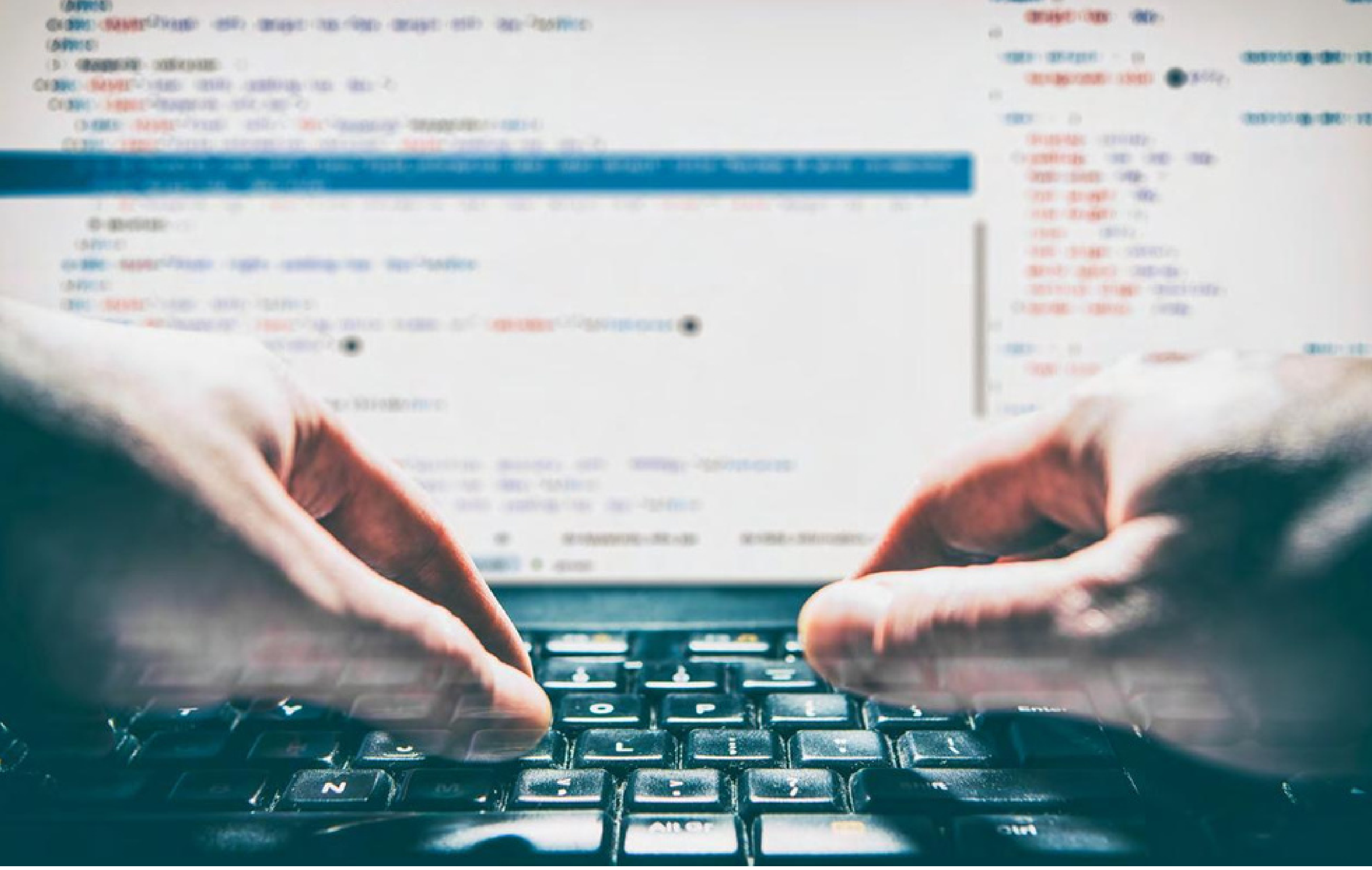
NEXUS ACADEMY

Full Stack MEAN Developer

Master's Program

Table of Contents



About the Course

This Full Stack MEAN Developer program provides complete knowledge of software development and testing technologies such as JavaScript, Node.js, Angular, Docker, and Protractor. You'll build an end-to-end application, test and deploy code, and store data using MongoDB.

```
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y"
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z"
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
ier_ob.select=1
ntext.scene.objects.activ
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_obj
ata.objects[one.name].se

int("please select exactl
```

About Software Development Industry

The global software engineering market is expected to grow to approximately USD $49.19 billion by 2028 at a CAGR of 7.15 percent in the forecast period of 2021 to 2028. Corporate initiatives focused on digital transformation will help drive demand for skilled software developers, which is reflected in current hiring trends and future projections.

