TSEA-ERP School

Director: James Mueke

CONTENTS

1. Software Development Course Syllabus

MAKE. SOLVE. CODE.

Change Starts With Code

FROM BEGINNER TO DEVELOPER
Today is your day.
Love problem solving, puzzles, creating meaningful things?

2. DIVE IN

Resurface with in-demand skills

When you learn to code, you have just increased your worth.
18 Real World Projects, one amazing career.

3. What’s The Plan?

Make. Solve. Code.

50:50

This is a Partnership

With the right combination of student and instructors, anything is possible. We are counting on you to put in the work, build amazing things, bring your ideas to life, and solve real-world problems. We provide the support, structure, empathy, knowledge, and passion to help you succeed.

Project Based Learning

This Immersive Bootcamp was created with our top-level goal of making the hiring partners happy first. When they are happy, you get offers, you're then happy, and so are we. This is the traditional classroom flipped on its head, kicked around a bit, and designed for real-world results.

It's About You

We will allow you to explore, flex your most powerful muscle...your brain. We are creating problem solvers. Teaching to the test? Not a chance. Spoon feeding you info? Never. Allowing you to become a valuable asset because you understand how to dive deep and surface with a solution? Yes.

4. Goal

Fill the talent gap that exists in every industry, every state. Companies from startups to big smokestacks want more Software Developers. Higher Ed has dropped the ball and doesn't create enough creators and problem solvers. That's where you and TSEA-ERP School come together to learn to code.

5. JavaScript, HTML, CSS

JavaScript Fundamentals

JavaScript basics and foundational concepts.

Variables, Data Types, Operators

Understanding core programming concepts.

Flow Control, Loops

Learning control structures.

Functions

Writing reusable code.

Arrays, Objects

Data structures in JavaScript.

Callbacks

Handling asynchronous operations.

Debugging, Testing

Ensuring code quality.

Git & GitHub

Version control practices.

Best Practices

Coding standards and methodologies.

Problem Solving

Approaches to coding challenges.

HTML & CSS Bootstrap

Web development basics.

Adobe XD, Wireframe & Prototype

Design and prototyping tools.

Mobile-first Design & Mobile Responsiveness

Creating responsive applications.

Project 1 Example

Algorithms - Problem Solving Problems

Project 2 Example

Day Trip Generator

Project 3 Example

Portfolio Site

Project 4 Example

Most Wanted

6. Course Syllabus Sections 3-4

Python OOP, Algorithms, Data Structures

Python Fundamentals

Key concepts in Python programming.

Object Oriented Python

Understanding OOP principles.

Classes, Objects

Python's structure for code organization.

Inheritance, Polymorphism, Encapsulation, Abstraction

Core OOP concepts.

SOLID Design Principles

Best practices for software design.

Collections - List, Dictionary, Set, Tuple

Various data structures in Python.

Algorithms, Data Structures

Essential algorithmic concepts.

Time Complexity & Big O Notation

Analyzing algorithm performance.

Design Patterns

Common solutions to recurring problems.

Project 5 Example

Robots vs. Dinosaurs

Project 6 Example

Rock, Paper, Scissors, Lizard, Spock

Project 7 Example

Build with Algorithms & Data Structures

Project 8 Example

Build with Design Patterns

7. Course Syllabus Sections 5-7

Python Web Frameworks & Web Services

1. **Database Concepts**
   Understanding databases and their use.
2. **MySQL**
   Learning SQL and database management.
3. **Database Normalization**
   Structuring databases effectively.
4. **Queries**
   Executing database commands.
5. **Entity Relationship Diagram**
   Mapping database relationships.
6. **REST APIs, Web Services**
   Learning about web services.
7. **Python Django & Web Application Frameworks**
   Building web applications.
8. **Advanced Design Patterns**
   Exploring higher-level patterns.
9. **Templating**
   Creating dynamic web content.
10. **Dependency Injection**
    Managing dependencies effectively.
11. **Data Mining, Data Analyzation**
    Working with data for insights.

Project 9 Example

SQL Queries

Project 10 Example

Superheroes

Project 11 Example

Trash Collector

Project 12 Example

Music Library

8. Course Syllabus Sections 8-10

React.js & Front-End Libraries

React.js

Understanding the React framework.

Class Components, Function Components

Differentiating component types.

State

Managing component state in React.

React Hooks

Using hooks for functional components.

Redux

Managing global state.

Redux Hooks

Integrating Redux with hooks.

Connecting Django Back-end & React.js Front-end

Full-stack integration.

Agile

Understanding Agile methodologies.

Data Visualization

Presenting data effectively.

Project 13 Example

Flashcards Study Tool

Project 14 Example

YouTube Clone

Project 15 Example

Social Media Platform

9. Course Syllabus Sections 11-12

C/#, ASP.NET Core

1

C/# Fundamentals

Basic concepts in C/# programming.

2

Object Oriented C/#

Implementing OOP in C/#.

3

ASP.NET Core

Building web applications with .NET.

4

MVC Design Pattern

Understanding the MVC architecture.

5

MSSQL

Learning about Microsoft SQL Server.

6

API Architecture & Build REST API

Creating APIs.

7

React.js Front-end

Integrating with React.

8

Axios

Making HTTP requests.

Project 16 Example

Box Service

Project 17 Example

Video Game Data Analyzation & Visualization

10. Course Syllabus Sections 13-14

Final Capstone Project

Project 18

Your individual final project where you bring all of your skills together. Every project from concept to design is developed by each individual student.