PROGRAMMER

AWDIZ FULL STACK DEVELOPER PROGRAM

MEAN MERN STACK SYLLABUS

98%

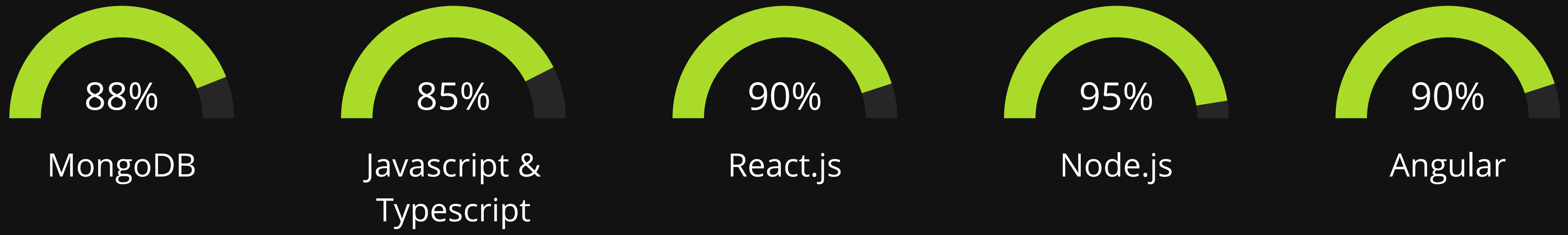
Placement Success

WELCOME TO FULL STACK PROGRAM

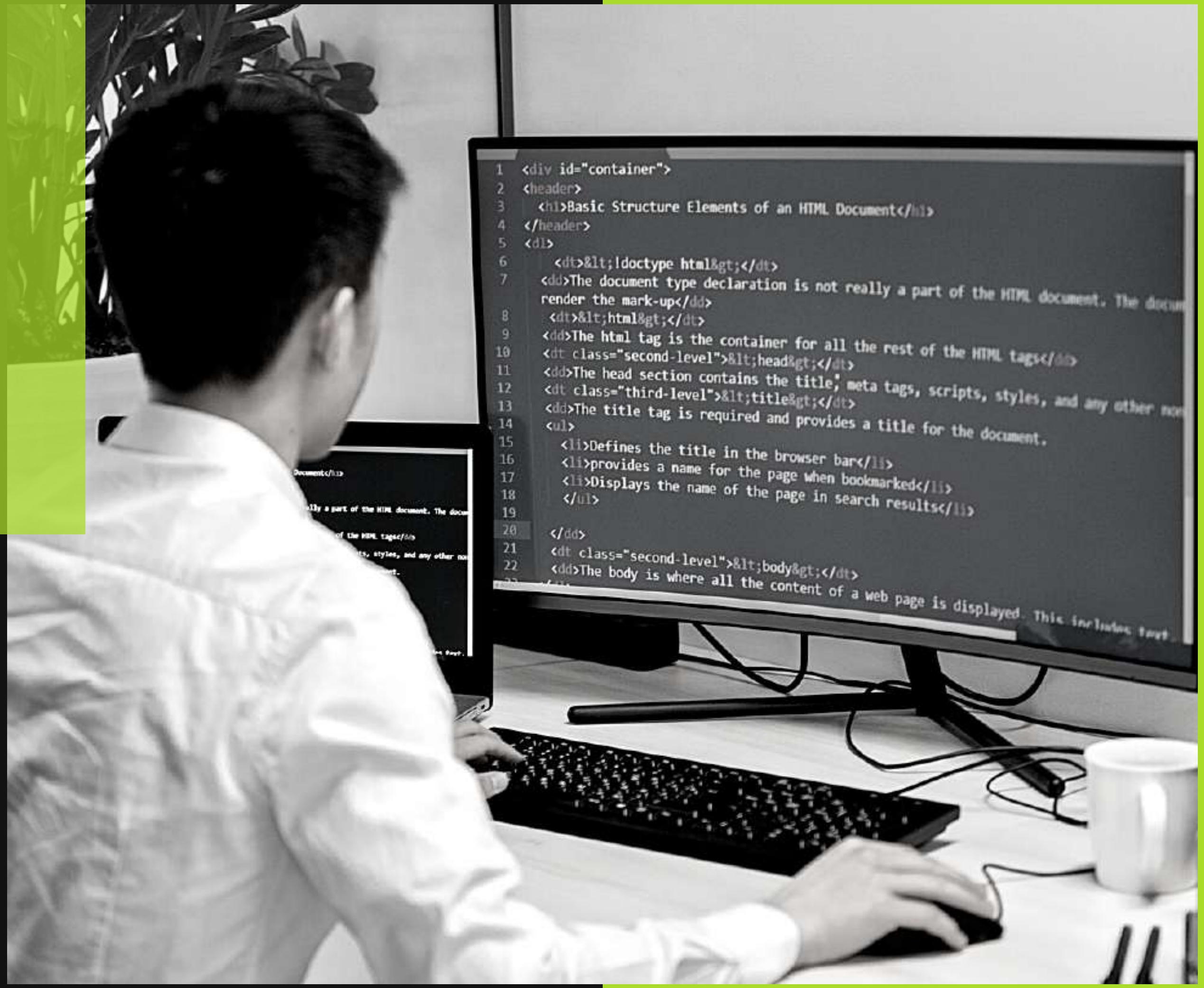
Over the past five years, AWDIZ has achieved an impressive 98% success ratio in placing students trained in the MERN stack. This remarkable accomplishment highlights our commitment to providing high-quality education and preparing students for successful careers in the field of web development. Our comprehensive MERN stack program equips students with the necessary skills and knowledge to excel in the industry

