ihub
INNOVATIVE PLACEMENT SOLUTIONS

powered by Quality Thought®

FULLSTACK PYTHON

WITH React JS+ Angular JS+AWS+DSA+AI

#302, Nilgiri Block, Aditya Enclave, Ameerpet, Hyd.,Telangana-500016
qualitythought.in | ihubtalent.com
81437 41509 | 91603 75471

ihub
INNOVATIVE PLACEMENT SOLUTIONS | powered by Quality Thought®

TRAINING

- Real Time Training
- Live Project/Use Cases
- LMS Access for 6 Months
- Resume Preparation
- Doubts Clarification
- 1 Mock Interview
- Interview Questions & Readiness Preparation
- Placement Referral Support
- Course Completion Certificate

Duration: Max of 4 months
Daily 1.5-3 hrs

Timings:
Morning 07:00AM-09:00AM | Evening 06:00PM-09:00PM

JOIP

- Full Day Training
- Real Time Training
- Live Project / Usecases
- LMS Access for 8 Months
- Soft Skills & Aptitude Classes
- Resume Preparation
- Doubts Clarification
- Weekly 1 Mock Interview
- Interview Questions Preparation
- Monthly Placement Screening Tests
- Assignments / Mock Tests
- Interview Readiness Sessions
- Mega Drive Selection Mandatory For Placements
- Pay After Placement
- Placement Assistance
- Course Completion Certificate

Duration: Max of 6 months - Daily 1.5-3 hrs
Timings: 08:00AM-06:00PM

I&I–100% PLACEMENT ASSISTANCE

- Full Day Training + Internship from Day 1
- Real Time Training
- Internship @IT Company-Ramana Soft
- Live Project/ Client Projects Implementation
- LMS Access Upto 1 Year
- Resume Preparation
- Doubts Clarification
- Weekly 1 Mock Interview
- Assignments/Mock tests
- Soft Skills & Aptitude Classes
- Monthly Placement Screening Test
- Interview Questions & Readiness Preparation
- Pay After Placement
- No Mega Drive Selection
- Personalised Assistance for Complex Tasks
- 100% Placement Assistance - Until you're Hired
- Internship Completion Certificate-6 Months from Ramana Soft or Client
- Course Completion Certificate - Quality Thought

- Max Of 6 Months | 8 Hrs Per Day
- 3rd & 4th Month Internship- 4 Hrs
- 1st & 2nd Month Internship - 2 Hrs
- Internship Duration: 6 Months
- Training Duration: 4 Months
- 5th & 6th Month Internship- 8 Hrs

QSIP–100% PLACEMENT ASSISTANCE

- Full Day Training Paid Internship from Day 1
- Real Time - Industry Training
- Internship @IT Company-Ramana Soft
- Live Projects / Client Projects Implementation
- LMS Access Upto 1 Year
- Resume Preparation
- Doubts Clarification
- Weekly 1 Mock Interview
- Assignments/MockTests
- Soft Skills & Aptitude Classes
- Monthly Placement Screening Test
- Personalized Support from a Dedicated Placement Officer
- Interview Questions & Readiness Sessions
- No Mega Drive Selection
- Personalised Assistance for Complex Tasks
- Eligibility Criteria 2020-2025 Passed Outs Only
- Only Freshers Internship
- HR Approval Mandatory
- Ramana Soft Interview Mandatory
- 100% Placement Assistance - Until you're Hired
- Internship Completion Certificate-6 Months from Ramana Soft or Client
- Course Completion Certificate - Quality Thought

180 Days Paid Internship Program Overview

- Max Of 6 Months | 8 Hrs Per Day
- 3rd & 4th Month Internship- 4 Hrs
- 1st & 2nd Month Internship - 2 Hrs
- Internship Duration: 6 Months
- Training Duration: 4 Months
- 5th & 6th Month Internship- 8 Hrs

Content Explanation

Counsellor Name / Phn :

Python Demo Date :

Python Demo Time :

Python Fee :

Python Duration :

Trainer Name :

Python Internship Execution @RamanaSoft

RamanaSoft Joining Document Requirements

Degree/PG Certificates 2 Copies
Inter Memo 2 Copies
SSC Memo 2 Copies
Adar Card 2 Copies
Pan card 2 Copies
3 Passport Sizes photos (Colour)