Many companies prefer to hire multi-skilled technology professionals such as automation test engineers. The average annual salary for an automation test engineer is USD $94,270 (ZipRecruiter).
Additional facts about the state of the software development industry:

- Junior developers are getting massive starting salaries compared to those of the last 20 years
- 83% of working professionals plan to upskill in 2023 (Business Today 2023)
- The phenomenal rise of consumer applications in both web and mobile is driven by the availability of open-source projects and libraries
- Smaller, quicker releases—which results in better productivity—are becoming crucial for software product success. Automation engineers are well-positioned to empower this trend.

Key Features

Comprehensive Applied Learning program

Choose from 3 industry-aligned capstone projects

Job-assist program included (India only)

160 hours of instructor-led training

20+ in-demand tools and skills

Build capstone projects in four domains and showcase your projects to recruiters

4 phase-end projects

Top Skills and Tools covered

✓ Agile	✓ Angular	✓ Docker
✓ Git	✓ Node.js	✓ Jenkins
✓ HTML	✓ HTTP	✓ Grunt
✓ CSS	✓ REST API	✓ AWS
✓ JavaScript	✓ Express.js	✓ GitHub
✓ BootStrap	✓ MongoDB	✓ API testing with Postman
✓ TypeScript	✓ AJAX	

Program Outcomes

By the end of the program, you will be able to accomplish the following:

- ✓ Cultivate an Agile mindset with the ability to quickly complete industry projects
- ✓ Acquire knowledge and skills to build dynamic end-to-end web applications
- ✓ Master software programming concepts—the basic building blocks of designing great apps
- ✓ Architect scalable back-end infrastructure
- ✓ Learn how to work with a front-end framework to create a front-end website
- ✓ Deploy static and dynamic websites in the cloud
- ✓ Prepare to kickstart your tech career in top IT companies

Who Should Enroll in this Program?

This MEAN stack program is designed for:

Freshers, software developers, IT professionals, engineers, test engineers, technical consultants, and analysts

Learning Path

1 Building a strong MEAN foundation

2 Develop rich frontend with Angular

3 Complete MEAN backend with NodeJS and MongoDB

4 Integrate and Deploy your MEAN app on Cloud

5 Full Stack MEAN Developer Capstone Project

Master's Certificate

STEP **1** 2 3 4 5

Building a strong MEAN foundation

Planning projects with Agile

- ✓ Introduction to Agile
- ✓ Scrum Roles
- ✓ Scrum Practices
- ✓ Scrum Estimation
- ✓ Scrum Planning
- ✓ Q&A with Knowledge Checks (5)

Version Control System

- ✓ Introduction
- ✓ Working with Git
- ✓ GitHub as a SCM tool
- ✓ Create and Clone a GitHub Repository
- ✓ Fork, Push, and Pull in Git
- ✓ Create a Pull Request in Git
- ✓ Push file to GitHub Repository
- ✓ Branching in Git
- ✓ Create a Branch in Git
- ✓ Switching Branches in Git
- ✓ Switching Branches in Git
- ✓ Merging Branches in Git
- ✓ Merging Branches in Git
- ✓ Check the status of a file
- ✓ Q&A with Knowledge Checks (5)
- ✓ Lesson-End Project

HTML/CSS

- ✓ Introduction
- ✓ Links, lists, and images
- ✓ Tables and forms
- ✓ ID, classes, header, and footer
- ✓ CSS foundation
- ✓ Selectors
- ✓ Text, colours, and forms styling
- ✓ Links and positioning

Javascript

- ✓ Javascript Introduction
- ✓ JavaScript basics
- ✓ Form Validation

- Primitives and Objects
- Primitives and Objects
- Functions and Prototyping
- Working with functions
- Functions and Prototyping
- IIFEs, callbacks, and closures
- IIFEs, callbacks, and closures-Assisted practice
- IIFEs and functions
- IIFEs and functions
- Maps and Classes
- Maps and Classes-Assisted practice
- Promises and Async
- Promises and Async- Assisted practice
- Ajax Calls
- Ajax Calls-Assisted practice
- Webpack and Modern JavaScript
- Webpack and Modern JavaScript-Assisted practice
- Babel
- Working with Babel
- Canvas
- Q&A with Knowledge Checks (5)
- Section-end project

