Job Assured Python Fullstack

Front End	HTML |Tailwind CSS | JavaScript | React Js | Node Js | Responsive (Bootstrap)
Language	Core Python | Advance Python Libraries
Framework	Django | Rest API
Database	SQL | MongoDB (NoSQL) | GIT

Post Training	Project Building | Mock Interview | CV building | Placement Preparation

HTML
Tailwind CSS

Module 1: Tailwind CSS Overview

- What is Tailwind?
- Tailwind CSS Vs CSS
- Installing Tailwind

Module 2: Tailwind Basics

- Utilities
- Preflight
- Duplication
- Prefixes
- CSS Units

Module 3: Typography

- Size and Shape
- Color and Opacity
- Alignment and Spacing
- Lists
- Typography Plugin
- Tailwind Forms

Module 4: Box

- Padding and Margins
- Borders
- Background Color
- Background Images
- Height and Width

Module 5: Page Layout

- Containers
- Floats and Clears
- Position and Z-Index
- Tables

- Grids
- Flexbox
- Box Alignment

Module 6: Animations

- Helpful Small Animations
- Transitions
- Transformation
- Other Appearance Things

Module 7: Responsive Design

- Tailwind Screen Width and Breakpoints
- Hide Base on Size
- Fewer Grid Columns on Small Devices
- Flex on Larger Devices

Module 8: Customizing Tailwind

- Configuration File Basics
- Change Default Values
- Change Generated Classes
- Configure Variant Prefixes
- Integrate with Existing CSS
- Access Tailwind from JavaScript
- Purge CSS

JavaScript Training Syllabus

Introduction

- What is JavaScript?

Developer Essentials

- The development workflow
- Selecting the right tools for the job
- Just enough HTML and CSS Understanding objects
- Understanding variables
- Making comparisons
- Understanding events

Starting to Code

- Writing your first script
- Internal vs external scripts
- Using comments in scripts
- Using the noscript tag in HTML

Interacting with Users

- Creating alert dialogs
- Understanding conditional statements
- Getting confirmations from users
- Creating prompts for users
- Understanding functions
- Making links smarter
- Using switch/case statements

- Handling errors

JavaScript Language Essentials

- Getting started
- Creating loops
- Passing values to functions
- Detecting objects
- Reading arrays
- Returning values from functions
- Writing arrays
- Building do and while loops
- Re-using functions

Creating Rollovers and More

- Creating a basic image rollover 🎬 How to write a better rollover
- Creating a three-state rollover
- Making rollovers accessible and 508 compliant
- Making disjointed rollovers
- Creating slideshows
- Displaying random images

Building Smarter Forms

- Getting started
- Creating jump menus
- Creating dynamic menus
- Requiring fields
- Cross-checking fields
- Displaying more informative errors
- Verifying radio button selections
- Setting one field with another field
- Verifying email addresses

Handling Events

- Responding to window events
- Responding to mouse movements
- Responding to mouse clicks
- Responding to onBlur form events
- Responding to onFocus form events
- Responding to keyboard events

Working with Cookies

- Demystifying cookies
- Writing a cookie
- Reading a cookie
- Displaying a cookie
- Counting with cookies
- Deleting cookies
- Handling multiple cookies
- Cookies in action

The DOM, Nodes, and Objects

- Understanding the DOM
- Adding nodes to the DOM
- Deleting nodes from the DOM
- Deleting specific nodes
- Inserting nodes into the DOM
- Replacing nodes in the DOM

Working with Dates and Times 🎬

- Displaying dates

- Displaying times
- Creating a countdown

Real World Applications of JavaScript

- Creating sliding menus
- Creating pop-up menus
- Creating slideshows with captions
- Creating

React JS

Module 1: ES6 Primer

- History of Javascript
- What is ES6
- A word on bable
- Block scope, let &const
- Template literals
- Arrow functions
- Spread and Rest operators
- Object literal improvements
- Destructuring
- Classes
- Inheritance <
- Static properties and methods
- Promises
- Iterators and Iterables
- Generators
- Modules
- Set and Map

Module 2: Introduction to React

- What is React?
- Why React?
- React version history
- React 16 vs React 15
- Just React - Hello World
- Using create-react-app
- Anatomy of react project
- Running the app
- Debugging first react app

Module 3: Templating using JSX

- Working with React.create Element
- Expressions
- Using logical operators
- Specifying attributes
- Specifying children

Module 4: It's all about components

- Significance of component architecture
- Types of components
- Functional
- Class based
- Pure
- Component Composition

Module 5: Working with state and props

- What is state and it significance
- Read state and set state

- Passing data to component using props
- Validating props using propTypes
- Supplying default values to props using defaultProps

Module 6: Rendering lists

- Using React key prop
- Using map function to iterate on arrays to generate elements

Module 7: Event handling in React

- Understanding React event system
- Understanding Synthetic event
- Passing arguments to event handlers

Module 8: Understand component lifecycle

- Understand component lifecycle

Module 9: Working with forms

- Controlled components
- Uncontrolled components
- Understand the significance to defaultValue prop
- Using react ref prop to get access to DOM element