Company Details: Ramana Soft
Project Process
Activity_Id Activity/Process Owner
1. Business Grooming
2. Functional Walk-thru
3. Make them to write Test cases
4. Execute Testing People
5. Report bugs to Dev(UI/Java)
6. Jira
7. Deployment
8. Regression Testing People

Testing Process Agile Testing-Sprint 1,2,3,....N
Sprint 1 Schedule Sprint-1 starts

1.Joining Formalities with HR
2.Providing Access
➪ Outlook ➪ Slack
➪ JIRA ➪ DB
➪ Application URL access and VPN
➪ Providing KT , Scrum call
➪ Providing KT, Scrum call
➪ Scrum call, Sprint grooming, Sprint planning
➪ Scrum call, Analysing requirements
➪ Analysis and Writing test scenarios, Scrum call
➪ Writing Test cases, Scrum call
➪ Writing Test cases, Scrum call
➪ Getting review of test cases, Scrum call
➪ Smoke testing, Scrum call
➪ Execution of test cases, Scrum call
➪ Execution of test cases, Scrum call
➪ Execution of test cases, Retest, Scrum call
➪ Retesting, Regression testing, Scrum call
➪ Sprint-1 Closer Sprint review and retrospective meeting, Sanity check and QA sign off, Scrum call Planning for second sprint

Tools
➪ Outlook ➪ Slack ➪ JIRA ➪ DB

Meetings
Daily Scrum Meetings
Sprint Planning
Sprint Retrospective
Sprint Review Meeting
Project Domain:
Banking,Insurance, Health Care, Retail, Ecommerce, IOT, AI/ML,Gaming,Travel

Aptitude & Reasoning

Quantitative

- **Basic Maths**
- **Algebra**
- **Percentages**
- **Profit And Loss**
- **Discounts**
- **Averages**
- **Time and Work**
- **Chain Rule**
- **Pipes and Cisterns**
- **Ratios**
- **Proportions**
- **Partnerships**
- **Time and Distance**
- **Trains**
- **Boats and Streams**
- **Simple Interest**
- **Compound Interest**

Reasoning

- Directions
- Letter Series
- Number Series
- Coding - Decoding
- Blood Relations
- Statement and Assumption
- Analogy
- Odd Man Out Series
- Venn Diagrams
- Mirror Images
- Water Images
- Arranging in Order
- Paper Folding / Cutting
- Grouping
- Counting the figures
- Clocks
- Calenders
- Seating Arrangements

