**Andrew Roessner**

[**Andrew.Roessner@gmail.com**](mailto:Andrew.Roessner@gmail.com)

**(908)-255-9622**

**Summary:**

* Over **5 years** of diverse experience in Information Technology, in all phases of software analysis: design, development and deploying N-tier web applications, client / server applications using **NodeJS, ExpressJS, MongoDB, Angular 2**, **TypeScript,** **ASP.Net** **MVC 5.0/4.5/4.0/3.5**, **ASP.Net Webforms 4.5/4.0/3.5**, **C#**, **Web Services**, **Web API**, **AngularJS**, **jQuery, JSON, Bootstrap, HTML, CSS, JavaScript and SQL Server 2012/2008, Visual Studio 2015/2012/2010.**
* Worked on **Angular2** Templates, directives, Routing and navigation, Components, Services, Pipes and Data Binding.
* Expertise in **AngularJS** controllers, directives, filters, factory and service resources, routings and events.
* Experience in building application using the **MEAN stack**: **MongoDB, AngularJS 2, Express.js and Node.js** to generate **RESTful API’s**.
* Experienced working in **ASP.NET** **MVC**. Using **ViewBag, ViewData** and **TempData** to pass data across controllers to views and controllers to controllers.
* Proven ability in creating **Views**, **Partial** **Views**, **View** **Models** and **Action** **Filters** in **ASP .NET** **MVC**.
* Experience in front-end web development and design using **HTML5, CSS3, Bootstrap, JavaScript, jQuery,** and **Ajax**.
* Experience in **SQL Server** Database Creating, Database Objects, Stored Procedures and User Defined Functions using T-SQL.
* Experience in developing **N-Tier** Architecture using Data Access Layer (**DAL**), Business Logic Layer (**BLL**) and User interface (**UI**), and Model View Controller (**MVC)**.
* Extensive Experience in accessing and managing data using **C#, ADO.NET,** and **Entity Framework**
* Worked on **GIT** for source code management and project management.
* Expertise in converting the requirement into the Technical Specifications and developed them using **C# and ASP.NET.**
* Worked extensively on creating and consuming **Web API**.
* Quick learner, Team player, Able to work independently.

**Skills and Interest:**

|  |  |
| --- | --- |
| **Languages:** | C#, Java, ASP.NET, Type Script, Java Script, jQuery, T-SQL, HTML 5, CSS 3 |
| **Technologies:** | ASP.NET 4.5/4.0/3.5, ASP.NET MVC 4.0/3.0, ASP.NET WEB API, Web Services, ADO.NET, Node.js, Express.js, Passport.js, Mongoose, AngularJs, Angular 2, SignalR, Entity Framework, AJAX, Json, nUnit, bootstrap |
| **Databases:** | SQL Server 2012/2008/2005, MySQL, MongoDB |
| **Tools:** | Visual Studio 2013/2012/2010, Visual Studio code, Atom, Fiddler, Postman |
| **Operating Systems:** | Windows 98/XP/Vista/7/8, Mac OS X 10.8 |
| **Methodology:** | Agile, Scrum |

**Professional Experience:**

**Liberty Tech Builders, Edison NJ Jun 2017 - Present**

**Role:** Angular 2 Developer

**Description:** The purpose of this project was to create an efficient Job Management System for Liberty Tech Builders.Liberty Tech Builders specializes in home repairs and needed the system to match incoming jobs with contractors that could do those jobs. Contractors ranged in both skill and experience so our solution would have to match them based on those criteria. Recruiters created job postings with necessary skills and experience that managers could approve or deny. Once jobs were approved, this system made it quick and easy for recruiters to identify contractors that met specific job requirements and attend to those jobs in a timely fashion. I created several api services that allowed me to edit, add, delete and get job postings as well as a service that allowed me to get a list of contractors that could handle a specified job. These services allowed me to create the web pages that showed the job postings based on varying criteria, edit a single job posting, add a new job posting, and get a list of contractors capable of completing each job.

**Responsibilities:**

* Used the **MEAN stack** (**MongoDB, AngularJS 2, Express.js and Node.js.**) to generate a **RESTful API**.
* Created our api server using **Node.js** modulesso our application could communicate with the database
* Created api services through the **Express** middleware that allowed us to get data from the api more easily and efficiently
* Utilized **Mongoose** to create the schema for our data models and create structure while still taking advantage of the flexibility in **mongoDB**.
* Parsed user input from the body of the web page into **JSON** format with the **Body Parser** module
* Created **API Service** that gets a list of applicants that matches with a job by its requirements
* Used **[(ngModel)]** to bind data from the model to the view and vice-versa
* Grouped Components and Modules that our application needed to function with a **Module**
* Specified Controllers and Views within our application using **Component**
* Generated new components for our project with the **Angular CLI**
* Used **Git** for source code management and version control

**Environment:**

MongoDB, Visual Studio Code, AngularJS 2, JavaScript, jQuery, CSS, Bootstrap, Node.js, Express.js, Mongoose, Command Prompt (Angular CLI), GIT

**Bruce Fox, Inc., New Albany, IN Jun 2016 - May 2017**

**Role:** AngularJS, .Net Developer

**Description:** Bruce Fox, Inc. is the premier custom awards supplier serving the promotional products industry. They design and produce the award solutions and branded displayable products of all kinds. The Customer Relationship management system is an application developed to increase customer management and follow-up, also provides an elevated understanding on how effective the company is converting previous quotes into actual sales. The system provides task alerts, emails and ability to follow up on customers in a timely manner. The proposed solution allows users to add notes to the customer files and can be used on any web enabled devices. The main objective behind developing this project is eliminate the use of inadequate system, which stores all the word documents on a shared drive and speed up the information retrieving system. The system included Quote Creation for two departments of the company: Bruce fox and DYR, Task Creation, User and role settings, customer settings, and contacts settings. The system was designed using Bootstrap concepts to allow seamless user interactions on various devices.

