PYTHON Fullstack Training

Course Duration: 6 months, Weekly 6 days, Each Class: 1 hr.
Trainer: D. Chaitanya (IIT Roorkee Alumni), 20+ yrs Exp.

a) **Back End Technologies:** ~ 3.5 month

I) Core Python:

- Demo:
  i. Why Python?
  ii. Who Uses Python today
  iii. What can we do with Python
  iv. How Python Developed and Supported
  v. Python – Technical Strengths.
  vi. What next?

- Python Interpreter
- Program Execution – programmer's view, Python's view
- Installation
  i. Python
  ii. All Related Software: PyCharm, Anaconda
  iii. Setup, configure Python in Laptop
  iv. IDLE – UI, usage, features
- All Numeric types in Python, coding/Hands-on
- Python Variables, objects, References, Shared References, coding/Hands-on
- Garbage Collection of objects
- All built in types in python: Strings, Lists, Dictionaries, Tuples, sets, Files
- Python Statements - coding/Hands-on
- Assignments, Expressions and Prints
  i. if-else, if-elif-else, if-else ternary expression
  ii. while and for loops
  iii. Comprehensions vs regular
  iv. Parallel Traversals: map and zip functions
  v. Other important functions: range, len, enumerate

- Iterations and Comprehensions - coding/Hands-on
- Python online Documentation
- Python Functions – def, nested functions
- Variable Scopes – basics, LEGB rules, global, nonlocal

- Function Arguments - coding/Hands-on
  i. Arguments and shared references
  ii. Arguments passing basics
  iii. Arguments matching basics
  iv. Arguments matching syntax
  v. Multiple Results

- Advanced Function Concepts – Recursive functions, Attributes, Annotations, lambda
- Generators and comprehensions – Generator functions, yield, Generator expression
- Python Modules
  i. Definition, why modules?
  ii. Typical Python program architecture
  iii. Import statement - coding/Hands-on
  iv. How Import works: Find it, Compile it, Run it
  v. Standard library modules
  vi. __pycache__ folder for byte code files
  vii. Module search path

- Module coding Basics
  i. Module creation
  ii. import statement, from statement, from * statement
  iii. Module Namespaces, Namespace dictionaries: __dict__
  iv. Reloading modules

- Module Packages
  i. Package import basics
  ii. Why Package imports?
  iii. Relative import basics
  iv. Why relative imports?
  v. Package Namespaces

- Advanced Module Topics
  i. Data hiding in modules - coding/Hands-on
  ii. Mixed usage modes: __name__ and __main__ - coding/Hands-on
  iii. The as extension for import and from - coding/Hands-on

- Introduction to Python Classes - coding/Hands-on
  i. Why Classes?
  ii. Classes, constructors and Instances
  iii. Method calls
  iv. Attribute inheritance search
  v. OOP is about code reuse
  vi. Subclassing by Inheritance
  vii. Polymorphism in Action
  viii. Class vs instance attributes
  ix. Storing objects in DB – Pickles & Shelves

- Coding with Classes - coding/Hands-on
  i. Abstract super classes
  ii. Nested classes
  iii. Classes vs Modules
  iv. Namespace dictionaries
  v. LEGB scopes rule revisited

- Operator Overloading - coding/Hands-on
  i. Constructors: __init__
  ii. Indexing, Slicing: __getitem__ and __setitem__
  iii. Attribute Access: __getattr__ and __setattr__
  iv. String Representation: __repr__ and __str__
  v. Right side and In-Place Uses: __radd__ and __iadd__
  vi. Call Expressions: __call__
  vii. Comparisons: __lt__, __gt__ and others
  viii. Boolean Tests: __bool__ and __len__
  ix. Destructors: __del__

- Special features of Classes - coding/Hands-on
  i. Inheritance: “IS-a” relationship
  ii. Composition: “HAS-a” relationship
  iii. Pseudo private class attributes
  iv. Bound and unbound method objects
  v. Class objects
  vi. “Mix-In” classes

- Advanced Class Topics - coding/Hands-on
  i. “New style” class model

ii. Diamond inheritance change
iii. MRO: Method Resolution Order
iv. Slots: Attribute Declarations
v. Properties: Attribute Accessors
vi. Static and Class methods
vii. The “super” built-in function