Data Interpretation

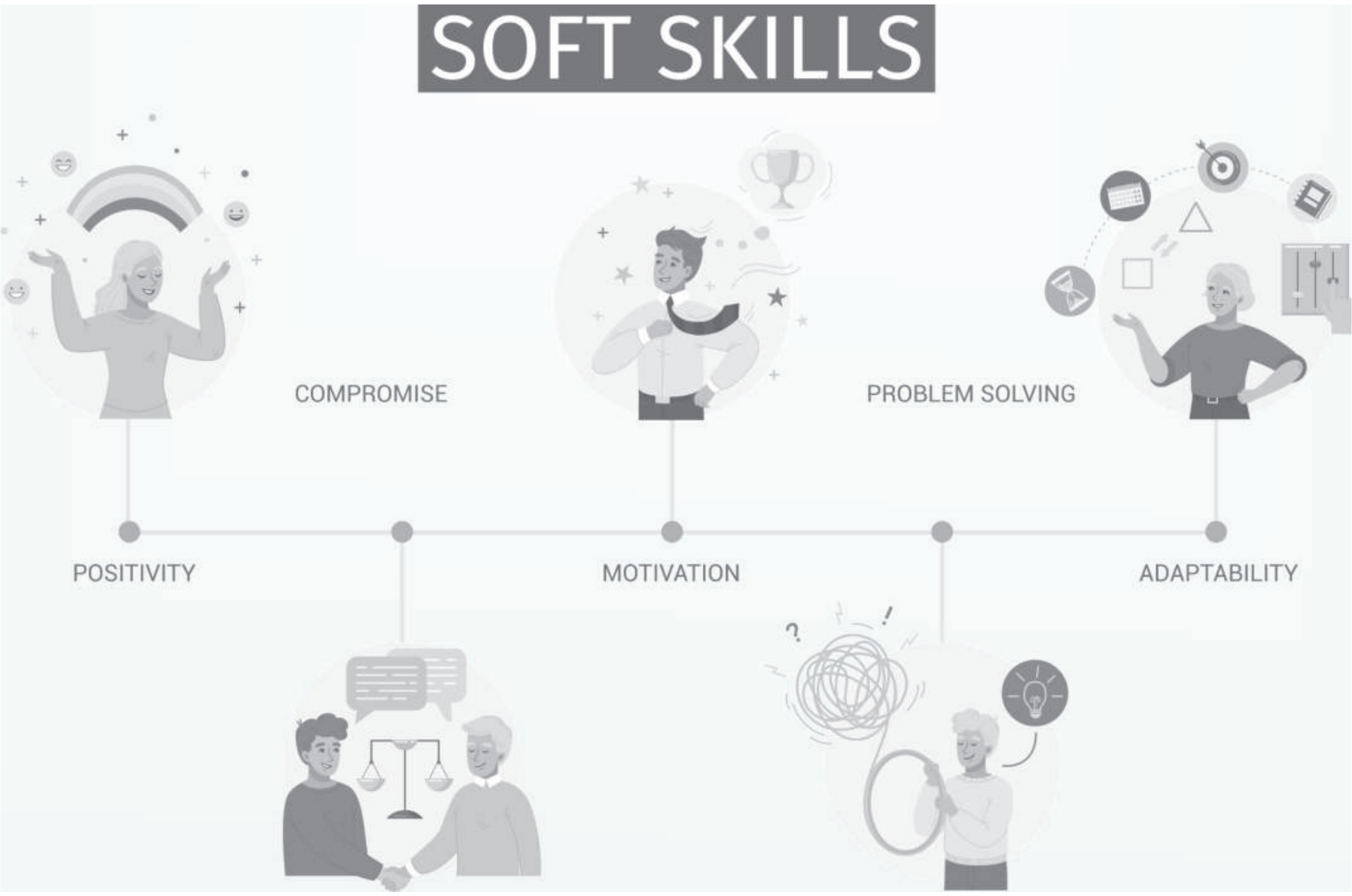
- Bar Charts
- Line Charts
- Pie Charts
- Table Charts

QUALITY IN SOFT SKILLS

- ➪ English Skills: Basic Grammar – Parts of Speech, Preposition, Tenses, Usage, Auxiliaries and Modals, Sentence Formation and Phonics – Sounds, Pronunciation and Articulation
- ➪ LSRW Skills: Listening, Speaking, Reading and Writing Skills, Techniques & Tips and its importance
- ➪ Communication Skills: Types, Levels, Styles, 7 C's, Barriers & How to overcome Barriers, Importance, Interpersonal Skills and Activities
- ➪ Business Communication Skills: Telephone Etiquettes, How to Write Official Letters, Drafting Official Emails, Writing Memos & Blogs and Professional Ethics
- ➪ Public Speaking Skills: Reduce Fear & Shyness, Openness & Transparency, Pre-Requisites for Public Speaking
- ➪ Presentation Skills: Preparation, Tips & Techniques and Body Language
- ➪ Employability Skills: Job Readiness – Resume Formats, How to Face Interview, PI / GD / JAM, Interview Questions, Power Dressing, Corporate Grooming, Goal Setting, Corporate Culture & Work Ethics

VALUE ADDITION:

- ➪ # International Tools: SWOT Analysis, SOAR Analysis, Transaction Analysis, Johari Window, IKIGAI Japanese Concept, Mind Mapping, and Enneagram Personality Type
- ➪ # VERSANT + Voice & Accent Training with Accent Neutralization to reduce MTI and Regional Slang



Core PYTHON

- **Python Introduction & setup environment**
- **What are the software's required to learn python**
  - a. Python 3.7.0 version installation
  - b. Visual studio code installation
- **An identifier(variable)**
  - a. What an identifier(variable)
  - b. Rules for an identifier(variable)
- **Data types in Python**
  - a. Integer data type
  - b. Floating data type
  - c. String data type
    - ➩ join()
    - ➩ len()
    - ➩ replace()
    - ➩ split()
    - ➩ strip()
    - ➩ rstrip()
    - ➩ lstrip()
    - ➩ upper()
    - ➩ lower()
    - ➩ slice operator with + index
    - ➩ slice operator with - index
- **Boolean data type**
- **Complex data type**
- **List data type**
  - ➩ List with their properties
  - ➩ append()
  - ➩ remove()
  - ➩ insert()
  - ➩ extend()
  - ➩ pop()
  - ➩ index()
  - ➩ sort()
  - ➩ sorted()
  - ➩ len()
  - ➩ copy()
  - ➩ clear()
- **Tuple data type**
  - ➩ Tuple with their properties
  - ➩ max()
  - ➩ min()
  - ➩ len()
- **Set data type**
  - ➩ Set with their properties
  - ➩ copy()
  - ➩ clear()
  - ➩ len()
  - ➩ intersection()
  - ➩ update()
  - ➩ union()
- **Dictionary data type**
  - ➩ Dictionary data type with their properties
  - ➩ keys()
  - ➩ values()
  - ➩ items()
  - ➩ popitem()
  - ➩ get()
  - ➩ copy()
  - ➩ clear()
  - ➩ update()