Through a combination of rigorous training, hands-on projects, and industry-relevant curriculum, we have consistently produced skilled professionals who are in high demand by employers

MASTERING MEAN MERN STACK SKILLS AT AWDIZ



Mastering these skills in the MEARN stack empowers students to become proficient full-stack developers capable of building modern, scalable, and efficient web applications using the four technologies that comprise the stack.



HANDS-ON LEARNING OVER 10+ LIVE PROJECTS

REAL-TIME CHAT APPLICATION
LIKE WHATSAPP

TASK MANAGEMENT SYSTEM
LIKE TRELLO

E-COMMERCE STORE
LIKE AMAZON

SOCIAL MEDIA PLATFORM
LIKE FACEBOOK

BLOGGING PLATFORM
LIKE MEDIUM

JOB PORTAL
LIKE NAUKRI

RECIPE FINDER APPLICATION
LIKE FOOD NETWORK

NEWS AGGREGATOR LIKE
GOOGLE NEWS

MUSIC STREAMING PLATFORM
LIKE SPOTIFY

ONLINE LEARNING PLATFORM
LIKE COURSERA

FULL STACK DEVELOPER COURSE SYLLABUS

HTML / HTML5

Applying Designs to WireFrames with HTML5 and CSS3
Introduction To Web And Basic HTML Tags
Building our Own HTML5 Video Player
Building Responsive Websites with HTML5 and CSS3
HTML5 and CSS3 Site Design our website project
Working with Forms
Deeper Understanding of Client-Server Communication
Understanding HTML5 Input Types

CSS / CSS3

Introduction to CSS, CSS Inheritance and Various Selectors
The Cascade, Specificity and Style Resolution
Important CSS Properties
Website Wireframing with HTML5 and CSS3
Structuring Content Using HTML Tags
Explaining Bootstrap latest version 4
Using various components to make UI
Making complete pages

TypeScript Module 1

Introduction to TypeScript
Why TypeScript? & Setting up TypeScript
Using Types for a better Code
Understanding the TypeScript Compiler
TypeScript and ES6
Using Classes to create Objects
Namespaces and Modules

TypeScript Module 2

Doing Contract Work with Interfaces
Generics
Behind the Scenes with Decorators
Using JavaScript Libraries (like jQuery) with TypeScript
TypeScript Workflows
Example: Using TypeScript together with ReactJS

TypeScript Module 3

JavaScript vs TypeScript
Type annotations, variable declarations, basic datatypes, type inference
Advanced types, type erasure and error behavior
Classes, constructors & methods, inheritance & polymorphism, access modifiers

JS Language Fundamentals

What Is Javascript
Internal Vs. External
document.write()
Alert & Confirm, Prompting Info
Prompting For Info
Conjugation & Arithmetic
Redirecting And New Windows
Empty Links & String Manipulation
Comparisons, Functions & Events
If Else, Else If, Switch Statements
getElementById()
Escaping, Arrays
For Loops, While Loops
Breaking Out Of Loops
Skipping A Loop Cycle

