MEDHA EDU TECH

educate elevate empower

Master the World of
Web Development

Full Stack Python with ReactJS, Cloud & AI

MET –

Transforming Learners
into Python Innovators

MSME
MICRO, SMALL & MEDIUM ENTERPRISES
सूक्ष्म, लघु एवं मध्यम उद्यम
OUR STRENGTH • हमारी शक्ति
Ministry of MSME, Govt. of India

CERTIFIED ISO 9001:2015 COMPANY

Google Partner

#startupindia

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Get Job Ready with our

7.7.7 CURRICULUM WITH 49 PROJECTS

TILL YOU COMPLETE THE PROGRAM.

Get Experienced to AI Assisted Mock Interviews to make you Job Ready

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Join Free Demo & Pre-Boot Sessions & Learn to Create your First App in just 3 Days!

LEARNING PATH

Mon-Fri

1-hour class,
1.5-hour hands-on practice,
project work, &
practical learning,
soft skills.

Saturday

Revision,
1-on-1 mentorship,
mock interviews,
soft skills &
industry insights.

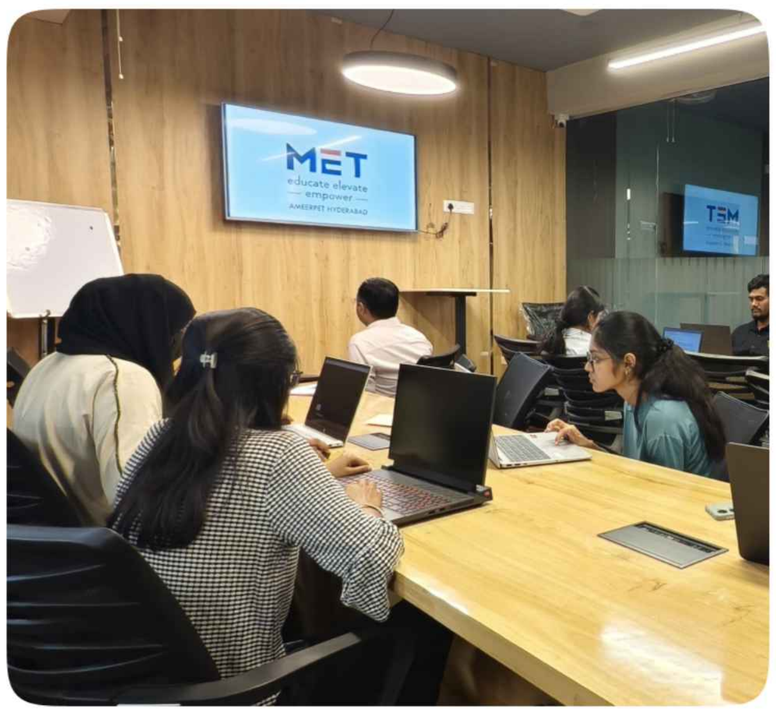
+91 8341570179 www.medhaedutech.com info@medhaedutech.com

