TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

TREENETRA EDUCATION

PYTHON DEVLOPMENT SYLLABUS

TREENETRA EDUCATION, #15, 1ST CROSS, MUNNEKOLLAL MAIN RD,
NEAR SGR DENTAL COLLEGE ROAD,
MARATHAHALLI, BENGALURU,
KARNATAKA 560037

(HR ISHWARI)
+91-7058-475-504(BATCH ADMISSION)
(HR OFFICE)
+91-8296-789-108(INQUIRY)(INTERVIEW)

BY CHANDRA SEKHAR

TECHNOLOGICAL STACK

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

1.PROGRAMING LANG	2.FRAMEWORK	3.DEVOPS TOOLS
PYTHON	DJANGO,FLASK,FAST API	DOCKER,GIT,JENKINS(CI/CD),GRROVY SCRIPT

4.DATABASE	5.COLUD TECHNOGY	6.API	7.FRONT END
SQL/MYSQL PYTHON SQL ALCHEMY DATABASE CONNECTIOM	AWS:- EC2,S3,,LAMBDA,CLOUD WATCH	POSTMAN REQUEST MODULE JSON	HTML,CSS,JAVA SCRIPT

8.OS	9.PROJECT	10.MANDATORY SKILS
UNIX EC2 AMAZON LINUX	REAL TIME(REPLICA) PROJECT TELECOMM AUTOMATIVE	JIRA,SLACK,TEAMS CONFLUENCE TASK MANAGEMENT REAL TIME INDUSTRY SENARIO[MEETING,CLIENT INTERACTION]

PYTHON

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

SR NO	MODULES NAME	COURSE CONTENT
1	INTRODUCTION	• What is PYTHON? • WHY PYTHON? • History of Python • Features of Python. • Why Python is General Language?/HLL? • Limitations of Python
2.	Python software Installation and Introduction	• Python Distributions, Anaconda Navigator • Download &Python Installation Process in Windows, Unix, Linux and Mac • Online Python IDLE • Python Real-time IDEs like Spyder, Jupyter Note Book, PyCharm, Different Modes of Python
3	Language Initials	• Python Identifiers(Rules and Regulations) • Reserved words and Keywords • Basic Data types in Python(Sequential, non-sequencial, ordered, non-ordered)
4	Data Type and Data Structure	• String(all methods ,slicing and string operation) • List, Tuple, Dictionary, Range(All Operation and Data structure work) • Set,frozenset,bytes, bytearray, None • Type Casting
5	Operators	• Arithmetic Operators • Comparison Operators • Python Assignment Operators • Logical Operators • Membership Operators • Identity Operators • Ternary Operator • Operator precedence • Difference between “is” vs “==”

PYTHON

SR NO	MODULES NAME	COURSE CONTENT
6	Flow Control Control Statement	• Conditional control statements , If , If-else ,If-elif-else ladder ,Nested-if-else • Loop control statement, for ,while ,Nested for loops ,Branching statements • Break,Continue,Pass • Case studies- Pattern making (letters and Diagrams
7	Functions	What is Function?--Advantages of functions., Syntax and Writing function,Calling or Invoking function Classification of Functions- On the basis of argument and return value No arguments and No return values,With arguments and No return values,No arguments and with return values Positional argument type functions,Default argument functions.variable length arguments function, keyword argumentsfunction(*arg),Variable length keyword arguments functions(**kwargs) zip() in Python What is variables .Global and Local Variable Anonymous functions(Lambda, filter, map, reduce)Code Optimization(Comprehenssion(list,dict) Function Aliasing DecoratorGenerator
8	Python Module and Package	• What is Module,Type Of Module- Pre Define, User Define,Function/Class Based Module,How to import modules • Module Alias,Math,random,os,sys,time,datetime • Organizing python project into packages,Types of packages – pre defined, user defined.,Package v/s Folder • .py file,Importing package,pip
9	File Handling	• What is Function?--Advantages of functions., Syntax and Writing function,Calling or Invoking function • Classification of Functions- On the basis of argument and return value • No arguments and No return values,With arguments and No return values,No arguments and with return values • Positional argument type functions,Default argument functions.variable length arguments function, • keyword argumentsfunction(*arg),Variable length keyword arguments functions(**kwargs) • zip() in Python • What is variables .Global and Local Variable • Anonymous functions(Lambda, filter, map, reduce)Code Optimization(Comprehenssion(list,dict) • Function Aliasing • Decorator,Generator