DOM Manipulation & Events

Project Setup and Details
First DOM Access and Manipulation
Events and Event Handling: Rolling the Dice
Updating Scores and Changing the Active Player
Implementing Our 'Hold' Function
The DRY Principle
Creating a Game Initialization Function
Finishing Touches: State Variables

Modern JavaScript 1

Using ES6 NPM Babel Webpack
Installing Node.js & NPM
Note: Tooling Changes
Modern Setup: Configuring Webpack
Webpack Dev Server
Setup: Babel
Planning Project Architecture with MVC
ES6 Modules,Introduction, setting up, running ES6 code, JS transpilers
Template literals, default parameters
Destructuring arrays & objects, Rest & spread

Modern JavaScript 2

Project Setup and Details
Building the Search Model, Controller, Search View
Implementing Persistent Data with localStorage
Building the Recipe Model, Controller, View
Building the Search Model, Controller, Search View
Rendering an AJAX Loading Spinner
Try Catch And Throw
Date & Math & Regular Expressions
Primitives vs. Objects
Navigator & Creating Objects: Object.create
setInterval And setTimeout
Inheritance and the Prototype Chain
Creating Objects: Function Constructors
Closures, Bind, Call and Apply

Asynchronous JavaScript

Promises, Async/Await and AJAX
Understanding Asynchronous JavaScript: The Event Loop
The Old Way: Asynchronous JavaScript with Callbacks
From Callback Hell to Promises
From Promises to Async/Await
AJAX and APIs
Making AJAX Calls with Fetch and Promises
Making AJAX Calls with Fetch and Async/Await
What Is Ajax
How To Execute Ajax
Requests & Response
readyStates And Status
Putting It Together
Ajax Search Suggest Tutorial
Ajax Database Select Menu Tutorial