Bootstrap

- Starter Template Guide
- Import Bootstrap into your application: Assisted Practice
- Browsers and Devices Compatibility
- Bootstrap: Themes
- Bootstrap: Alerts
- Implement Alerts: Assisted Practice: Assisted Practice
- Bootstrap: Cards, Tables, and Lists
- Develop a static webpage
- Bootstrap: Accordion and Carousel
- Implement a Slideshow to a Static Webpage
- Bootstrap: Forms and Form Validations
- Forms: Assisted Practice
- Bootstrap: Modal Components
- Implement a Modal to a Static Web Page: Assisted Practice
- Bootstrap: Nav and Navbar
- Implement Navbars to a Static Web Page: Assisted Practice
- Bootstrap: Pagination and Progress
- Pagination and Progress: Assisted Practice

STEP 1 **2** 3 4 5

Develop rich frontend with Angular

JSON server

- ✓ Introduction
- ✓ JSON Server
- ✓ Create a JSON server File
- ✓ HTML elements
- ✓ Create HTML Elements for JSON Server
- ✓ Set up a JSON DB
- ✓ Performing GET, POST, PUT, and DELETE in JSON DB
- ✓ Fetching Data

Working with Angular Application

- ✓ Introduction to Angular
- ✓ Angular Components
- ✓ Creating an Angular Project: Assisted Practice
- ✓ Data Binding and Event Handling
- ✓ Data binding: Assisted Practice
- ✓ Event Handling : Assisted Practice
- ✓ Scope
- ✓ Root Scope Progam: Assisted Practice
- ✓ Animations
- ✓ Create an Angular Application using Animations: Assisted Practice
- ✓ Angular Expressions
- ✓ Working with Angular Expressions: Assisted Practice
- ✓ Directives
- ✓ Working with Directives: Assisted Practice
- ✓ Pipes
- ✓ Working with Pipes: Assisted Practice
- ✓ Nesting Components
- ✓ Nesting Components
- ✓ Component Level interactions: Assisted Practice
- ✓ Component Level interactions
- ✓ Forms
- ✓ Validations
- ✓ Reactive forms in Angular: Assisted Practice

- Angular Route
- Navigation
- Angular Route and Navigation
- Services and Injectables: Assisted Practice
- Services and dependency
- Services and Injectables
- Routing Mechanisms
- Routing Mechanisms: Assisted Practice
- Authentication with JWT and Security
- REST API Calls
- Deploying an Angular application on server

Jasmine

- Introduction
- Installation and testing
- Testing a Basic Application
- Testing an application
- Testing Source code and using
- Various Functions
- Jasmine Features
- Use Matchers, Spies, Checks in Jasmine
- Jasmine Hooks
- A Simple Hook Demo
- Asynchronous specs
- Debugging Jasmine Tests

Create your first Progressive Web App with Angular

- Set up PWA
- Set up PWA: Assisted Practice
- PWA - Deep dive
- Application to Home Screen
- Creating a Responsive App Design
- Design a Responsive App
- Angular Application Shell
- Notification Management
- Notification Management: Assisted Practice
- Service Worker
- Service Worker: Assisted Practice
- Dynamic Data
- Working with Offline Page

STEP 1 2 **3** 4 5

Complete MEAN backend with NodeJS and MongoDB

NodeJS

- Introduction
- Installing and creating an app: Assisted Practice
- Node server
- Creating a Node Server: Assisted Practice
- Requests and Responses
- Working with Requests and Responses: Assisted Practice
- Asynchronous Node.js- part 1
- Blocking and Non-Blocking in Node.js: Assisted Practice
- Asynchronous Node.js- part 2
- Event loop: Assisted Practice
- Modules
- By Reference and By Value: Assisted Practice
- Create and test a module: Assisted Practice
- Export and Require
- Events and Event Emitter
- Working with Events: Assisted Practice
- Functions and arrays
- Node.js: HTTP
- HTTP request and callback : Assisted Practice
- HTTP Paser: Assisted Practice
- TCP/IP and routing
- APIs in nodeJS: Assisted Practice
- Node.js: App deployment
- Node.js: App deployment: Assisted Practice
- Node Package Manager
- Install Node Package Manager
- File System: Assisted Practice
- File System commands
- Events
- Perform various events in Node js: Assisted Practice