a. Bytes data type
b. ByteArray data type
c. Frozenset data type
d. Range data type
e. None data type
f. Working with input() function with their rules
g. Typecasting in python
h. Working with eval() function with their rules

- **An operators in Python**
  - a. Arithmetic operators
  - c. Assignment operators
  - d. Logical operators
    - ➩ Logical and operator
    - ➩ Logical or operator
    - ➩ Logical not operator
- **Equality operators**
- **Comparison operators**
- **Chaining operators**
- **Ternary operators**
- **Special type of operators**
  - ➩ Identity operators
  - ➩ Membership operators
- **Bitwise operators**
  - ➩ Bitwise and operator
  - ➩ Bitwise or operator
  - ➩ Bitwise exclusive or
  - ➩ Bitwise complement operator
  - ➩ Bitwise left-shift operator
  - ➩ Bitwise right-shift operator
- **Working with Input & Output functions**
  - ➩ a. Input() & print() functions
  - ➩ b. Working separator attribute
  - ➩ c. Working with end attribute
  - ➩ d. Formatted string
  - ➩ e.Replacement operator
- **Command Line argument (CLA) in Python**
  - a. Working with sys module with argv variable
  - b. argv variable with various operations
- **Control Statements in Python**
  - ➩ Decision making or conditional statements
  - ➩ If statement
  - ➩ nested if statement
  - ➩ if else statement
  - ➩ if elif else statement

- **Iterative statements**
  - ➪ for loop ➪ nested for loop
  - ➪ while loop
  - ➪ nested while loop

- **Transfer statements**
  - ➪ pass statement ➪ break statement
  - ➪ continue statement

- **Working with zip () function**
- **List comprehension**
- **Tuple comprehension**
- **Set comprehension**
- **Dictionary comprehension**

- **Functional Programming language in Python**
  - a. What is function
  - b. Types of functions
  - c. Why do use functions in real time applications
  - d. How to create a function in python
  - e. What is __name__==“”__main__“
  - f. Formal parameters
  - g. Actual parameters
  - f. Arguments in function
    - ➪ Positional argument
    - ➪ Default argument
    - ➪ Keyword argument
    - ➪ Variable length argument
    - ➪ Keyword variable length argument
    - ➪ Difference between *ob1j & **obj2

- **Nameless function**
  - ➪ Working with lambda keyword
  - ➪ filter() function ➪ map() function
  - ➪ reduce() function

- **Inner or Nested function**
- **Packages in Python**
  - a. What is module
  - b. What is package
  - c. What is library
  - d. What is framework
  - e. How package is important in real world software's
  - f. Complete structure of package
  - g. Complete structure of nested package

- **Modular Programming Language in Python**
  - a. Why modular programming language
  - b. Import & export data from one to another module
  - c. Various possibility of import & export the data
  - d. Working with reload () functions
  - e. Working with math module
  - f. Working with random module

- **Modular Programming Language in Python**
  - a. Why modular programming language
  - b. Import & export data from one to another module
  - c. Various possibility of import and export the data
  - d. Working with reload () functions
  - e. Working with math module
  - f. Working with random module

- **Pandas Library**
  - a. What are pandas
  - b. How to install pandas
  - c. How pandas ruling in data science applications
  - d. Working on Data Frame object
  - e. Working with pandas predefine functions
    - ➪ head() function ➪ tail() function
    - ➪ max() function ➪ min() function
    - ➪ count() function ➪ sum() function
    - ➪ sum(1) function ➪ sort() function

- **Working with iterating methods in pandas**
  - a. Iteritems ()
  - b. Iterrows ()
  - c. Itertuples ()