- Exception Basics - coding/Hands-on
  i. Why Exceptions?
  ii. Default Exception handler
  iii. Catching Exceptions
  iv. Raising Exceptions

- Coding Exceptions - coding/Hands-on
  i. The try/except/else statement
  ii. try/finally statement
  iii. raise statement
  iv. assert statement
  v. with/as context managers
  vi. Nesting Exception Handlers

- Exception Objects
  i. Class based exceptions
  ii. Why Exception hierarchies?
  iii. Built-in Exception Classes
  iv. Custom Exceptions

II) <u>Django Framework:</u>

- Django overview, installation
- Web Application Architecture
- Creating and setup a virtual environment
- Project and Apps in Django
- First Project creation and Apps creation, App Regitration
- URL Patterns, creating URLs, Mapping URLs
- path, include in URL mappings
- Django Templates, Create Templates
- DTL – Django Template Language
- Variables, filters, Tags in Django Templates
- Template inheritance

- Partial Templates
- Static files (CSS, Images, etc…)
- Including static files in templates
- Django-DB Connection settings configuration
- Models, Defining & Migrating models
- Django Model Fields, field options
- Admin Panel, creating superuser
- Model registration to Admin Panel
- Django Python Shell
- All CRUD operations with Models
- Bulk CRUD operations with Models
- Types of DB Relationships in Django
- Build One to Many DB Relationship code
- Build One to One DB Relationship code
- Build Many to Many DB Relationship code
- Building HTML Forms, Django Forms, Model Forms
- Validating fields in all forms.
- Validators in Django
- CSRF Token
- Views in Django
- Function based views
- Class based built in views
- File uploading
- Sending Emails with and without Django

III) Django REST Framework:

- Why REST Framework
- REST APIs
- REST API interaction with React JS
- Json data
- DRF Serializers
- DRF Viewsets in detail
- DRF Routers in detail
- DRF Permissions
- DRF Mixins in detail
- DRF Authentication

- Token Authentication
- DRF Sessions
- DRF Settings

IV) Database:

- MySQL Installation, Heidi SQL Installation.
- What is RDBMS?
- RDBMS Terminology.
- Python MYSQL Database Access
- Create Database Connection
- DDL Operations with Databases
- Accessing Database
- DQL Operations with Databases
- DML Operations with Databases
- Aggregations in Database
- Table Joins
- Interacting DB with Python using python libraries.
- Installation of Python libraries for DB access.

b) **Front End Technologies:** ~ 1.5 month

I) HTML5:

- Introduction to HTML
- Basic Structure of HTML
- HTML Editors
- HTML Tags
- Paragraphs, Headings and Text
- Formatting Tags
- HTML Lists
- HTML Images
- HTML Tables
- HTML Forms
- HTML Media

II) CSS3:

- Introduction to CSS
- Types of CSS
- CSS Properties, Selectors and Values
- Applying CSS to HTML
- CSS colors
- CSS Box Model, Margins, Padding,
- Borders
- CSS Text & Font Formats
- CSS Advanced Topics (Effects, Animations, Shadows, Etc.,)

III) Javascript:

- Introduction to JavaScript
- How to Apply JavaScript
- Displaying Output in JavaScript
- Understanding JavaScript Syntax
- Variables & Datatypes
- Operators
- Math and String Manipulations
- Conditional and looping Statements
- Functions
- Validations
- Events
- ES6 Features, http API call

IV) Bootstrap:

- Introduction to Bootstrap
- Bootstrap Setup
- Bootstrap Containers
- Bootstrap Grids
- Bootstrap Tables
- Bootstrap Buttons, Navbars, Alerts
- Bootstrap Carousel
- Bootstrap Forms

V) React JS:

- Introduction to React
- Components
- Props and State
- Handling Events
- Conditional Rendering
- Hooks
- Routing and Navigation
- API Integration
- State Management (Redux)
- Styling in React
- Optimizing Performance
- Forms and Validation
- Testing React Applications
- Deployment
- Advanced Topics (Optional)
- Real-world Projects
- Best Practices and Code Quality