- Debugging Node JS Application: Assisted Practice
- Debugging with Visual Studio
- Database connectivity
- Database connectivity commands: Assisted Practice
- Template Engines
- Multiprocessing in Node.js
- Working with Child API: Assisted Practice

ExpressJS

- Introduction
- Installation of ExpressJS along with Hello world program
- Working with Express.js
- Installation of Express js: Assisted Practice
- ExpressJS frameworks
- Working with Express.js frameworks: Assisted Practice
- Configuration
- Configuration commands: Assisted Practice
- Types of Middleware
- Middleware in Express js app: Assisted Practice
- Request Handlers
- Working with Request Handlers: Assisted Practice
- Response methods
- Working with Response methods: Assisted Practice
- Error Handling
- Error Handling commands: Assisted Practice
- CRUD operations
- CRUD operations: Assisted Practice

SQL

- Introduction to SQL
- Demo: Creating Databases and Tables: Assisted Practice
- Filtering, Consolidating, and grouping data
- Logical Operators: Assisted Practice
- SQL Commands
- Demo: Insert, Update, and Delete Records from Table: Assisted Practice
- Demo: Using Select Statement with various clauses: Assisted Practice
- Demo: Working with SubQueries: Assisted Practice
- Lesson3: SQL Joins
- Demo: Joins on Tables: Assisted Practice
- Demo: Working with Related tables: Assisted Practice

- SQL Scripts
- Triggers and cursors
- Creating Triggers and cursors
- Built-in Functions- part 1
- Built-in Functions- part 2
- Working with various built in SQL functions: Assisted Practice

MongoDB

- Introduction to MongoDB
- JSON and BSON structure: Assisted Practice
- Databases, Collections & Documents
- Creating Databases & Collections: Assisted Practice
- Accessing Structured Data with arrays: Assisted Practice
- Introduction to NoSQL database
- NoSQL commands: Assisted Practice
- MongoDB as Document Database
- MongoDB as a document database: Assisted Practice
- Document Store: Example
- Schemas part-1
- Schemas part-2
- Schema guidelines: Assisted Practice
- Shell and server in MongoDB
- Exploring the shell and the server: Assisted Practice
- Scaling and Replicating
- Managing memory
- Memory Management: Assisted Practice
- Relationships in MongoDB
- Relationships in MongoDB: Assisted Practice
- Create Operation
- Create and insert: Assisted Practice
- Read Operation
- Read Operation commands: Assisted Practice
- Update Operation part-1
- Update Operation part-2: Assisted Practice
- Update operation commands
- Delete Operation
- Delete Operation commands: Assisted Practice
- Operations in MongoDB
- Operations in MongoDB: Assisted Practice

STEP 1 2 3 **4** 5

Integrate and Deploy your MEAN app on Cloud

Mocha

- Introduction to Mocha
- Hello world unit test with Mocha: Assisted Practice
- Mocha Overview
- Testing a Promise: Assisted Practice
- Detect multiple calls to done(): Assisted Practice
- Assertion Librabries: Assisted Practice
- Asynchronous and Synchronous Code: Assisted Practice
- Arrow Functions: Assisted Practice
- Hooks: Assisted Practice
- Dynamically generating tests: Assisted Practice
- Timeouts: Assisted Practice
- Command-line Usage
- Plugins, Fixtures, Interfaces, and Reporters in Mocha
- BDD, TDD, Exports, QUnit and Require-style interfaces: Assisted Practice
- Reporters: Assisted Practice
- Configuring Mocha
- Testing Mocha and test duration: Assisted Practice
- Running Mocha in Browser: Assisted Practice

Docker

- Docker Overview
- Docker Image: Build a customer docker image: Assisted Practice
- Introduction to Docker Hub
- Push an image to Docker Hub: Assisted Practice
- Introduction to Docker Swarm
- Container a Deployment Using Docker Swarm: Assisted Practice
- Container Scaling with Docker Swarm: Assisted Practice
- Distribute your app across a cluster: Assisted Practice

- Deploy your app to production: Assisted Practice
- Setting up Jenkins pipeline with Docker: Assisted Practice
- Introduction to Docker Compose
- Docker Compose: Assisted Practice
- Docker CE
- Docker CE: Assisted Practice