PYTHON

SR NO	MODULES NAME	COURSE CONTENT
10	OOPs	• Procedural v/s Object oriented programming, Classes and Objects, How to define class in python • Types of variables – instance variables, class variables. Types of methods – instance methods, class method, static method, Object initialization, ‘self’ reference variable, ‘cls’ reference variable, Property () object threory • Principles of OOP – Encapsulation, Abstraction (Data Hiding) • Creating object properties using setaltr, getaltr functions,Inner classes,Class re-usability • Inheritance – single, multi-level, multiple, hierarchical and hybrid inheritance and Diamond inheritance, Method resolution order (MRO) • super () • Constructors in inheritance • Object class • Duck typing interview question • What is polymorphism • Runtime polymorphism • Overriding • i) Method overriding • ii) Constructor overriding • Method overriding in Multiple inheritance and Hybrid Inheritance • Overloading • i) Method Overloading • ii) Constructor Overloading • iii) Operator Overloading
11	Exception Handling	• What is Exception?, Why exception handling?, Syntax error v/s Runtime error • Exception codes – AttributeError, ValueError, IndexError, TypeError… • Handling exception – try except block, Try with multi except, Handling multiple exceptions with single except block • Finally block, Try-except-finally, Raise keyword • Case study of finally block, Custom exceptions / User defined exceptions • Need to Custom exceptions
12	OS module	• What is Module,Type Of Module- Pre Define, User Define,Function/Class Based Module,How to import modules • Module Alias,Math,random,os,sys,time,datetime • Organizing python project into packages,Types of packages – pre defined, user defined.,Package v/s Folder • .py file,Importing package,pip

DJANGO

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

SR NO	MODULES NAME	COURSE CONTENT
1	**DJANGO**	**1:Introduction to Django** **Introduction** Overview of Django The MVC/MVT architecture Setting up the development environment **Getting Started** Installing Django Creating a new Django project Understanding project structure **Basic Concepts** URLs and views Django’s request-response cycle First Django app: Creating and running a simple view **2:Models and Databases** **Introduction to Models** Defining models Database setup and configuration **Database Migrations** Making migrations Applying migrations

DJANGO

SR NO	MODULES NAME	COURSE CONTENT
1	**DJANGO**	**3:Templates and Static Files** **Templates** Creating templates Template language and syntax Template inheritance **Static Files** Managing static files (CSS, JavaScript, images) Configuring static file handling **4:Forms and User Input** **Forms** Creating and handling forms Form validation Model forms **User Input** Handling POST requests Processing form data **5:Authentication and Authorization** **User Authentication** Built-in authentication system User registration and login **Permissions and Authorization**

DJANGO

SR NO	MODULES NAME	COURSE CONTENT
1	**DJANGO**	**6:Advanced Models and Queries** **Advanced Model Concepts** Model relationships (OneToOne, ForeignKey, ManyToMany) Custom model methods and managers **Complex Queries** Aggregations and annotations Q objects and query expressions **7:REST APIs with Django REST Framework (DRF)** **Introduction to DRF** Setting up Django REST Framework Serializers and views **Building APIs** CRUD operations Authentication and permissions in APIs **T8:esting and Debugging** **Testing** Writing tests in Django Using Django's test framework **Debugging** Common debugging techniques Using Django Debug Toolbar

FLASK FRAMEWORK

SR NO	MODULES NAME	COURSE CONTENT
1	**FLASK FW**	**1: Introduction to Flask** **Introduction** Overview of Flask Flask vs. Django: A comparison Setting up the development environment **Getting Started** Installing Flask Creating a simple Flask application Understanding the project structure **Basic Concepts** Routes and views Flask’s request-response cycle Running the Flask development server **2: Templates and Static Files** **Templates** Introduction to Jinja2 templating engine Creating and rendering templates Template inheritance **Static Files** Managing static files (CSS, JavaScript, images) Configuring static file handling in Flask **Week 3: Forms and User Input**

FLASK FRAMEWORK

SR NO	MODULES NAME	COURSE CONTENT
1	**FLASK FW**	**3: Forms and User Input** **Forms** Creating HTML forms Handling form submissions Validating form data with WTForms **User Input** Handling GET and POST requests Processing form data and providing feedback **4: Models and Databases** **Introduction to SQLAlchemy** Setting up SQLAlchemy with Flask Defining models and creating a database **Database Operations** Performing CRUD operations Querying the database Understanding relationships between models **5: Authentication and Authorization** **User Authentication** Setting up user authentication Implementing login and registration functionality **Authorization**