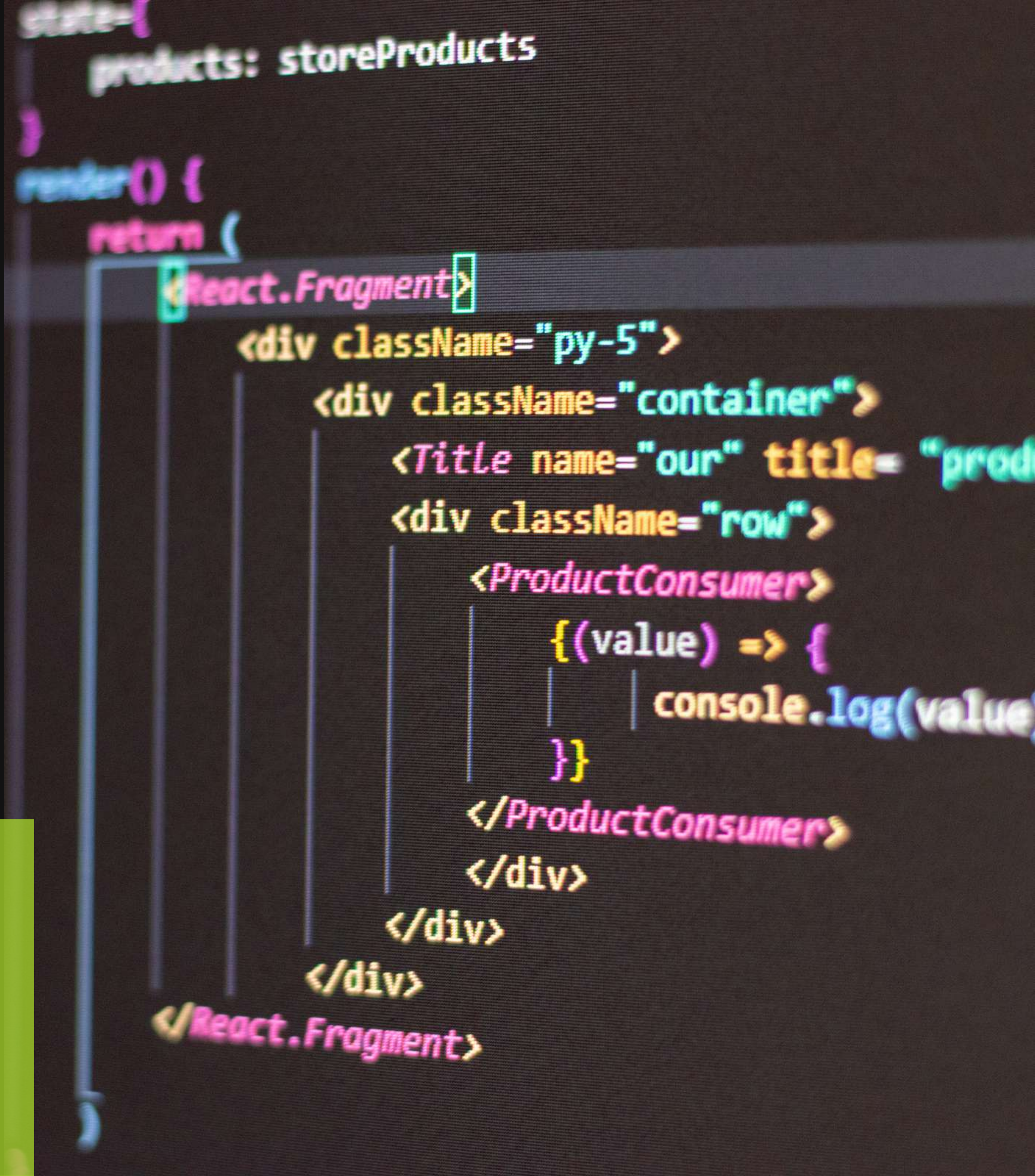
Fundamentals of jQuery

What Is jQuery & Installing jQuery
Syntax Selectors Events
When A Key Is Pressed Effects & Callbacks
.html() & Changing CSS
Animations & Attributes
jQuery.Ajax & Finishing jQuery
The Prototype Chain in the Console
First Class Functions: Passing Functions as Arguments
First Class Functions: Functions Returning Functions
Immediately Invoked Function Expressions (IIFE)

Angular Developer

Angular vs Angular 2 vs Angular 8
CLI Deep Dive & Troubleshooting
Project Setup and First App Editing
What is TypeScript?
A Basic Project Setup using Bootstrap for Styling
Debugging
Components & Databinding Deep Dive
Live Course Project Components & Databinding
Directives Deep Dive
Live Course Project Directives
Using Services & Dependency Injection
Live Course Project Services & Dependency Injection
Changing Pages with Routing
Live Course Project Routing
Understanding Observables
Live Course Project Observables

Handling Forms in Angular Apps
Live Course Project Forms
Using Pipes to Transform Output
Making Http Requests
Live Course Project Http
Authentication & Route Protection in Angular
Dynamic Components
Angular Modules & Optimizing Angular Apps
Deploying an Angular App
Working with NgRx in our Project*
Angular Universal*
Angular Animations
Adding Offline Capabilities with Service Workers
A Basic Introduction to Unit Testing in Angular Apps
Angular Changes & New Features
Custom Project & Workflow Setup
TypeScript for Angular 2 Usage



```
state={
  products: storeProducts
}
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title= "prod
          <div className="row">
            <ProductConsumer>
              {(value) => {
                console.log(value
              }}
            </ProductConsumer>
          </div>
        </div>
      </div>
    </React.Fragment>
}
```

React.js Module 1

Introduction to React.js and the React ecosystem
Setting up a React development environment
Creating and rendering React components
JSX syntax and its usage in React
Managing component state
Handling events in React
Conditional rendering and rendering lists
Styling React components
Component composition and reusability
Props and prop types in React
React forms and form handling
React Router for client-side routing
React lifecycle methods
Fetching data from APIs in React
Handling asynchronous operations with Promises and async/await
Error handling in React

React.js Module 2

State management in React using Redux
Redux middleware and asynchronous actions
Working with React hooks
Context API for state management
React performance optimization techniques
React and CSS-in-JS libraries
Unit testing React components
Deployment strategies for React applications
Building responsive and mobile-friendly UIs with React
Animations and transitions in React
Internationalization and localization in React
Accessibility best practices in React
Server-side rendering with React
Building progressive web applications (PWAs) with React
Integration with backend frameworks (e.g., Node.js, Express)

MongoDB Developer Module 1

NoSQL databases & Key value pair
Understanding the Basics & CRUD Operations
Schemas & Relations: How to Structure Documents
Exploring Shell & Server
Using MongoDB Compass Explore Data Visually
MongoDB with Node and Express
Diving Into Create Operations
Read Operations - A Closer Look
Windows Setup