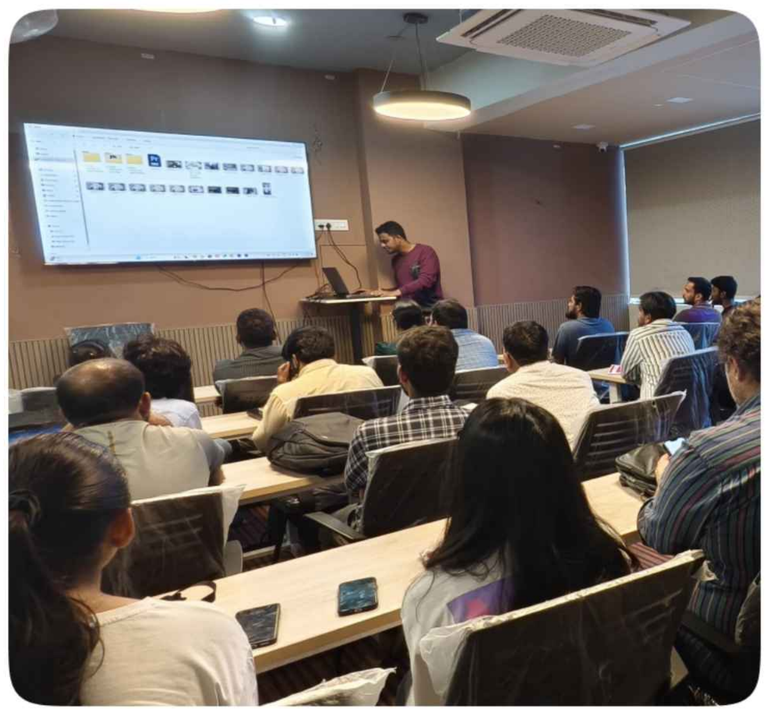
About MET

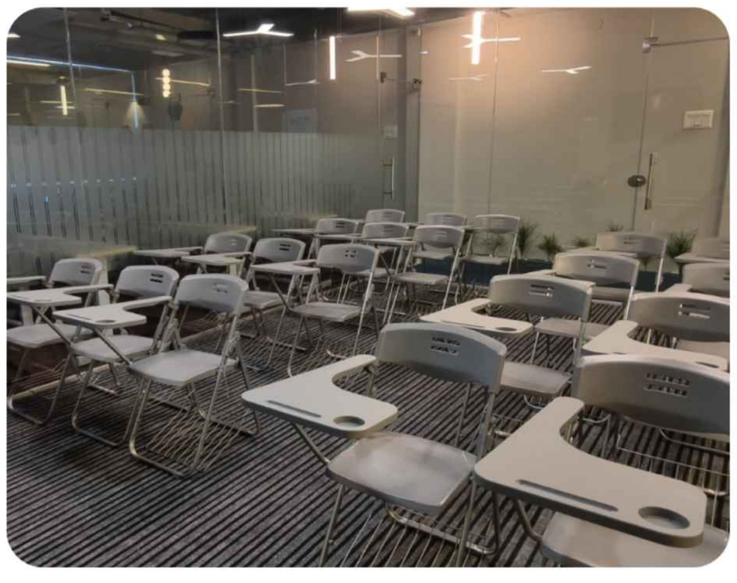
At Medha Edutech, we take pride in delivering top-tier education through exceptional instructors with extensive real-world expertise. Our comprehensive programs in Full Stack Development, Data Science, Generative AI, Cyber Security, and UI Development are meticulously crafted to meet the evolving demands of the tech industry. Having successfully upskilled over 20,000 students across 15+ countries, we remain committed to excellence in education.

We are committed to empowering learners with practical knowledge through hands-on training and live projects. Guided by industry-experienced trainers, our courses ensure a seamless blend of theory and real-time application. Whether you aim to enhance your skills, transition into a new career, or embark on your professional journey, Medha Edutech stands as your trusted partner in achieving success. Recognized for the ***India Excellence Awards 2024,*** we continue to set benchmarks in quality education. As a leading software training academy, we are dedicated to making learning accessible, impactful, and transformative.

World-class infrastructure on campus brings the feel of an IT company







Monthly Hackathons + Career Accelerator Workshops

BUILD. COMPETE. GROW.

Level up with high-impact, course-aligned hackathons every month.

Solve real-world challenges, race the clock, and stand out from the crowd.

Get live mentorship, showcase your skills, and push your limits.

WIN BIG

Build a **Strong Portfolio,** Certificates, Internships & Exclusive Rewards.

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Why Choose MET

At MET, we turn learners into achievers with hands-on training, certifications, Placements, and global opportunities!

- Instructors with vast Software Industry Expertise
- Access free spoken english, Aptitude & Reasoning Classes
- Transform yourself with Personality Development Classes
- Get Hands on Projects
- Regular Assignments and Evaluations
- Weekly Tasks and Skill Assessments
- Practical and Industry-Centric Curriculum
- 24/7 LMS Access
- Low-Cost Fee Structure
- Extensive Practical Learning
- CV/Resume Development Support
- Professional Portfolio Guidance
- Personalized Mentorship
- Weekly Interview Simulations
- Premium International Standard Campus

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Your Smart Learning Partner Medha Academy LMS

LMS LIKE NEVER BEFORE

Built for the Future

✓ Get **auto-uploaded recorded sessions** daily after your class.

✓ Access your **entire course library** from anywhere, anytime.

