Present
Washington DC Metro Area

- **Netlist, Inc: Enterprise Management System: Design, Implement, and Maintain Application**

- Overview
 - Application is an enterprise workflow management system which is integrated with data records from a MAS 200 accounting system. The primary goal of the project is to automate manual tasks and improve the efficiency of business operations. Workflow data is captured and entered from various departments: Sales Operations, Supply Chain Operations, Finance, and Senior Management. The workflow data is then integrated with records from the accounting system which is exported from MAS 200 on a scheduled basis. This design allows for the implementation of custom automation and data capture without an expensive and time consuming upgrade to a new accounting package. Netlist, Inc. is a computer memory technology company. Written using C# (ASP MVC), SQL Server, jQuery, and Perl.
- Sales Orders
 - Capture and report on additional information such as: true end customer, representative firms, sales managers, sales region, factory commit and ship dates, and production delays.
 - Reduced significant amount of manual labor by implementing automated sales order status customer email notifications.
 - Reduced significant amount of manual labor by providing real time reporting for sales representatives and managers.
- Invoices
 - Improved efficiency and performance monitoring by generating real time reports with combined information from sales orders and invoices. The additional information captured from sales orders allows for more comprehensive reporting.
- Action Items
 - Improves task completion and management through direct task assignments which can be related to various workflow objects: sales orders, meetings, projects, customers, and product lines.
- Customer Records
 - Capture data for: company, contact, sales admin, employee hierarchy.

- Meeting History
 - Improved record keeping and follow through for general and management meetings. Allows information to be record in real time during a meeting. Related action items can be created and assigned directly from the meeting history entry interface.
- **Eagle Publishing: Service Oriented Architecture Maintenance**
 - Vardy Interface: Add New Publication Dividend Pro. Added support for content management, subscriber security, bulk email system integration, and order processing.
 - www.nicholasvardy.com/dividend-pro/
- **12Feet: Remed System Features, Bug Fixes, and Maintenance**
 - Provided software development services to client from previous position at Support Services LLC.
 - Implemented support for Nessus API.
 - Fixed various bugs caused by PHP version upgrade.

Eagle Publishing, Inc. Lead Software Developer

March 2008 to July 2011
Washington, DC

• Service Oriented Architecture (SOA): Implement, Maintain, and Refactor Backend Services

- Overview:
 - Architecture is loosely based on REST-like principles. Service methods are implemented as simple GET or POST requests, which allow for increased flexibility in consumption from clients. This lets simple tools like curl and wget to be used as clients for the architecture and allows non-developers to query data from services. Services leverage Memcached for caching where possible, mostly in the Content and Aggregator services. Written in PHP, MySQL, XML, and XSLT. Example of services implemented are:
- Content Service:
 - Provides data manipulation and IO methods for content types. Example methods are: create, read, update, delete, and find. Example content types are: articles, authors, images, newsletters, product offers, and email templates. Content is stored in MySQL. Methods are called via GET or POST requests. Data is returned back in XML.
- Aggregator Service:
 - Allows multiple content types with complex relationships, or types needing complex processing to be combined or processed and then accessed as a single XML document result.
- Email Service:
 - Provides ability to send content (mostly articles) from CMS to email recipients. Email can be sent via local SMTP server or via bulk email service provider Alterian (was Paramount / Blue Hornet API).
- Payflow Service:
 - Provides an interface for charging credit cards. Uses Payflow Pro XMLPay API. Written in PHP and uses HTTP/XML to communicate with PayPal (provider) and consumers (web order processing).
- Memcache Service:
 - Allows direct manipulation of entries in cache including content and PHP sessions. Extremely useful for debugging and test purposes, as you can access the contents of PHP sessions in a clustered environment from other contexts such as unit tests.
- User Security Service:
 - Provides authentication via Active Directory. Written in PHP using MySQL and LDAP.