- **NumPy Library**

a. What is NumPy
b. How NumPy is ruling in data science applications
c. How to install NumPy
d. Working with zero to nth dimension arrays
e. What is ndim
f. What is ndmin
g. Slicing with numpy
h. Working with shape attribute
I. Working with reshape function
j. Applying the loops on NumPy
k. Working with predefine functions in NumPy

- Advance Data Structure in python

a. Working on Stack with their rules
b. Working with Queue with their rules
c. Working binary tree with their rules
d. Working with linked list
  ➪ Single linked list
  ➪ Double linked list

- **Modular Programming Language in Python**
  - a. Why modular programming language
  - b. ülmport & export data from one to another module
  - c. Various possibility of import and export the data
  - d. Working with reload () functions
  - e. Working with math module
  - f. Working with random module

- **Pattern Examples**
- **Important Interview Questions & Answers**

Adv. PYTHON

▶ Object oriented Programming language in Python.

a. What is class
b. How to create class
c. What is an object
d. How to create an object
e.What is constructor
f. What is Instance method
(Non static method)
g. What is class method
h. What is static method
I. What is Instance variable
j. What is Static variable
k. What is Local variable
l. Working with GC module
m. Working with Inner classes
n. hat is composition and aggregation

▶ Inheritance

a. Duck-Typing
b. Operator overloading
c. Method overloading
d. Method overloading with default argument
e. Method overloading with variable length argument
f. Constructor overloading
g. Constructor overloading with default argument
h. Constructor overloading
with variable length argument
I. Method overriding
j. Constructor overriding
➪ Working on Encapsulation
➪ Abstract method ➪ Abstract class
➪ Interface ➪ Concrete classes
➪ Access modifier

▶ File Handling in Python

a. Why file is required
b. What is file handling
c. How to open a file
d. Working with various modes of file
e. Working with write() and write lines()
f. Working with read()
and read line() and read lines()
g. Working with 'with' statement
h. Working with pickling & unpickling
I. Working with CSV module
j. Working with Zipping and Unzipping
k. Working with object serialization
and object deserialization

▶ Exception Handling in Python

a. Types of errors in programming language
b. What is exception
c. What is main objective of an exception
d. Working with try & except block
e. Working with default exception
f. Working with try & except & else & finally block
g. Working with nested try &
except & else & finally block
h. Difference between try & finally block

▶ Decorators in Python

a. What is decorator
b. Why decorator is required
c. Working with
@decor_name decorator
d. Working with decor function

▶ PBDC in python

a. Why PDBC
b. Working XAMPP tool for
MySQL Database
c. How to install MySQL. Connector drivers
d. Performing all database queries

▶ Generators in python

a. What is generators
b. Why do we require generators
c. Working with yield keyword

▶ Multi-Threading

a. What is multi-threading
b. Types of multi-threading
c. What is Thread
d. How many ways we can
create thread in python
e. How to improve the application
performance with threading
f. Synchronization and Asynchronization

▶ Assertion in Python

a. What is assertion
b. Types of assertion
c. Working with assert keyword to
develop testcases
d. Scripts for to perform
debugging operations using assertion

▶ Web Scraping with Regular expression

a. What is regular expression
b. Working with re module in python
c. Working with character classes
d. Working with predefine classes
e. Working with quantifiers
f. Regex object for Indian mobile number
g. Regex object for email
h. Working with predefine functions
➪ match() ➪ fullmatch()
➪ search() ➪ findall()
➪ sub() ➪ subn()
➪ split()
I. What is web scarping
j. How to fetch real time data
using web scarping process

Web development

(UI or Front End)

- **HTML & HTML5**

  a. What are the software require to learn UI
  b. Working with Fav icon for our frontend
  c. Working with heading tag
  d. Working with formatted tag
  e. Working with paragraph tag
  f. Working with marquee tag
  g. Working with image tag
  h. Working with anchor tag
  I. Working with table tag
  j. Working with form and its components
  k. Developing the complete form with validation
  l. Working with HTML 5 tags
  m. Working with div tag

- **CSS & CSS3**

  a. What is CSS
  b. Types of CSS
  ➪ Inline CSS
  ➪ Internal CSS
  ➪ External CSS

  c. What are selectors and its types
  ➪ Using tag-based selector
  ➪ Using class-based selector
  ➪ Using Id selector
  ➪ Using group by selector
  ➪ Using universal selector
  d. Working with float property
  e. Working visibility property
  f. Working with display property