MongoDB Developer Module 2

Update Operations
Understanding Delete Operations
Working with Indexes
Working with Geospatial Data
Understanding the Aggregation Framework
Working with Numeric Data
MongoDB & Security
From Shell to Driver
Introducing Stitch
Roundup

```
require File.expand_path(
# Prevent database
abort("The Rails environment
require 'spec_helper'
require 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver =
Category.delete_all; Category
Shoulda::Matchers.configure do
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end

# Add additional requires

# Requires supporting ruby
# spec/support/ and its
# run as spec files by default.
# in _spec.rb will both be
# run twice. It is
# end with _spec.rb.
```

No results found for 'mongoid'

Express

Installing and Exploring Node.js
Node.js Module System
File System and Command Line Args
Improved Development Workflow and Debugging
Asynchronous Node.js
Accessing API from Browser
API Authentication and Security
Real-Time Web Applications with Socket.io
Improved Development Workflow and Debugging
Working with Express.js
Express JS with MongoDB and Sqlite
Working with Dynamic Content & Adding Templating Engines
The Model View Controller (MVC)
Dynamic Routes & Advanced Models
SQL Introduction
Understanding Sequelize
Working with NoSQL & Using MongoDB

Installing and Exploring Node.js
Node.js Module System
File System and Command Line Args
Improved Development Workflow and Debugging
Asynchronous Node.js
Accessing API from Browser
API Authentication and Security
Real-Time Web Applications with Socket.io
Improved Development Workflow and Debugging
Working with Express.js
Express JS with MongoDB and Sqlite
Working with Dynamic Content & Adding Templating Engines
The Model View Controller (MVC)
Dynamic Routes & Advanced Models
Understanding Sequelize
Working with NoSQL & Using MongoDB

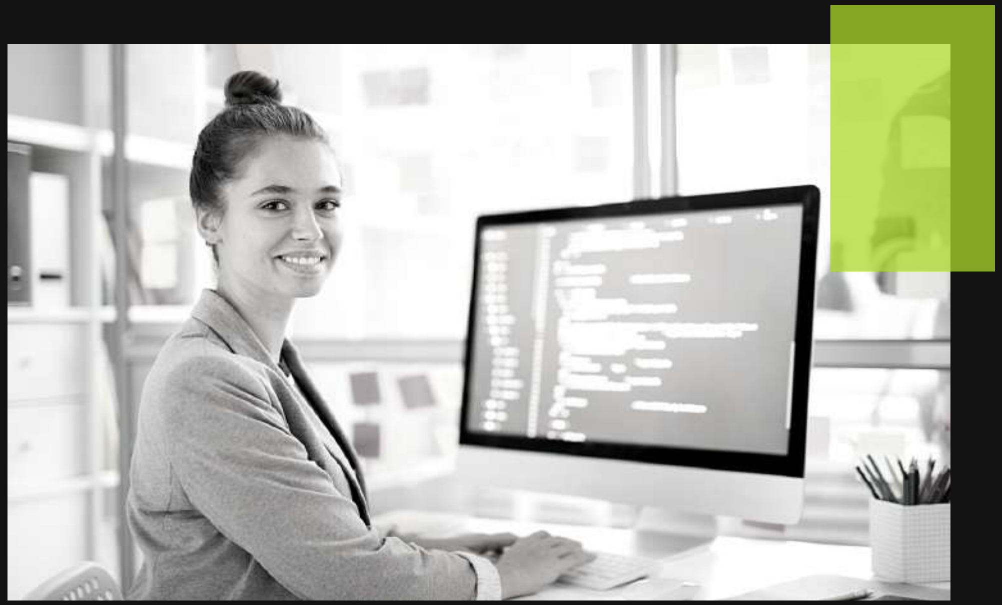
```
var iiObject = Class.create({
    initialize: function(pos, y){
        if (typeof(y) == 'undefined')
            this.pos = pos || new iiVec2();
        else this.pos = new iiVec2(pos,y);
        this.drawCenter = true;
    },
    setPos: function(v,y){
        if(typeof(y) == 'undefined')
            this.pos = v;
        else {
            this.pos.x = v;
            this.pos.y = y;
        }
    },
    move: function(v,y){
        this.pos.add(v,y);
    },
    draw: function(ctx){
        if (this.drawCenter){
            //draw middle dot
            ctx.fillStyle = '#fff';
            ctx.fillRect(this.pos.x-2, this.pos.y-2, 4, 4);
        }
    }
});

var iiShape = Class.create(iiObject, {
    initialize: function($super, pos, y, vel, velY){
        $super(pos, y);
        if (typeof(velY) == 'undefined')
            this.vel = vel || new iiVec2();
        else this.vel = new iiVec2(vel,velY);
    },
    update: function($super, dt){
        if (typeof(this.drawables) != 'undefined')
            for (var i=0;i<this.drawables.length;i++)
                if (typeof(this.drawables[i].update) != 'undefined')
                    if(!this.drawables[i].update())
                        return false;
        this.move(new iiVec2(this.vel.x*dt, this.vel.y*dt));
        if (typeof(this.torque) != 'undefined') this.rotation+=this.torque;
        if (this.bounds != null && ((this.bounds.top != null && this.pos.y <
            this.bounds.top) || (this.bounds.right != null && this.pos.x > this
            .bounds.right) || (this.bounds.bottom != null && this.pos.y > this.
            bounds.bottom) || (this.bounds.left != null && this.pos.x < this.
            bounds.left)))
            return false;
        return true;
    },
```