• CMS Interface: Develop, Maintain, and Refactor Content Management System

- Wrote CMS interface. Provides a web based interface for managing content stored in the content service. Edit forms are implemented in Ajax using jQuery library. Also provides list and search functionality. Authentication uses Active Directory which provides single sign on security. Written in PHP, XML, XSLT, XHTML, CSS, and JavaScript / jQuery.
- **Fabian Interface: Develop, Maintain, and Refactor Website (www.fabian.com)**
 - Implemented using PHP, XML, XSLT, HTML, CSS, and JavaScript. Site and article data are read from Content and Aggregator services as XML over HTTP. Order payments are handled using Payflow service. Site content was imported from existing versions of three separate websites which ran on our legacy CMS.
 - Implemented dual mode security where users may be authenticated against either Active Directory or Quickfill (a proprietary system). Edit links to CMS are displayed for Active Directory users which increases ease of content management.
 - Wrote an SQL based caching service for Quickfill (subscription fulfillment system). The initial implementation talked to Quickfill directly via ODBC but query times were far too slow. Written in PHP and MySQL.
 - Implemented Financial Inbox, Lemon List, and ETF Performance Report using Ajax. Data for these interfaces is provided as XML by the Content service. Written using JavaScript / jQuery.
 - Implemented CSS based layout. Optimized images for site from design mockups using Adobe Photoshop.

- **Skousen Interface: Develop, Maintain, and Refactor Website (www.markskousen.com)**

- Implemented using PHP, XML, XSLT, XHTML, CSS, and JavaScript / jQuery. Site leverages existing SOA infrastructure which was implemented for the Fabian Interface project. Site content was imported from existing versions of five separate websites which ran on our legacy CMS. All of the features available in the Fabian Interface are available on this website.
- Refactored large sections of logic into a shared library which decreases overhead and improves usability. This will allow additional sites to be added quickly in the future.

- **Alterian Implementation: Switch from Paramount/Blue Hornet (Bulk Email Service Provider)**

- Designed and implemented email lists in new Alterian system. Previous design used separate accounts for each of the business segments: email acquisition and newsletters, paid subscription fulfillment, geopolitical marketing, and financial marketing. The new design features a single master email list which is broken down into business segments using smart lists. This allows the history and activity for an email address to be stored in one location.
- Modified Email Service to interact with Alterian API in place of Blue Hornet. Example features include: sending articles and newsletters from CMS/Content Service as emails to bulk recipient lists, adding new subscribers to recipient lists, managing subscriber preferences, and adding subscribers to opt-out lists. Written in PHP. Communicates with Alterian API via SOAP. Method results are returned in XML.
- Implemented Win32 service to modify global opt outs to business segment opt outs. When a spam complaint occurs, a recipient is added to the global opt out list in Alterian. This was undesirable as the business segments needed to be processed and handled as separate units. The service parses a report (XML), adds each recipient to the local opt out list for a particular business segment, and then removes them from the global opt out list. Written in PHP using XML and SOAP with Alterian API.
- Implemented a process for exporting and updating records from our subscription fulfillment system (Quickfill) into Alterian. Records are exported in DBF format and are then loaded into a MySQL database. The process then builds a complete list of subscribers in CSV format and uploads this list to Alterian where it is imported via FTP drop folder. Written in PHP using MySQL and Curl.
- Exported email recipient records from Paramount and imported them into Alterian. Records were exported in a combination of XML and CSV. Records were imported using CSV and FTP drop folder support in Alterian. Scripts were written in PHP and Perl. Imported over a million records.

- **Redstate: Develop and Maintain Website (www.redstate.com)**

- Wordpress Multi User (WPMU):
 - Production database contained 113,000 user accounts.
 - Implemented caching via Memcache. Caching is enabled for anonymous users only. Was able to add this feature with very small changes to the core by hijacking the output buffer in PHP. The contents of the sidebar plug-ins are cached separately. This change was necessary both in terms of performance and shared caching requirements for a five node cluster.
 - Implemented custom plug-ins to provide equal functionality of old Django based site (recent comments, recent diaries, site-wide tags, recommended diaries, front page posting, etc).
 - Rewrote site layout to use CSS positioning.
- Django (previous version):
 - Maintained initial version which was based on Django. This version was provided to Eagle by an outside consultant.
 - Implemented fixes for incorrect use of query model which increased performance. Also implemented query object caching subclass which allowed for atomic caching of individual results.
 - Wrote import script to transfer user and blog data from Django to WPMU.