Jenkins

- Introduction to Jenkins
- Jenkins Overview
- Creating the First Jenkins Job: Assisted Practice
- Jenkins Plugins
- Create and deploy a Maven project on Tomcat: Assisted Practice
- Build Jobs and Jenkins Security
- Creating a basic Jenkins pipeline: Assisted Practice
- Creating a basic build job to clone a git and display the contents: Assisted Practice
- Jenkins Metrics to Improve Quality
- Managing and Monitoring Jenkins
- Managing Plugins on Jenkins: Assisted Practice
- Managing Users on Jenkins: Assisted Practice
- Automated and Continuous Deployment
- Deployment of an App as per Jenkins project: Assisted Practice
- Deployment of an app with Jenkins pipeline: Assisted Practice
- Jenkins Integration with GitHub: Assisted Practice
- Jenkins Pipeline
- Pipeline code project: Assisted Practice
- Enable Jenkins Pipeline Plugin: Assisted Practice

AWS

- Cloud Introduction
- Sign up for AWS: Assisted Practice
- EC2 Introduction
- Instance Types of EC2
- EC2 Pricing
- Launch and connect to an EC2 Linux instance: Assisted Practice
- Change the volume size of the instance: Assisted Practice
- Placement Groups
- Launch instances in a placement group
- Security Groups
- VPC

- ✓ EBS
- ✓ Create an EBS volume: Assisted Practice
- ✓ Attaching an Amazon EBS Volume to an Instance
- ✓ Format and Mount an EBS volume : Assisted Practice
- ✓ EBS snapshots
- ✓ Create EBS snapshot
- ✓ Initialize a volume restored from a snapshot on Linux: Assisted Practice
- ✓ ELB
- ✓ Auto Scaling
- ✓ Create and deploy a load balancer: Assisted Practice
- ✓ Auto Scaling with launch templates: Assisted Practice
- ✓ Amazon S3 introduction
- ✓ Amazon S3 bucket
- ✓ Create a bucket: Assisted Practice
- ✓ Bucket Restrictions and Limitations
- ✓ Delete an S3 bucket
- ✓ Empty an S3 Bucket
- ✓ S3 objects
- ✓ Object key and metadata
- ✓ Storage classes
- ✓ Set the storage class of an object
- ✓ Operations on Objects
- ✓
- ✓
- ✓ EFS
- ✓ IAM
- ✓ Creating an IAM user: Assisted Practice
- ✓ Creating an IAM role: Assisted Practice
- ✓ Creating an IAM group: Assisted Practice
- ✓ Understanding policies and permissions: Assisted Practice
- ✓ Web hosting: Assisted Practice
- ✓ Deploying your application: Assisted Practice
- ✓ Session end project