Module 10: Routing with react router

- Setting up react router
- Understand routing in single page applications
- Working with BrowserRouter and HashRouter components
- Configuring route with Route component
- Using Switch component to define routing rules
- Making routes dynamic using route params
- Working with nested routes
- Navigating to pages using Link and NavLink component
- Redirect routes using RedirectComponent
- Using Prompt component to get consent of user for navigation
- Path less Route to handle failed matches

Module 11: Just Redux

- What is redux
- Why redux
- Redux principles
- Install and setup redux
- Creating actions, reducer and store

Module 12: Immutable.js for immutable datastructures

- What is Immutable.js?
- Immutable collections
- Lists
- Maps
- Sets

Module 13: React Redux

- What is React Redux
- Why React Redux
- Install and setup
- Presentational vs Container components
- Understand high order component
- Understanding mapStateToProps and mapDispatchtToProps usage

Module 14: ReduxSaga(Redux middleware)

- Why redux middleware
- Available redux middleware choices
- What is redux saga
- Install and setup redux saga

- Working with Saga helpers
- Sagas vs promises

Module 15: Unit testing in react

- Understand the significance of unit testing
- Understand unit testing jargon and tools
- Unit testing react components with Jest
- Presentational vs Container components
- Unit testing react components with enzyme

Module 16: New Features in React JS

- Understand error boundaries (new in React 16
- Understand react portals (new in React 16)

Module 17: Webpack primer

- What is webpack
- Why webpack
- Install and setup webpack
- Working with webpack configuration file
- Working with loaders
- Quick word on code splitting, lazy loading, tree shaking
- Setting up Hot Module Replacement

Module 18: Server-side rendering with React

- What is server-side rendering (SSR)?
- Why SSR
- Working with renderToString and renderToStaticMarkup methods

MongoDB

Introduction to NoSQL databases

- Objectives
- What is NoSQL?
- Why NoSQL?
- Difference Between RDBMS and NoSQL Databases
- Benefits of NoSQL
- Types of NoSQL
- Key-Value Database
- Document Database
- Column-Based Database
- Graph Database
- Consistency
- Availability

MongoDB - A Database for the Modern Web

- What is MongoDB?
- JSON, JSON Structure
- BSON
- MongoDB Structure
- Document Store Example
- MongoDB as a Document Database
- Transaction Management in MongoDB
- Easy Scaling
- Scaling Up vs. Scaling Out
- Vertical Scaling
- Horizontal Scaling
- Features of MongoDB
- Secondary Indexes
- Replication

- Memory Management
- Replica Set
- Auto Sharding
- Aggregation and MapReduce
- Collection and Database
- Schema Design and Modeling
- Reference Data Model
- Data Types
- Core Servers of MongoDB
- MongoDB's Tools
- Installing and Starting MongoDB on Linux
- Installing and Starting MongoDB on Windows

CRUD Operations in MongoDB

- Data Modification in MongoDB 🎬 Batch Insert in MongoDB
- Ordered Bulk Insert
- Performing Ordered Bulk Insert
- Unordered Bulk Insert
- Performing Un-ordered Bulk Insert
- Inserts: Internals and Implications
- Performing an Insert Operation
- Retrieving the documents
- Specify Equality Condition
- Retrieving Documents by Find Query
- $in, $or , and “AND” Conditions
- $or Operator
- Specify AND/OR Conditions
- Retrieving Documents by Using FindOne, AND/OR Conditions
- Regular Expression
- Array Exact Match
- Retrieving Documents for Array Fields
- $Where Query
- Cursor
- Retrieving Documents Using Cursor
- Pagination
- Advance query option
- Update Operation
- Updating Documents in MongoDB
- $SET
- $Unset and $inc Modifiers
- $inc modifier to increment and decrement
- Replacing Existing Document with New Document
- $Push and $addToSet
- Positional Array Modification
- Adding Elements into Array Fields
- Performing AddToSet
- Removing Documents
- Performing Upsert and Remove Operation

Indexing and Aggregation

- Introduction to Indexing
- Types of Index
- Properties of Index
- Single Field Index on Embedded Document
- Compound Indexes

- Sort Order
- Ensure Indexes Fit RAM
- Multi-Key Indexes
- Compound Multi-Key Indexes
- Hashed Indexes
- Sparse Indexes
- Text Indexes
- Text Search🎬 Index Creation
- Remove Indexes
- Modify Indexes
- Demo—Drop and Index from a Collection
- Measure Index Use
- Control Index Use
- Geospatial Index
- MongoDB's Geospatial Query Operators
- $GeoWith Operator
- Proximity Queries in MongoDB
- Aggregation
- Pipeline Operators and Indexes
- MapReduce
- Aggregation Operations
- Use Distinct and Count Methods
- Use the Group Function
- Developing Node JS Application with MongoDB

SQL Training

- Introduction to RDBMS
- Create Database
- Create Table
- Select
- Select Count
- Select Distinct
- SWGHO
- Where
- Group By
- Having
- Order By