  ➪ none

  ➪ inline
  ➪ inline-block
  ➪ block
  ➪ flux

  g. Working with position property
  ➪ static
  ➪ absolute
  ➪ fixed
  ➪ sticky
  ➪ inherit

  h. Working with media query

- **JavaScript**

  a. Why JavaScript
  b. What is JavaScript
  c. How many ways we can JavaScript
  ➪ Inside the body tag
  ➪ Inside the head tag
  ➪ External JS
  d. Working with variable declarations
  f. Working with document. Write()
  g. Working with console.log()
  h. Working Dialog boxes
  ➪ Alert() or window.Alert()
  ➪ Comfirm() or window confirm()
  ➪ Prompt() or window.promprt()
  j. Working with Data types
  ➪ Primitive data type
  ➪ Non primitive data type
  k. Working with operators
  l. Working with control statements
  m. Working with events
  n. Working with functions
  o. Working with DOM
  p. Working with High order functions
  q. Working with promises in JS
  r. Working with OOPS in JS
  ➪ Creating a class
  ➪ Creating an object
  ➪ Constructor
  ➪ Inheritance
  ➪ Super keyword
  ➪ Encapsulation
  ➪ Prototypes
  ➪ Polymorphism

BOOTSTRAPT (4 & 5)

- Why bootstrap
- What is bootstrap
- Features of bootstrap
- What is grid system
- What are offset classes
- Working with typography
- Working with buttons
- Working with jumbotron
- Working with Progress bar
- Working with paginations
- Working with forms
- Working with cards
- Working with navbar tag
- Working with model
- Working with panel
- Working with validations states
- Working with toggle and collapse classes

Django

- Prerequest to learn Django
- What is Django
- Features of Django
- How to create a project
- How to create application
- Working with complete file structure in Django after creating Django project & application
- How to create more than one application
- How to create a urls.py file at application to improve performance
- Working with MVT design pattern
- Working with templates folder for frontend development
- Working with Static folder for frontend design development
- Implementing JavaScript in Django
- Implementing bootstrap in Django
- Working with model class in Django
- Working with Django forms
- Working with Django model relationship
  a. One To One Relationship
  b. Many To One Relationship
  c. Many To Many Relationship

- Django Exceptions
  a. Working with predefine exception
  b. Working with custom exception
- Django ORM
- Django Cookies & Sessions implementations
- Django Custom Routing
- Django Image uploading
- Django file uploading

Django Rest Framework

- Why Django rest framework is required
- What is API
- What is Web API
- What is Rest-Ful API
- How to create restful API's using Django rest framework
- Working with postmen tool to test our restful API's

AngularJS

- Prerequest to learn angularjs
- What is angular JS
- Working with angular JS directives
- Working with one way and two data binding
- Working with Angular filters

Angular 14 Version

- Introduction to typescript
- What is typescript
- How to install and develop the typescript
- What is Transpolations
- Installing of NodeJS
- How to install angular framework
- How to create an application
- File and folder structure of angular application
- Working with one way data binding
  a. Interpolation data binding
  b. Property binding
  c. Class binding
  d. Style binding
  e. Event binding
- Working with Two-way data binding
- Working with custom component
- Integrating bootstrap in angular
- Working with *ngFor and *ngIf and *ngSwitch
- Working with predefine pipes
- Working with custom pipes
- Working with unit testing in angular
- Working with Routing in Angular

ReactJS

- Why react compare to another framework
- What is react
- Installing of ReactJS
- File & folder structure of react application
- Functional component in ReactJS
- Class component in ReactJS
- Working with Custom component
- Working with CSS in reactJS
- Working with Bootstrap integration
- Working with JSX
- What is state in ReactJS
- States using functional component
- States using class component
- Working with Props in ReactJS
- Working with Hooks in reactJS
- Working with Redux operations
- Working with MYSQL integration in ReactJS

Mysql or
Oracle Database

- ▶ Why database
- ▶ What is database
- ▶ What is SQL
- ▶ How to install MYSQL database
- ▶ Working with DDL commands

a. create command
b. alter command
c. drop command
d. rename command
e. truncate command

- ▶ Working with DML commands

a. insert command
b. update command
c. delete command
d. select command

- ▶ Working with constrains

a. primary key
b. foreign key
c. unique key
d. null key

- ▶ Working with order by clause
- ▶ Working with where clause
- ▶ Working with having clause