STEP 1 2 3 4 **5**

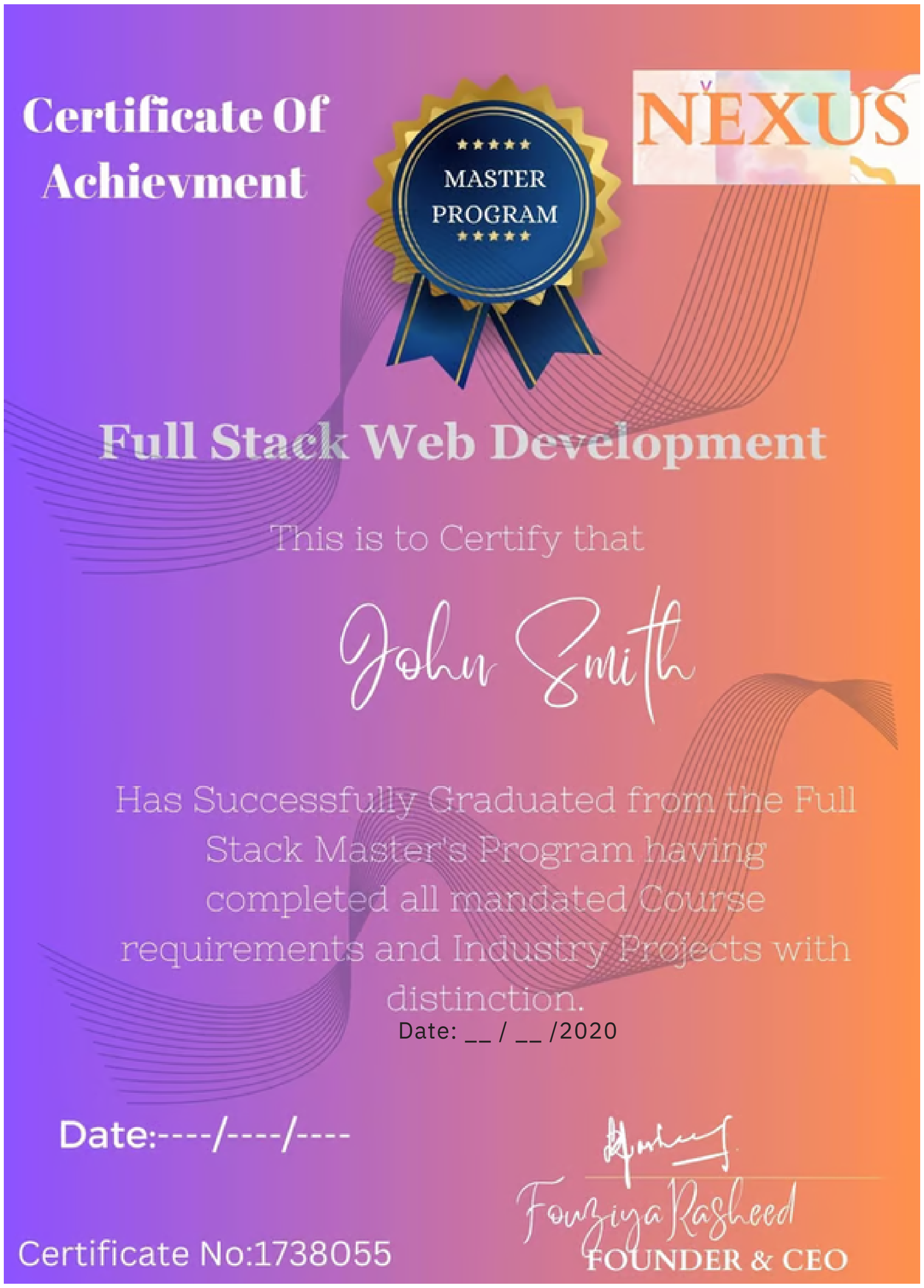
Full Stack MEAN Developer Capstone Project

E- commerce	Create a dynamic and responsive Java e-commerce web application using technologies such as Angular, React, MongoDB, and Docker.
Food Delivery	Build a food-delivery app from scratch using your knowledge of the Spring framework, web services, and MongoDB with a strong back end to support operations.
Entertainment	Create an entertainment application like BookMyShow using your back-end knowledge, API development, and HTML and CSS skills.
Healthcare	Build a dynamic and effective healthcare app. Create a rich UI for effective diagnosis and health advice with Angular, Javascript, TypeScript, ES6, and many more.

Certificates

CERTIFICATE
OF ACHIEVEMENT

MEAN STACK



Certificate Of Achievment

MASTER PROGRAM
NEXUS

Full Stack Web Development

This is to Certify that

John Smith

Has Successfully Graduated from the Full Stack Master's Program having completed all mandated Course requirements and Industry Projects with distinction.

Date: __ / __ /2020

Date:----/----/----

[illegible]

Fouziya Rasheed
FOUNDER & CEO

Certificate No:1738055

Classroom-Level Immersion: Delivered Digitally

Anywhere Anytime Access

Online Enrollment and Access on Web and Mobile

Online Self-Learning

Learner Watches the Video

Chapter-End Quizzes

Live, Interactive Classes

Live Virtual Classroom

Live Interaction and Mentoring

Hands-On Experience

Final Assessment

Project Work

Virtual Labs

Simplilearn Certification Criteria

80% Course Completion

+

Project Submitted and Accepted

Internal, and External Certification

Final Exam and Certification

PeopleCert All talents, certified.

Project Management Institute.

Scrum Alliance

EXIN

Microsoft

AXELOS GLOBAL BEST PRACTICE

THE Open GROUP

Corporate Training

Top clients we work with:

VISA	salesforce	P&G	amazon	Schneider Electric	Duke
mastercard	Microsoft	pepsi	DELL	Capgemini	TATA CONSULTANCY SERVICES
HSBC	CISCO	Mondelēz International	vmware	accenture	THE WORLD BANK
BARCLAYS	IBM	Mercedes-Benz	GE	BOEING	orange
UBS	hp	Ford	pwc	Pfizer	WPP

Features of Corporate Training:

- Tailored learning solutions
- Flexible pricing options
- Enterprise-grade learning management system (LMS)
- Enterprise dashboards for individuals and teams
- 24X7 learner assistance and support