Tango Tew

Software Developer

Industry Equivalency

18	14	14	12	12
C#	ASP.NET MVC	ASP.NET Web API	SQL	Angular

Equivalency in months

ABOUT ME

I am a full-stack developer with extensive experience in building multi-tier applications using multiple software platforms from client-side to server-side applications. Built server-side applications, Services, libraries and testing components using REST API, .Net Core, ADO.NET Entity Framework and SQL database, Created responsive client-side applications utilizing ASP.NET Core MVC, JavaScript, HTML, Angular, and React Js.

- ➢ Highly experienced with **.Net Core, C#, CSS, HTML, JavaScript, Angular, ASP.NET Core MVC, ASP.NET Core Web APIs, Relational SQL, and None Relational Databases.**
- ➢ Utilized **CSS, Bootstrap, HTML,** and **JavaScript** to create a semantic and responsive web with a high user engaging experience.
- ➢ In-depth knowledge of front-end **Single Page Application** development using the **Angular JavaScript framework.**
- ➢ Utilized **Pipeline Methodology** to transition from development to operations by implementing **Continuous Integration** with the help of **Docker Hub, Azure DevOps** and **Microsoft Azure** platforms integrated with **GitHub**.
- ➢ Leveraged **MongoDB, MySQL, MSSQL,** and **PostgreSQL DB** to define, store and manipulate data relational as well as non-relational data.
- ➢ Experienced with Object relation mapping using **ADO.NET, Entity Framework, LINQ.**
- ➢ Leveraged **ASP.Net MVC Angular** to create a responsive and well structured client-side presentation layer.

➢ Created **Rest APIs** using **SOA** principles to utilize a fast, flexible and language-independent web service and consumed it in the **ASP.NET MVC** web app.

➢ Utilized **JSON** and **XML** to make **API** calls or send serialized data to the server and deserialized those data by parsing them upon return to the client.

➢ Ensured user input validation to be inline with current program functionality using to validate **Client-Side** as well as **Server-Side** validation **rules**.

➢ Included **Data Annotation Validators** or **Attributes** to validate user input on the Server-Side.

➢ Adhered to **SOLID** principles in software development for code readability, reusability, and maintainability.

➢ Leveraged **Git version** for code version control and **Github SCM** for code storage and team.

➢ Utilized Git for automated pipelining integrated with **Azure DevOps** and docker hub.

➢ Deployed and managed infrastructure as code (**IaC**)using **AWS Cloud formation.**

