JACOB PHILLIPS
SOFTWARE ENGINEER

📞 720-467-3290

✉ JacobSPhillips33@gmail.com

🔗 linkedin.com/in/JacobPhillips33/

📄 jacobphillips33.github.io

🐙 github.com/JacobPhillips33

🌀 codewars.com/users/JacobPhillips33

🔺 Project Videos and Images

SKILLS

.NET Core | ASP.NET Core MVC

C#

SQL – MySQL

HTML | CSS | JavaScript

RestAPI | WebAPI

ORM – Entity Framework | Dapper

LINQ | JSON

Visual Studio | Visual Studio Code

EDUCATION

Full-Stack Software Engineering
TrueCoders
View NFT Certificate

BS in Petroleum Engineering
Minor in Economics and Business
Colorado School of Mines
View Diploma

Additional Certificates – *View All*

C# – Sololearn
SQL – Sololearn

SUMMARY

I am a Petroleum Engineer turned Software Engineer who specializes in a variety of languages and frameworks including ASP.NET Core MVC, C#, and SQL. I enjoy taking complex problems, breaking them down into manageable parts, and then turning the process into an application that will solve this problem automatically from now on. While I have fun developing some projects on my own, I find it highly rewarding to collaborate with and learn from other individuals who share my passion.

EXPERIENCE

Honeywell Properties   2015 - 2022
President and Co-Founder

- Co-owner and operator of small value-add real estate investment company
- Lead property management teams for company-owned rental units
- Hired and managed contractors to complete repairs, maintenance, and improvements on properties
- Responsible for finding, evaluating, and acquiring properties
- Communicated directly with company investors

Disclaimer:
  While everything listed here is true, it sounds more impressive than it was. The reality is that I scraped along for years trying to get this business off the ground while supplementing my income with various side jobs, gigs, and contract work.

DXP|Quadna   2012 – 2015
IPS Sales Engineer | Applications Engineer

- Lead teams to deliver highly technical and customized innovative pumping packages
- Directly in charge of communications between clients and our engineering and management teams
- Responsible for making sure all materials, instruments and equipment were included and correctly specified
- Responsible for designing and engineering an innovative pumping solution to meet the client's needs
- Created P&IDs (Process and Instrument Diagrams) to illustrate the process and equipment needed

PROJECTS

Customizable Workouts

ASP.NET Core MVC | C# | SQL | HTML | CSS | JS | Rest API | Entity Framework | Dapper | JSON | LINQ

This ASP.NET Core MVC application allows you to choose from over 1,300 exercises to make yourself a list of favorite exercises and/or create, customize, and delete as many workouts as you want. Time to get fit!

Project Highlights:
- Utilizes Rest API to build MySQL database with over 1,300 exercises
- Utilizes full CRUD functionality with MySQL database
  - Ability to Insert, Select, and Delete exercises from the Favorite Exercises table
  - Functionality to Create, Alter, and Drop tables from the database – these are the custom Workout tables
  - Capability to Insert, Select, Update, and Delete exercises from each of the user created Workout tables
- Search and sort features allow users to sort and/or filter by every column in tables
- Implemented encapsulation to hide sensitive information such as the connection string

Geolocation – TacoParser

C# | .NET Core | Geocoordinates | TDD | Logging | Console Application

Provided a CSV file containing Taco Bell locations in Alabama, this C# .NET Core Console application will determine which two Taco Bells are farthest apart and provide their distance apart in miles.

Project Highlights:
- Parses through the provided CSV file for each location's name, latitude, and longitude
- Converts the latitude and longitude into a Geocoordinate
- Utilizes a nested loop to compare the distance between all of the Geocoordinates
- Utilized logging and TDD (Test Driven Development) to assist with the coding process
- Modifiable to read any CSV file specifying name, latitude and longitude

Possible Patterns Calculator

C# | .NET Core | Graph Theory | Combinatorics | Geometry | Algorithms | Mathematics | TDD

This application will calculate the number of patterns possible on a 3x3 grid given a user specified starting point and a user designated number of movements for the pattern.

Project Highlights:
- Created multiple algorithms and methods to calculate the number of possible pattern solutions
- Utilized TDD (Test Driven Development) to ensure the methods were calculating correctly
- Grid:
  > A B C
  > D E F
  > G H I
- Rules:
  - Grid points will not be repeated; they can only be used once, at most.
  - Patterns will not pass through a point unless it has already been used. For example: Cannot go from A -> G unless D has already been used.
  - Solution example: Making 3 moves beginning with grid point A has 31 possible patterns.

Weather Map API
C# | .NET Core | Web API | JSON | Console Application

This C# .NET Core Console application provides up to date weather in your area. Users simply enter a city of their choice to see what the weather is like right now!

Project Highlights:
- Utilizes a Web API to call and parse a JSON file containing weather for the chosen city
- This weather information, which includes the temperature, pressure, humidity, wind-speed, and a description such as "snow," is then presented to you in a readable format.

Sudoku Solution Validator
C# | .NET Core | Algorithms | Data Structures | Console Application

This C# .NET Core Console application will evaluate a completed Sudoku Solution to determine whether it is a valid solution. It will return "True" if it is a valid solution or "False" if it is not.

Project Highlights:
- Created multiple algorithms to evaluate and validate each aspect of the inputted data structure
- Algorithms ensured each row, column, and 3x3 grid contained all of the digits from 1 to 9

"Best Buy" Database
ASP.NET Core MVC | C# | SQL | HTML | CSS | Rest API | Dapper | JSON | LINQ

This ASP.NET Core MVC application utilizes CRUD functionality to modify the products contained in the tables of a mock BestBuy Database.

Project Highlights:
- Utilizes Dapper (an ORM) to interface with a MySQL Database containing mock "Best Buy" data
- Ability to Insert, Select, Update, and Delete products contained in the database
- Implemented encapsulation to hide sensitive information such as the connection string

Roman Numeral Converter
C# | .NET Core | Algorithms | Console Application

Don't know Roman Numerals? No problem! This C# .NET Core Console application converts from Roman Numerals to Arabic ("regular") Numbers or from Arabic Numbers to Roman Numerals.

Project Highlights:
- Created multiple algorithms to perform the conversions
- Performs calculations for any number between 1 – 3999
- Example conversions:
    - MMMDCCXXIV = 3724
    - 1986 = MCMLXXXVI

Additional Projects
Random Activity Generator | Rock Paper Scissors Game | Ron vs Kayne API | Exercise Database API