FLASK FRAMEWORK

SR NO	MODULES NAME	COURSE CONTENT
1	**FLASK FW**	**6: Blueprints and Modular Applications** **Introduction to Blueprints** Understanding the need for blueprints Creating and using blueprints to structure the application **Modular Applications** Organizing code with blueprints Sharing data between blueprints**7: REST APIs with Flask** **Building RESTful APIs** Introduction to Flask-RESTful Creating API endpoints Handling request and response data **Advanced API Concepts** Authentication and authorization for APIs Pagination, filtering, and sorting **8: Testing and Debugging** **Testing** Writing unit tests for Flask applications Using Flask's test client **Debugging** Common debugging techniques Using Flask-DebugToolbar

API

MODULES NAME	COURSE CONTENT	
API Request Module	Introduction to REST API What is WebServices? Why WebServices so Popular? Overview of SOAP WebServices and REST WebServices What is REST API? How is different from SOAP WebServices? Base URL and REST Resources Understanding of GET, POST, PUT, DELETE Getting Started with REST API Testing Understanding REST API Testing Part Manual Testing on REST API using POSTMAN How to Trigger New Request and Validate Response using REST API	REST API Automation >>REQUEST MODULE Overview Automation on REST Applications Benefit of Automation Testing for Applications Setting up Project Tools for Automation Testing Understanding How to setup for Automation Testing Mock Sample Testing How to validate Responses Depth Automating REST API Automating POST request with Payload Applying Advanced validation on Responses Passing Previous values into subsequent Request Business case for Error Responses

DEVOPS TOOLS

SR NO	MODULES NAME	COURSE CONTENT			
1	DOCKER	• Introducing Docker,Comparing VM and Docker • Docker –An Architectural overview,The Docker Hub A brief Introduction • Preparing docker-machine – Installation and configuration,Start	• containerizing • Play with docker images • Dockerfile Directives,USER and RUN,RUN Order of Execution • ENV,CMD vs. RUN,ENTRYPOINT,EXPOSE • Docker Container Volume Management – An introduction,Docker	• Networking concepts • List and Inspect,Create and Remove,Assign to Containers • Exercise: Creating a Custom Image from a Dockerfile,Exercise: Managing Containers • Exercise: Adding External Content to Containers	
2	git	**Introduction to Git** Version Control/Revision Control system, The types of VCS, The benefits of using VCS What is GIT? The difference between GIT and other VCS, Where to use Git and where not to use it Web Scale architecture	**Getting Started With Git And Its Architecture** Install Git on Windows/Mac/Linux/Unix Understand Git file life cycle Create Git repository - Local repository and configure it to GitHubCreate a repository on GitHub and clone it,Understand basic Git commands,Git command shortcuts.Revisit the Git file lifecycle with Git terminology	**Branching in Git** How Git internally manages branchesHow to switch between branches and different commitsTwo way merge and three way mergeMerging StrategiesMerging and rebasing (using Source tree)Git tagsDifference between branching and tagging	**Git Methodology** What is GIT workflow? Advantages of workflowDifferent types of workflows in GitCentralized WorkflowFeature branch workflowGitflow WorkflowForking WorkflowHow to use git in real time open source projects
3	Jenkins	• Introducing Continuous Integration and Jenkins • Agile Development • Continuous Integration • History of Jenkins • State of the Jenkins community	• Installing and Running Jenkins • Running Jenkins from the jar file • Installing Jenkins in a servlet container	• A Jenkins Job • Creating a job • Configure a job • Run a job manually • Run a job when source code is checked into version control • Run a job on a regular	schedule Change reporting Code coverage Static Analysis Performance reporting Style checking

CLOUD-AWS

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

SR NO	MODULES NAME	COURSE CONTENT
1	EC2	• Virtualization • Amazon Elastic Compute Cloud (EC2) and Its Benefits • Amazon Machine Image (AMI) • Security Groups in AWS • Authentication through Key-pair • Public IP vs. Elastic IP • Instance Store • Elastic Block Store (EBS), Its Features and Volume Types
2	Database Services and Analytics	• Amazon RDS and its benefits • Read Replica RDS • IAM Authentication • DynamoDB
3	Networking and Monitoring Services	• AWS CloudWatch • AWS CloudTrail • AWS Config

