Python Fullstack Development

Core Python

Introduction:

- History
- Features
- Setting up path
- Working with Python Basic Syntax

Data Types Introduction:

- Words order, unordered, sequential, non-sequential, mutable and immutable.
- Introduction - List, Tuple, String, Dictionaries, Set, Frozenset.
- Accessing - List, Tuple, String, Dictionaries, Set, Frozense.
- Shallow and deep copy.

Conditional Statements:

- If, If- else and Nested if-else.

Looping:

- For, While and Nested loops, for-else.

Control Statements:

- Break, Continue & Pass.

Functions:

- Defining a function, Calling a function, Function Arguments, Global & local variables.

Higher order functions:

- List, Dict, Generator Expressions/Tuple Comprehension.
- Map, Filter, Reduce, Lambda (Anonymous functions).

File Operation:

- Open and With open.

Input-Output:

- Printing on screen, Reading data from keyboard.

Parsers:

- Json, Csv, xml, serialization.

Advance Python

OOPs Concept:

- Class
- Object
- Inheritance
- Abstract
- Polymorphism
- Encapsulation

Liked Concepts:

- Types of Methods (Instance, Static, Class)
- MRO
- __str__, __repr__, __call__, __new__, __init__
- Operator Overloading
- Descriptors (Property - setter, getter)

Modules:

- Module, package, Importing module
- base64
- OS, Sys, Path
- Datetime
- Collections
- Argparse
- Logging
- Email sending
- Pdb and ipdb
- Subprocess
- Async
- Random

Exception Handling:

- Exceptions and Exceptions Handling.
- User Defined Exceptions.

Concepts:

- Iterators, Generators, Closers, Decorators.

Multithreading:

- Thread, Starting a thread, Threading module, Synchronizing threads, Multithreaded, Priority Queue.

Regular expressions:

- Match, Search, findall, split, sub and Patterns.

Database:

- Introduction, Connections, Executing queries, Transactions, Handling error.

Django

Introduction:

- What is Django
- Django and Python
- Django's take on MVC
- How to get and install Django

Getting started with DJango (About core three files):

- models.py
- views.py
- urls.py

Migrations:

- Introduction to Migrations
- Data Migrations

Django url patterns and views:

- Designing a Good url Scheme
- Generic Views

DJango Forms:

- Form Classes
- Validations
- Authentication

Django and REST API's:

- DJango REST-API
- DJango Piston

Unit Testing with Django:

- Overview of Unit Testing
- Using Python's Unittest2 Library
- Test
- Test Databases
- Doctests and Debugging best practices

Selenium

Introduction:

- What is automation?
- What is software testing?
- What is selenium?

What is Webdriver:

- Webdriver components

Webdriver for different browsers:

- Configuring webdriver with different browsers
- Configuring with webdriver_manager

What are locators:

- Normal Locators
- Custom Locators
- Css Selector
- Xpaths
- Function in xpath
- Axes

Forms:

- Textboxes
- Radio button
- Checkboxes
- File upload

Select class:

- Selecting dropdowns

Action chain:

- Drag and drown
- Keyboard actions
- Mouse actions

Tables:

Date picker:

Browser options:

- Headless browser
- Browser popups
- Developer options

Pagination:

Window handling:

Alert popups:

Pytest, Page Object model:

Linux

Introduction to Linux

- History and Philosophy of Linux
- Linux Distributions: Overview and Comparison
- Understanding Open Source Software

Getting Started with Linux:

- Installing Linux (Dual Boot and Virtual Machines)
- Basic Linux Terminology (Kernel, Shell, etc.)
- Navigating the Linux File System

Basic Commands and Operations:

- Command Line Interface (CLI) Basics
- Common Commands (ls, cd, cp, mv, rm, etc.)
- File and Directory Operations

File Permissions and Process Management:

- Understanding File Permissions and Ownership
- Changing Permissions and Ownership
- Basic Process Management (ps, top, kill)

Package Management:

- Installing and Updating Software
- Using Package Managers (apt, yum, etc.)
- Understanding Repositories