- **Human Events: Develop and Maintain Website (www.humanevents.com)**

- Increased the maximum number of simultaneous page views. Fixed comments implementation where comments were not previously cached by Akamai. Modified JavaScript to load each comment as a separate document, which allows the comments to be cached by Akamai but still provide an up to date list of comments for an article. This increased the maximum number of simultaneous page views by an extremely wide margin. Previously, the internal web servers would lock up if linked from a popular site like www.drudgereport.com or a popular radio show like Rush Limbaugh.
- Implemented basic security for comments, allowing for individuals to be banned from the system.
- Add, update, and remove sections on home page as requested by editorial staff.
- Email Acquisition: Develop, Maintain, and Refactor System
- Modified system so that it communicates with Alterian. When a subscriber registers their information is stored in a cache in MySQL. A Win32 service runs in the background and adds these records to Alterian. Written in PHP using MySQL and SOAP.
- Redesigned web interface of acquisition campaign search and editor which improved usability.
- Refactored subscription process to allow an end user to select from a list of newsletters while signing up. Previously, the list was static and was not selectable by subscribers.
- Added capability for storing information other than email address (first name, zip code, etc).

- **Rivendell: Developed, Deployed, and Maintained an Open Source Radio Broadcast Application**
 - Implemented threaded audio playback in RDLogedit for use on Windows using Qt threads. RDLogedit is used to compile play lists from individual audio files for broadcast to air. Modified to compile using either Qt3 on Linux or Qt4 on Windows (using conditionals and compatibility calls) due to limited budget and cost of Trolltech commercial licensing.
 - Implemented roaming profiles for broadcast Linux workstations using NFS, KDE, and the PAM Mount library. This allowed broadcasters to easily access their data any available workstation.
 - Implemented a high availability NFS cluster for broadcast audio storage. Used Debian Linux (etch) and Heartbeat (2.1.3). A CRM style (version 2) configuration was used.
 - Researched and implemented native authentication from Linux machines to Active Directory servers. Implemented on Debian Linux using LDAP and Kerberos (libpam-krb5, libkrb53). Password authentication was performed using Kerberos, Name Services were provided by LDAP.
- **Plone CMS: Implemented Unicode Codepoint Translation in Plone for Website (www.rfa.org)**
 - Implemented UTF-8 code point translation for Mandarin and Uyghur management and preview sub sites in Python. This allowed content to be stored using a single character set but be displayed in a number of character sets. The Mandarin sub site supported traditional and simplified code points. The Uyghur sub site supported Latin, Cyrillic, and Arabic code points. This was done at the Zope library layer. It was necessary to interrupt the HTML output render stream in order to perform the translation.
- **Security Identification Badge Redesign: Redesigned RFA Security ID Badge**
 - Used Adobe Photoshop and Illustrator to modify the graphics used for the RFA ID badge. Supplied technical graphics support, printer configuration support, and layout advice to the project manager.
- **Microsoft Exchange: Deployed and Administered Microsoft Exchange Email Services**
 - Email services were previously provided by Postfix running on Linux.
 - Simplified services for end users by implementing domain aliasing for remote offices in Asia. This was configured using Iron Port (a filtering appliance) and Exchange.
 - Implemented LDAP query based distribution groups to work around conflicts with primary Active Directory security groups which existed before the migration.
- **TechWiki: Implemented a Wiki for use by the Technical Operations Division**
 - Increased quality and usability of technical documentation by planning, and installing a Wiki for technical documentation using the MediaWiki package.
 - Reduced management overhead by implementing Active Directory authentication support. Used Microsoft Certificate Services, OpenSSL, and PHP 5.
- **Podcast: Maintained and Refactored Podcast Services using PHP and Apache**
 - Reduced cost of maintenance by modifying directory scanning logic to exist in a single function, in a single file (library). Previously scripts were copied in their entirety for each Podcast that was added. Also merged customizations that had been implemented for separate language services into the single library.
 - Increased ease of administration and deployment by creating a binary Debian package.
- **News-X: Develop, Maintain, and Refactor Web Based News Story Aggregation System**
 - Implemented system daemons which polled for stories from various agencies using FTP, HTTP, and serial port communications. Written in Python, used MySQL.
 - Refactored class inheritance to reduce code duplication.
 - Increased search performance by implementing MySQL full text search. Previously wildcard SQL queries were used. Also increased speed by removing redundant queries used for match result counting.
 - Refactored to use UTF-8 encoding for multiple language support (English and Chinese). This allowed news stories to be stored in a single database table.
 - Revamped web interface using graphics and CSS (Photoshop, Illustrator).
 - Implemented code documentation using Doxygen.