Node.js

- Node.js
- Introduction and Foundation
- Introduction
- The Node.js framework
- Installing Node.js
- Using Node.js to execute scripts
- The Node Package Manager
- The Node Package Manager
- Creating a project

- The package.json configuration file
- Global vs. local package installation
- Working with shrink-wrap to lock the node modules versions
- What is shrink-wrap
- Working with outdated command
- Working with asynchronous programming 🎬 Asynchronous basics
- Callback functions
- Working with Promises
- Advance promises
- Using Request module to make API calls
- Building an HTTP Server with Node.JS using HTTP APIs
- The HTTP protocol
- Building an HTTP server
- Rendering a response
- Processing query strings
- Processing posted data
- File System
- Synchronous vs. asynchronous I/O
- Path and directory operations
- __dirname and __filename
- Asynchronous file reads and writes
- Buffers, Streams, and Events
- Using buffers for binary data
- Flowing vs. non-flowing streams
- Streaming I/O from files and other sources
- Processing streams asynchronously
- Multi-Processing in NodeJS
- Working with Child Process API
- Working with Cluster API for multi-core servers

Python Training Course Syllabus

Session 1: Introduction

- Python Overview
- History Of Python
- Cpython, Jython, PyPy
- Python Features
- Areas Of Application of Python
- Understanding More About Python
- Writing your First Python Program
- Interactive Mode Programming
- Script Mode Programming
- Why do we need Python?
- Program structure

Session 2: Installation and Execution

- Python 2.7 vs Python 3
- Local Environment Setup
- Installing Python on different Platforms(Windows and Linux)
- Python Interpreter and Python Interactive Shell
- Python IDE(Pydev,Pycharm,VIM)

Execution steps

- Interactive Shell
- Executable or script files

- User Interface or IDE

Session 3: Memory management and Garbage collections

- Object creation and deletion
- Object properties

Data Types and Operations

- Numbers
- Strings
- List
- Tuple
- Dictionary
- Other Core Types

Session 4: Statements and Syntax

- Assignments, Expressions and prints
- If tests and Syntax Rules
- While and For Loops
- Iterations and Comprehensions

Session 5: File Operations

- Opening a file
- Using Files
- Other File tools

Session 6: Functions

- Function definition and call
- Function Scope
- Arguments
- Function Objects
- Anonymous Functions
- Variable scope and Returning Values
- Lambda Functions
- Object-Oriented Concepts

Session 7: Modules and Packages

- Module Creations and Usage
- Module Search Path
- Module Vs. Script
- Package Creation and Importing

Session 8: Classes

- Classes and instances
- Classes method calls
- Inheritance and Compositions
- Static and Class Methods
- Bound and Unbound Methods
- Operator Overloading
- Polymorphism

Session 9: Exception Handling

- Standard Libraries
- Modules Used in Python (OS, Sys, Date and Time etc.)
- The Import statements
- Module search path
- Package installation ways
- Errors and Exception Handling
- Handling multiple exceptions

Session 10: Advanced Concepts

NumPy and Pandas

- NumPy - arrays

- Operations on arrays
- Indexing slicing and iterating
- Reading and writing arrays on files
- Pandas - data structures & index operations
- Reading and Writing data from Excel/CSV formats into Pandas

Django Course Syllabus

Introduction to Django:

- Introduction
- Why Django?
- Batteries Included
- Django Principles
- What you Should Already Know
- Course Overview

Django Get Started Hands On

- Installing Data base
- Bringing up Web server
- Defining list and list slicing

Starting Django Project

- Creating a Django Project
- Demo: Creating a Django Project
- The Model-Template-View Pattern
- Demo: Hello, World!
- Mapping URLs
- Demo: URL Mapping
- Django Views
- Demo: Templates

Models

- Demo: Adding Models
- Django Model Classes
- Manage.py Database Commands
- Demo: The Admin Interface
- The Django Admin Interface
- Demo: The Model API
- Save and Delete
- The Model API
- Database Relations

Adding User Home Page

- Demo: Adding Login and Logout Views
- More about URL Mappings
- Demo: A Template for the Home Page
- Authorization with Django
- More about Django Templates
- Demo: Adding the Home View
- URL Mappings for Apps 🎬 Demo: Template Inheritance
- Template Inheritance
- Demo: Login Required
- Demo: Showing Game Data on the Home Page
- Demo: A Custom Manager Class
- The Template Context
- Templates: For and Include Tags

Forms

- Demo: Adding a HTML Form

- Using Django Forms
- Demo: Adding Stypng to the Form with Crispy-Forms
- Demo: Field Options
- Field Options
- Demo: Showing Invitations in a List
- Demo: Accepting Invitations
- Demo: Named Groups
- Named Groups in URLs

Django and REST APIs

Django REST framework

Django-piston

Odds and Ends

- Introduction
- Class-based Views
- Demo: Class-based Views
- Demo: Adding User Signup
- Generic Views
- Debugging Django
- Demo: The Django Debug Toolbar
- Resources

GITHUB