- ▶ Transactional commands

a. rollback
b. commit
c. save point

- ▶ Working with joins
- ▶ inner join
- ▶ outer join
- ▶ cross join
- ▶ full join

Flask

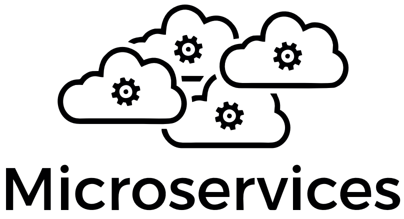
- Prerequest to Learn Flask
- What is Flask
- Why Flask compares to Django
- How to install flask
- How to create flask applications
- How to integrate routing in flask
- How to develop frontend development using flask
- How to connect database with flask

Flask with Flask restful with microservices

- What is flask restful
- What is microservice
- How to develop microservice based restful API'S
- Implementing the microservice using flask restful

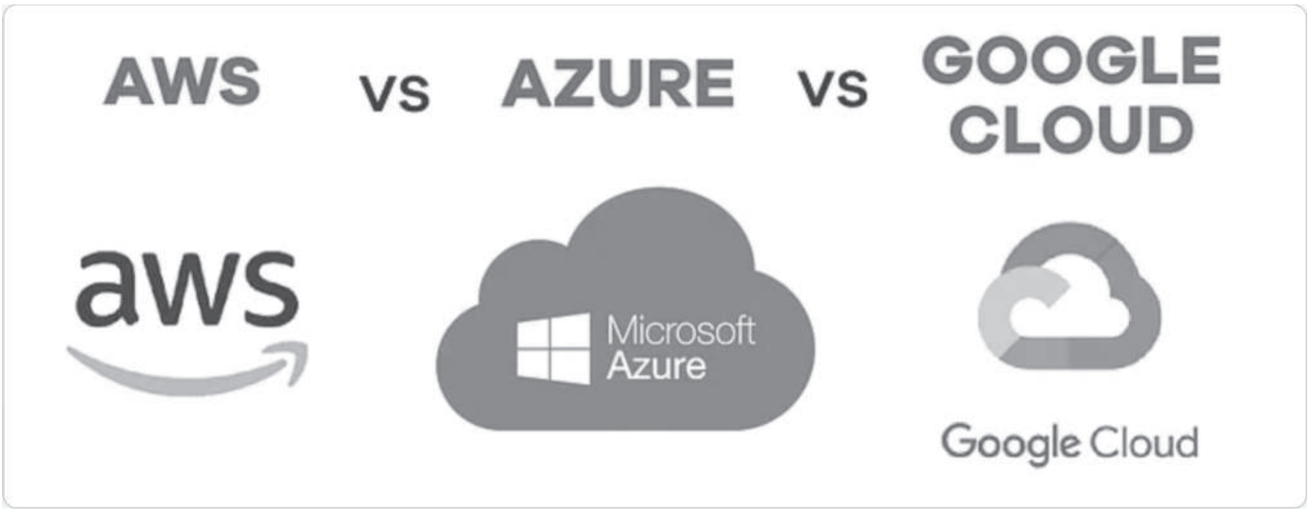
FastAPI with Microservices

- What is Fast API
- Why FastAPI compare to flask restful & Django rest framework
- Develop the restful APIS with FastAPI with mongo DB implementation



Cloud with Python integration

- AWS with Python integration



Artificial Intelligence(AI) with open AI (Tools) & NLP & Data Structures & Algorithms

- Applications of AI
- Histroy of AI
- Types of AI
- Future of AI
- Type of Agents
    - A. Intelligent Agent
    - B. Agent Environment
    - C. Turning Test in AI

Unification in FOL

- Resolution in FOL
- Forward Chaining and Backward Chaining
- Backward Chaining and Forward Chaining
- Reasoning in AI
- Indective vs Deductive reasoing

Problem Solving in AI

- Search Algorithm
- Uninformed Search Algorithm
- Informend Search Algorithm
- Hill climbing Algorithm
- Means End Analysis

Adversarial Search

- Adversarial Search
- Minimax Algorithm
- Alpha-Beta Pruning

Knowledge Represent

- Knowledge based Agent
- Knowledge Representation
- Propositional Logic
- Rule of Inference
- First Order-logic