Education & Certification

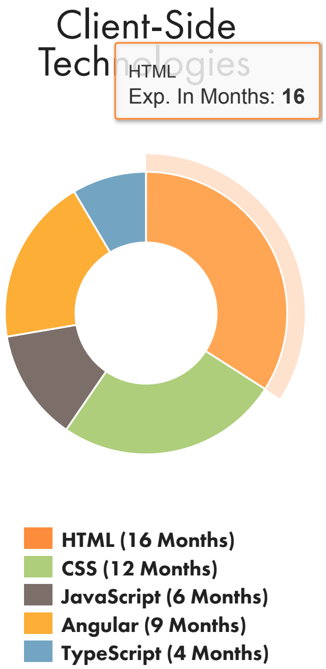
Bachelor's Degree - Computer Science

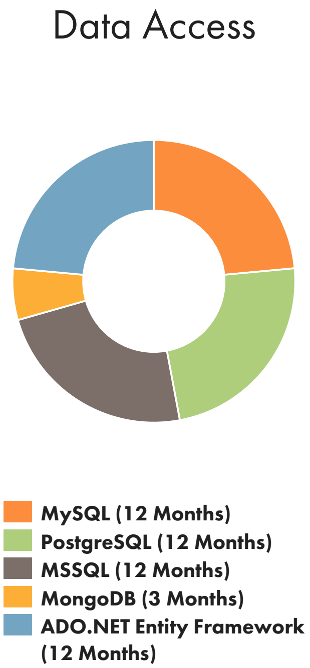
🏛 Dire-Dawa University - Jun 2016

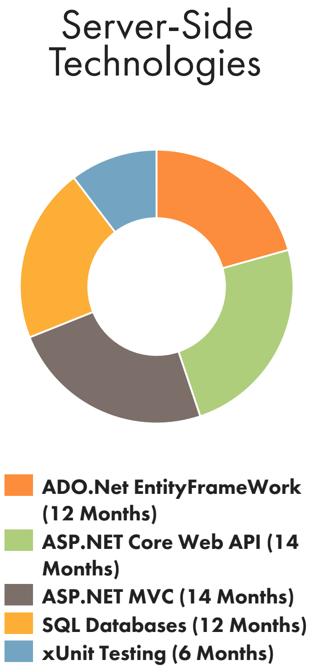
GPA : 3.7

Skill Matrix

All durations represent Equivalency







Client-Side Technologies

- HTML
- CSS
- JavaScript
- Angular
- TypeScript
- ReactJs
- Bootstrap

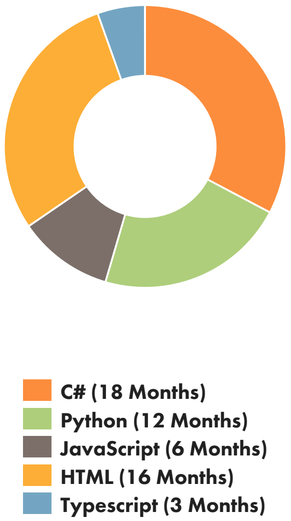
Data Access

- MySQL
- PostgreSQL
- MSSQL
- MongoDB
- ADO.NET Entity Framework
- LINQ

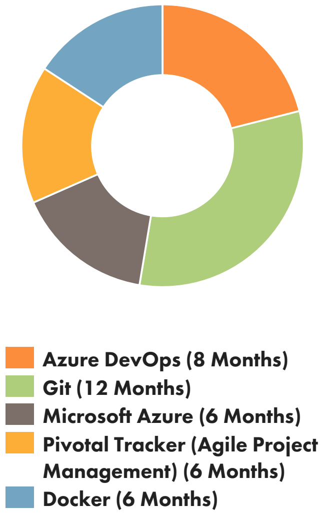
Server-Side Technologies

- ADO.Net EntityFrameWork
- ASP.NET Core Web API
- ASP.NET MVC
- SQL Databases
- xUnit Testing
- NoSQL Database

Languages



Development Tools



Languages

- C#
- Python
- JavaScript
- HTML
- Typescript
- Java

Development Tools

- Azure DevOps
- Git
- Microsoft Azure
- Pivotal Tracker (Agile Project Management)
- Docker
- Kubernetes
- AWS

Projects

RVTR Campsite

Campsite is an open source hospitality management platform showcasing the future of connected travel, enhancing experiences for any travel plan around culture, leisure, nature or venture. The mission is to provide a set of services to support personalized experiences for pleasure travelers; modern workplace experiences for business travelers; analytics, insights, and recommendations for hospitality managers.

Roles / Responsibilities

- Leveraged Microservices Architecture patterns to design Microservices for RVTR Campside.
- Played a major role on team environment dynamics with the ability to contribute expertise and follow leadership directives at appropriate times.
- Created a RESTful Web API service using ASP.NET Core Web API to generate data for RVTR Campsite which was then consumed on the front-end by Angular.
- Integrated and ensured the implementation of Single Page Application (SPA) architecture using Angular routing.

- Debuged Angular code using browser tools such as ng-inspector and the developer console in various browsers.
- Worked in an agile environment using Pivot Tracker to deliver higher-quality software far more rapidly.
- Performed unit test on libraries and APIs using xUnit and on Angular using Jamine and Karma.
- Leveraged GitHub to build, test and deploy Reservation Microservice for RVTR Campside application.
- Designed and deployed API specification with testing case(Swagger).

Environment / Technologies

C#, ASP.NET Web API, MongoDB, ADO.NET Entity Framework, Angular, JavaScript, Testing, DevOps, Docker, HTML5, CSS3, SQL, BDD, TDD, Microsoft Azure, .NET, Continuous Integration, Microservices, TypeScript, Jasmine, Karma, REST, Kubernetes

Revature Appointment Portal

When it comes to making a doctor's appointment, patients face many obstacles from choosing the right doctor and scheduling an appointment to keeping track of their appointments. All this can be more simplified by using an organized system where patients can find all the information they need with a press of a button. Instead of using the traditional and unorganized way of making an appointment which consumes more time and energy, Our system will manage and organize the process of making appointments with ease and flexibility to save time and effort for both the patient and the doctors.

Roles / Responsibilities

- Used Visual Studio Code as an Integrated Development Environment of choice.
- Created DAO layer using ADO.Net Entity Frameworks for REST APIs to efficiently communicate with database.
- Utilized Code-first approach to have more control over creation of database while focusing on the code.
- Utilized REST API implemented with ASP.NET Web API to leverage SOA design pattern.
- Implemented automated continues Integration to easily build, test, pack, analyze and deploy, leveraging Azure DevOps, Sonar Cloud, Docker Hub and Azure App Services.
- Utilized docker compose to create images for ASP.NET Core MVC, Angular and ASP.NET Core Web API and used docker hub as a remote Registry.
- Implemented client-side using ASP.NET Core MVC with JavaScript and used HttpClient as the protocol to make calls and consume the APIs.
- Consumed APIs by implementing client side with Angular to utilize SPA functionality.
- Leveraged Github for team coordination and version control.
- Implemented Unit Testing for both ASP.NET MVC and ASP.NET Web APIs.

Environment / Technologies

C#, ASP.NET Web API, SQL Server, ADO.NET Entity Framework, HTML, JavaScript, CSS, Microsoft Azure, ASP.NET MVC, Angular

Pizza Parlor

piZZZa is a 24-hour pizza ordering and delivery service application. Order one of our specialty made pizzas, make-your-own pizza with a variety of ingredients or indulge in a footlong churro. Be sure to check out our other sides, desserts, sauces, and drinks when creating your order. This application not only makes the process of ordering food online easier, but from anywhere at anytime.

Roles / Responsibilities

- Implemented DAO and MVC design pattern using ADO.Net entity framework and ASN.Net MVC respectively.
- Used code-first approach to create my database based on the written Pizza ordering POCO classes.

- Leveraged a code-first 3NF formulation on database to increase efficiency in retrieving and storing data using ADO.Net entity framework
- Designed responsive web page that can be easily utilized on any platform.
- Monitored my code using static analyses and optimized them to achieve A grade, and optimized build time using Sonar Cloud.
- Used git version control to manage the code and shared it with my manager on Github as the agreed SCM.
- Built dynamic HTML pages using Razor to take user choices to make an order.
- Implemented xUnit testing to determine if the behavior of the function is as expected.

Environment / Technologies

C#, ASP.NET Web API, SQL Server, ADO.NET Entity Framework, HTML, JavaScript, CSS, Angular