**Responsibilities:**

* Created and implemented the application using **AngularJS, MVC ASP.net, MongoDB, JavaScript, JQuery, CSS,** and **Bootstrap.**
* Used **Visual Studio Code** for project manipulation
* Specified our app dependencies with a **Module** through the **ng-app** directive
* Added controllers to the module to control data in different parts of the application
* Controlled what happened with data using **Controllers** specified through the **ng-controller** directive
* The **ng-model** directive was used to bind data from the model to the view and vice-versa
* **$scope** “passed” data between the view and the controller
* Used **angular validation** such as **$touched, $invalid** and **$dirty** for client-side validation
* Validated user login credentials through a **custom service**
* Utilized **MongoDB** for its flexibility and speed which was important for a CRM application
* Created a web api using **ASP .net Web API 2**
* Used **GIT** for source code management and version control

**Environment:**

AngularJS, MVC ASP.net, MongoDB, Visual Studio Code, JavaScript, CSS, Bootstrap, HTML5, GIT

**Flash Global Logistics, Inc., Pine Brook, NJ Jan 2014 - May 2016**

**Role:** MVC Developer

**Description:** Flash Global Logistics, Inc. has global presence of its 570 forward stocking locations in the Flash Network and its 13 Multi-Client Distribution Centers around the world. Flash maintains 4 Global Command Centers in Pine Brook, NJ, Mexico City, Mexico, Tokyo, Japan, and Amsterdam, Netherlands. Flash Global Logistics has a tool called FLASHTRAC©, Flash’s proprietary, real time inventory visibility web based system created specifically for the critical parts and products industry, Flash takes standard and customizable tracking, reporting and order fulfillment, Invoicing Process to new standards for the Client. Flash-Lite is a mobile App targeting smart devices to allow users to access few key modules of FLASHTRAC application using their mobile devices. Flash-Lite provides functionality to users such as real time inventory tracking, real time order tracking and order placements and chatting with customer care for tracking.

**Responsibilities:**

* Designed and implemented the application with **ASP.NET MVC 4**, **SQL Server 2012,** **Entity Framework, jQuery, AJAX**, **HTML5**, **CSS**, **Bootstrap** and **JavaScript**.
* Responsible for end to end application scoping, design, planning, development, implementation and post-implementation maintenance and support
* Created **View Models** to encapsulate all models and classes used on different views.
* Used **Entity Framework database first** approach to create data models and for data retrieving and manipulation using **LINQ** and **Lambda Expressions**.
* Created **Stored Procedures, Views** and **Triggers** in **SQL Server**.
* Created **Custom Data Validation Attributes** for data fields.
* Created **Custom Action Filter Attributes** for the Action methods in Controller.
* Implemented **N-tier** architecture (UI layer, Business logic layer, Data access layer).
* Implemented Authentication and Authorization using **ASP.NET Identity**.
* Implemented push notifications system using **SignalR**
* Implemented bundling and minification of scripts and styles to improve performance.
* Different Layouts were created for having different design for different user roles (customer, employee).
* Used **jQuery** and **JavaScript** to implement client-side functionality.
* Used **ASP.NET AJAX Helper classes** and **jQuery AJAX** for making asynchronous AJAX calls in views.
* Used **JSON** data to send and receive information along with GET, POST, PATCH, DELETE **AJAX** calls.
* Actively involved in Bug Fixing during the build and deployment process and debugging.
* Used **Team Foundation Server** for source code management and version control

**Environment:**

ASP.NET MVC4, Microsoft SQL Server 2012, Entity Framework, JavaScript, jQuery, Ajax, HTML5, CSS, Bootstrap, Team Foundation Server

**Nestle, Arlington, Virginia Aug 2012 - Aug 2013**

**Role:** ASP.NET Developer

**Description:** Nestle is world's largest food and beverage company. The purpose of this project was to create an inventory management system for Nestle. Nestle is the world’s largest food and beverage company driven by enhancing the quality of life and contributing to a healthier future. The project I worked on was an Enterprise and Commerce IT (ECIT) project, which integrates Nestle products and services into channel program offerings for sale to customers around the world. This project supports pricing, order management, order tracking, product fulfillment, product shipping, entitlements and invoicing processes. Management could search their products to identify top sellers or items that rarely sell for example. The product list shows products that are close to out of stock first by default so items to be reordered can be easily identified. I created the page that allowed storeowners to search for specific inventory items or items that met desired criteria. I also created a login page which checked their credentials against the database and if validated stored a token in the session that allowed the user to move from page to page.

**Responsibilities:**

* Created web application using **Asp.net** webforms and styled web pages with **CSS** and **HTML5**
* Connected to the database and manipulated data using **ADO.net**
* Used a **Master Page** with **Bootstrap** to keep layout consistent between pages within the application
* Redirected users who were not logged in to Login screen using **Session** variables
* Used **Cookies** to remember Username on Login screen
* **Asp validator controls** were utilized for client-side validation on text fields
* Relieved stress on the database by using **cache** to store data pulled from the database
* Used **User Controls** to define properties, methods, and existing web server controls for login Control
* Connected to **Microsoft SQL Server** for data retrieval and manipulation
* Used **GridView** to display data tables(enabled paging and formatted **GridView** to be visually appealing)

**Environment**:

ASP.net, ADO.net, CSS, HTML5, JavaScript, JQuery, Bootstrap, Microsoft SQL Server 2012, GridView, Team Foundation Server