LINUX/MS

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

SR NO	MODULES NAME	COURSE CONTENT			
1	LINUX	• Understanding Linux Concepts • Download, Install and Configure • System Access and File System • Linux Fundamentals Linux System Administration Networking, Servers and System Updates	Commands Syntax File Permissions (chmod) File Ownership (chown, chgrp) Getting Help (man, what is etc.) TAB completion and up arrow keys Adding text to file Standard output to a file (tee command)	File Maintenance Commands File Display Commands Filters / Text Processing Commands (cut, sort, grep, awk, uniq, wc)	Linux vs. Windows Commands Quiz, Homework and Handouts Compare Files (diff, cmp) Compress and un-compress files/directories (tar, gzip, gunzip) Truncate file size (truncate) Combining and Splitting Files (cat and split)
2	MANDATORY SKILS	• JIRA • SLACK • TEAMS • CONFLUENCE	• TASK MANAGEMENT • REAL TIME INDUSTRY SENARIO[MEETING • CLIENT • INTERACTION]		

PROJECT-TELECOM

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

Introduction to OSS and BSS	OSS Components	BSS Components	Integration of OSS/BSS	Service Fulfillment and Assurance
• Overview of OSS/BSS • Evolution and History • Importance in Telecommunication Networks • Key Differences between OSS and BSS	• Network Management Systems (NMS) • Service Managemen • tFault Management • Configuration Management • Performance Management • Security Management	• Customer Relationship Management (CRM) • Order Managemen • tBilling and Revenue Management • Product Lifecycle Management • Customer Self-Service	• Data Flow between OSS and BSS • Process Automation and Workflow Management • Integration Technologies (APIs, ESB)Interoperability Challenges	• Service Fulfillment Processes • Service Assurance Processes • Activation and Provisioning • Monitoring and Reporting

Network and Service Management	Revenue Management and Billing	Customer Relationship Management (CRM)	Order and Inventory Management	Next-Generation OSS/BSS
• End-to-End Service Management • Network Planning and Optimization • Real-Time Network Analytics • Service Level Agreements (SLAs)	• Billing Systems and Processes • Charging Mechanisms (Prepaid, Postpaid) • Revenue Assurance • Fraud Management	• Customer Data Management • Sales and Marketing Automation • Customer Support and Service • Customer Experience Management (CEM)	• Order Handling and Processing • Inventory Systems • Supply Chain Management • Product Catalog Management	• Virtualization and Cloud Computing • Software-Defined Networking (SDN)Network Function Virtualization (NFV)IoT and OSS/BSS Integration • Artificial Intelligence and Machine Learning in OSS/BSS

PROJECT-AUTOMATIVE

TREENETRA EDUCATION OF SOFTWARE TECHNOLOGY

TREENETRA

Introduction to Automotive IT	Vehicle Electronics and Embedded Systems	Communication Protocols in Vehicles	Connected Vehicles and Telematics	Advanced Driver Assistance Systems (ADAS)
• Overview of Automotive IT • Evolution of Automotive Electronics and Software • Importance of IT in the Automotive Industry • Automotive IT Market Trends and Future Outlook	• Basics of Automotive Electronics • Microcontrollers and Microprocessors in Vehicles • Embedded Systems Design and Development • Real-Time Operating Systems (RTOS)Automotive Sensors and Actuators	• In-Vehicle Networking (CAN, LIN, Flex • Ray, Ethernet)Vehicle-to-Everything (V2X) Communication • Wireless Communication Technologies (Bluetooth, Wi-Fi, 5G)Diagnostic Communication Protocols (UDS, OBD-II)Data Communication and Integration	• Introduction to Connected Vehicles • Telematics and Infotainment Systems • Over-the-Air (OTA) Updates • Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication • Data Privacy and Management in Connected Vehicles	• Introduction to ADAS Technologies • Sensor Technologies (Radar, LiDAR, Cameras)Algorithms for ADAS (Computer Vision, Sensor Fusion)Safety and Reliability in ADAS • Testing and Validation of ADAS

THANKS

TREENETRA EDUCATION

(WWW.TREENETRA.IN)

+91-7058-475-504

+91-8296-789-108