Uncertian Knowledge R

- Probabilistic Reasoning in AI
- Bayes Theorem in AI
- Bayesian Belief Network

AI (Artificial Intelligence)

- Example of AI
- AI Essay
- AI in Healthcare
- AI in Education
- AI in Agriculture
- AI in Engineering Applications
- AI and Robotics
- Scope of AI
- Agent of AI
- AI in Stock
- AI as services
- AI in Banking
- Cognative AI
- AI in Automative Industry
- AI in ManuFacturing Industry
- AI In Civil Engineering
- AI in Gaming Industry
- AI in HR
- AI in Medicine
- AI in Marketing

DSA (Data Structure & Algorithms)

- Indtruction to Data Structure
- Algorithms
- Perfomance Analysis
- Asymptotic Notations
- Arrays
- Structures
- Pointers
- Dynamic Memory allocation
- Stacks Implementation
- Stack Implementation using pointer (dynamic)
- Queues
- Double Ended queue (Deques)

Creating your
Secure Future
together!!

ihub
INNOVATIVE PLACEMENT SOLUTIONS
powered by
Quality Thought®

Our Recent Successful Candidates

6.1 LPA
Bharathi
Temenos Pvt Ltd

4.6 LPA
Yogesh
accenture

4.6 LPA
Charan
accenture

3 LPA
Chandra Shekar
Tech Mahindra

LPA
M Anusha
Potla Tech Solotion

4 LPA
Vamshi
Kapil IT Soft Solutions

6.5 LPA
Harsha
Sterling

4 LPA
Chetan
Exousia International

3.5 LPA
K Venkata Siva
truminds

3 LPA
Aliya Fatima
RG2IT Solutions

2.4 LPA
Swapna
V-Soft

3.2 LPA
Rithick Poojari
Vollmond IT

3.5 LPA
Shibani Choudhary
Wipro Technologies

4.6 LPA
Hemanth
Accenture

2.9 LPA
B Durga Prasad
EVERESTDX

2.4 LPA
Tejaswini
V-Soft

2.9 LPA
Naveen Bhukya
EVERESTDX

3.5 LPA
Khasim
V-Soft

6 LPA
Surya Kumar
Sonatafy Tech

2.4 LPA
G Nalini
Hanya Auto Technologies

4.8 LPA
Manikanta
Service Pack

7 LPA
Bittu Kumar
NavSoft

4.7 LPA
Feroz
Codetrue

3.5 LPA
Wasim
V-Soft

2.4 LPA
S Sai Kumar
V-Soft

3 LPA
Hanna Marry
Suvarna Tech

3 LPA
Aditya
Quest global

4.8 LPA
Veerababu
Perenial Codeit Solution

4.5 LPA
Chandu Teja
Truminds

14 LPA
Pakirappa
Qualcomm

ihub
INNOVATIVE PLACEMENT SOLUTIONS
powered by
QualityThought®

Creating your
Secure Future
together!!

RamanaSoft Offices
Ameerpet **Madhapur**

Our Students Are Placed In

tcs TATA CONSULTANCY SERVICES	Capgemini	LTIMindtree	accenture	66degrees	AFFINE	High Noon Consulting
AGILISIUM	ALTIMETRIK	brillio	CGI	cisco	Cloud4C	amazon
CLOUD9	CloudCover	Coforge	Cognizant	Convergytics	CSS CORP	Medtronic
databricks	dataflex	DXC TECHNOLOGY	Deloitte.	EMIDS	Foray	STIC Soft
genpact	HCL	HSBC	i2D	IBM	INFINITY DESIGN	CAPRUS IT
Infosys	Legato	MINDGRAPH	Mindtree	motivitylabs Innovation as a Service	photon	meslová
pwc	quantiphi	RENAULT Passion for life	SIGMASOFT Virtual Molding	SOFTILITY	syren cloud	splashBI
TIGER ANALYTICS	wipro	ENVISION	virtusa	ValueLabs	zensor	SOLIX
JRD SYSTEMS	ValueLabs	archents	magnasoft	Innominds Powering the Digital Next	techforce.ai	techolution
SOLUGENIX	Nendrasys	S&P Global	Aadvi	iPrism Technologies	ENVISION	Sunseaz Technologies Pvt Ltd

QualityThought
✆ 81437 41509

Quality Thought Infosystems India (P) Ltd.

#302, Nilgiri Block, Ameerpet, Hyderabad-500016 | www.qualitythought.in | info@qualitythought.in