Superlative Technologies, Inc.
Software Developer

May 2004 to August 2006
McLean, Virginia

- **Security Video Cameras: Implemented Video Management and Camera Control Applications**
 - Implemented a Pocket PC based network video feed viewer (MFC C++, MJPEG).
 - Implemented a large scale, web based situation room video surveillance management system using ASP/JScript, Microsoft SQL Server, Perl, XHTML, and JavaScript. This system provided a client control service and a centralized management interface. Workstations running the client service were responsible for displaying video and other surveillance data and could be controlled via the web interface.
 - Created a quad view, cross-browser, web based video feed control interface with JavaScript (and ActiveX for Internet Explorer). This interface supported the control, surveillance, and configuration of pan-tilt-zoom (PTZ) enabled video cameras (TCP/IP based).
- **General Duties**
 - Updated and maintained the company intranet (Linux, Apache, MySQL, and PHP). Refactored the code to be compatible with PHP5. Implemented custom reports for use by the Finance and Human Resources departments.

Xram, Inc.
Software Developer

October 2003 to May 2004
Chantilly, Virginia

- **Joint Readiness Training Center: Implemented Recruiting and Personnel Management Application**
 - Created and maintained a web based multi-user recruiting and consultant management system with PHP, Perl, and MySQL. This recruiting and management system was used to recruit, hire, and manage translators for use in military training operations (JRTC, Fort Polk).
- **Security Video Cameras: Implemented Video Capture and Camera Control Applications**
 - Implemented a windows video capture service in C++/DirectX which was used to capture video images from security cameras and store the captured images in JPEG format.
 - Created a cross-browser, web based video feed PTZ control and surveillance interface with JavaScript (and ActiveX for Internet Explorer, TCP/IP based).
 - Wrote a windows service in Perl to control a Pelco pan-tilt-zoom (PTZ) video camera via serial port. This service implemented the Pelco D Protocol (read/write).

Support Services, LLC
Software Developer and System Administrator

November 2002 to October 2003
Norwalk, Connecticut

- **Remed: Implemented Web Based Multi User Ticket Management System**
 - Wrote a web based management interface used for remediating security vulnerabilities. The security vulnerabilities are detected by a network scanner and output in XML format. Written in PHP using PostgreSQL.
- **Financial Reports: Implemented Web Based Financial Reporting System**
 - Wrote a web based management interface to a financial reporting system using PHP and MySQL.
- **Hosting: Implemented and Administered Virtual Email Hosting Services**
 - Setup servers to provide email services capable of hosting multiple domains. User account information is stored in MySQL. Email server is Postfix.
 - Implemented user management interface. Written in PHP and MySQL.

Hispanic Online
Web Design Consultant and Linux Admin

September 2001 to May 2002
Miami, Florida

- **Web Servers: Administration and Maintenance**
 - Apache, MySQL, and PHP compilation, configuration and troubleshooting.
 - Web server content backups and log file rotation and archiving (shell / cron scripts).
- **Hispanic Heritage Plaza**
 - Created templates using PHP.
 - Used Adobe Photoshop to create slices for publishing to web.

References

- **References furnished upon request.**