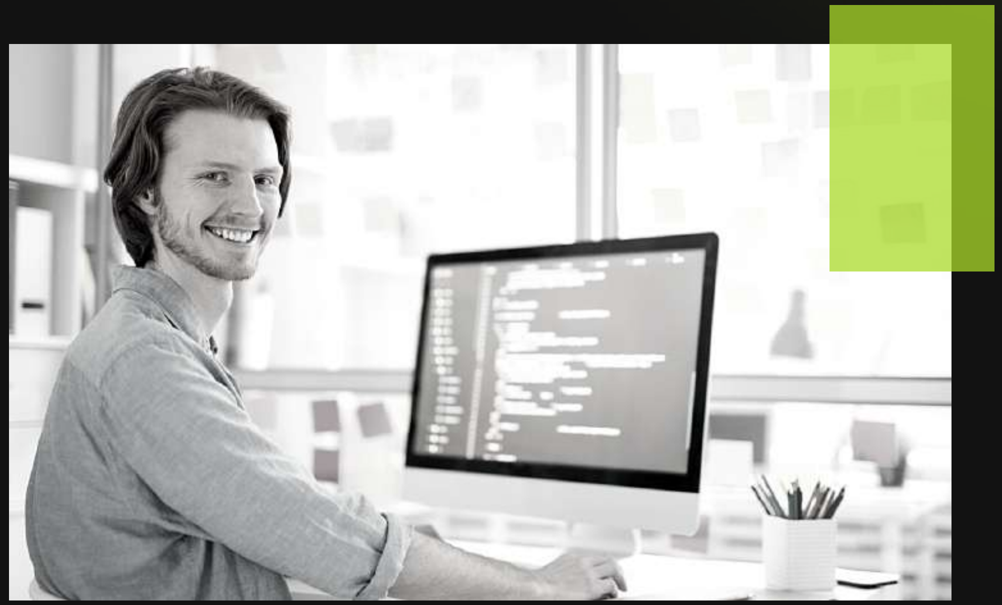
Node.js

Installing and Exploring Node.js
Node.js Module System
File System and Command Line Args
Improved Development Workflow and Debugging
Asynchronous Node.js
Accessing API from Browser
API Authentication and Security
Real-Time Web Applications with Socket.io
Improved Development Workflow and Debugging
Working with Express.js
Express JS with MongoDB and Sqlite
Working with Dynamic Content & Adding Templating Engines
The Model View Controller (MVC)
Dynamic Routes & Advanced Models
SQL Introduction
Understanding Sequelize
Working with NoSQL & Using MongoDB

Working with Mongoose
Sessions & Cookies
Adding & Advanced Authentication
Understanding Validation
Sending Emails & Error Handling
File Upload & Download
Sorting, Pagination, and Filtering
Understanding Async Requests
Adding Payments
Working with REST APIs - The Practical Application
Understanding Async Await in Node.js
Understanding Websockets & Socket.io
Working with GraphQL
Testing Node.js Applications
Node.js as a Build Tool & Using npm

BECOME A TRUE PROGRAMMER





THANK YOU