Text Editing and Shell Scripting:

- Basic Text Editing with Nano and Vi
- Introduction to Shell Scripting
- Writing and Executing Simple Scripts

Networking Basics in Linux:

- Basic Networking Commands (ping, ifconfig, netstat)
- Configuring Network Interfaces
- Understanding Firewalls and SSH

System Administration Basics:

- User and Group Management
- Disk and Storage Management
- System Monitoring and Logs

Security in Linux:

- Basic Linux Security Concepts
- Secure Password Practices
- Introduction to SELinux/AppArmor

Practical Exercises and Real-world Applications

- Setting up a Simple Web Server

- Creating Basic Shell Scripts for Automation
- Troubleshooting Common Issues

Course Review and Further Resources:

- Review of Key Concepts
- Pathways for Advanced Learning
- Community and Online Resources

GIT – Version Control

Introduction to Version Control:

- What is Version Control?
- Overview of Git: History and Purpose
- Git vs Other Version Control Systems

Setting Up and Configuring Git:

- Installing Git (Windows, Linux, macOS)
- Basic Configuration: Setting up User Name and Email
- Introduction to Git Interfaces (CLI, GUIs)

Git Basics:

- Understanding Repositories
- The Three States: Modified, Staged, Committed
- Creating Your First Repository

Git Operations and Commands:

- Basic Git Commands: init, add, commit, status, log
- Understanding Commits and Commit Messages
- Viewing Changes: diff and log

Branching and Merging:

- What is a Branch?
- Creating and Switching Branches
- Basic Merging and Handling Merge Conflicts

Working with Remotes:

- Understanding Remote Repositories
- Cloning a Repository
- Pushing and Pulling Changes

Collaborating with Git:

- Working with Others: Fetch, Pull, Push
- Understanding Forks and Pull Requests
- Collaborative Workflow Patterns (e.g., Feature Branch Workflow)

Advanced Git Features:

- Stashing Changes
- Tagging Releases
- Basic Rebasing

Undoing Changes:

- Reverting Commits
- Resetting and Cleaning
- Dealing with Mistakes

Using Git with Hosting Services:

- Overview of Services like GitHub, GitLab, and Bitbucket
- Remote Repository Management
- Introduction to Basic CI/CD Concepts

Best Practices and Tips:

- Writing Good Commit Messages
- Keeping a Clean History
- Branch Management Strategies

Practical Exercises and Projects:

- Setting Up a Personal Project on Git
- Contributing to an Open Source Project
- Collaborative Simulation with Peers

Course Review and Further Resources:

- Review of Key Concepts
- Advanced Topics in Git (Submodules, Hooks)
- Community and Online Resources

SQL – Database

Introduction to Databases:

- Why Data Base?
- Importance of Databases in Computing?
- Various types of data types?

Relational Database Management System (RDBMS):

- Introduction to Relational Databases
- Relational Algebra and Calculus
- Entity-Relationship Model (ER Model)
- Normalization and Denormalization

SQL (Structured Query Language):

- Basics of SQL (CREATE, SELECT, INSERT, UPDATE, DELETE)
- Advanced SQL Queries (JOINS, GROUP BY, HAVING)
- Operators (IN, LIKE, ISNULL, Union)
- Views, Indexes, and Triggers
- Sub Queries (Inner Query & Outer Query)
- Data Modeling using SQL
- Analytical Functions (Rank, Dense_Rank, Partition by)
-

Database Design:

- Database Design Process
- Functional Dependencies and Normalization
- Designing Tables and Relationships

NoSQL Databases:

- Introduction to NoSQL Databases
- Types of NoSQL Databases (Document, Key-Value, Graph, Column-family)
- Comparison with Relational Databases

Database Management and Transactions:

- ACID Properties
- Transaction Management
- Concurrency Control and Isolation Levels

Remember that this is a detailed outline and can be adjusted based on the duration of the course, the depth of coverage desired for each topic, and the level of expertise of the participants. Additionally, as technology evolves, new tools and practices might emerge, so it's essential to keep the course content up-to-date to reflect the latest trends and best practices.