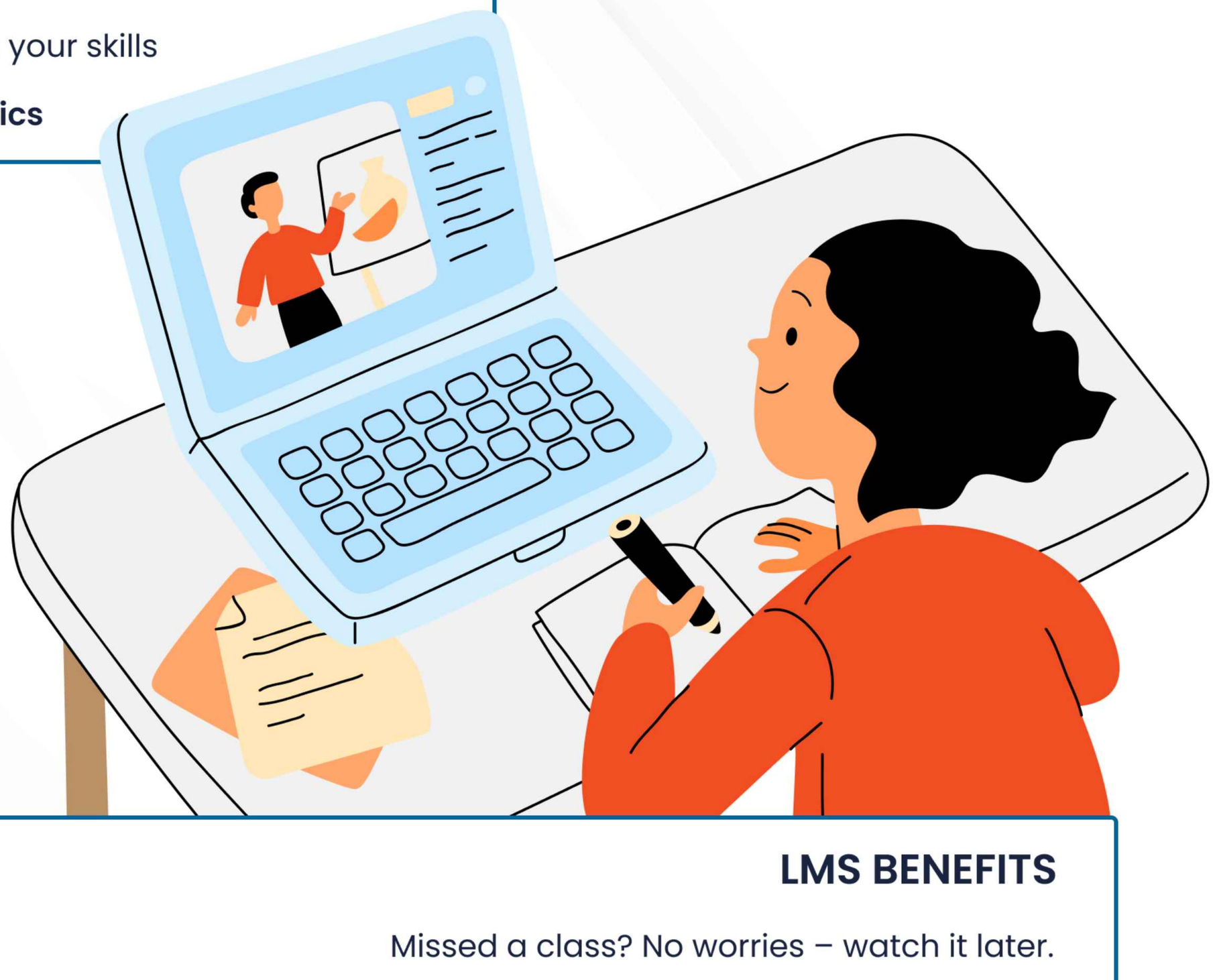
TRACK YOUR PROGRESS WITH

Assignments & materials organized by modules

Topic-wise quizzes

Assessments to test your skills

Performance analytics



LMS BENEFITS

Missed a class? No worries – watch it later.

All your learning is in **one place.**

Prepares you better for interviews and projects.

Suitable for **working professionals, students,** and **career switchers.**

Learning isn't limited to the classroom anymore. ***With MET LMS,*** *the classroom follows you.*

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Who Can Learn Full Stack Python?

Beginners(Engineering freshers)

Start your coding journey with Python's easy-to-learn syntax.

Web Developers

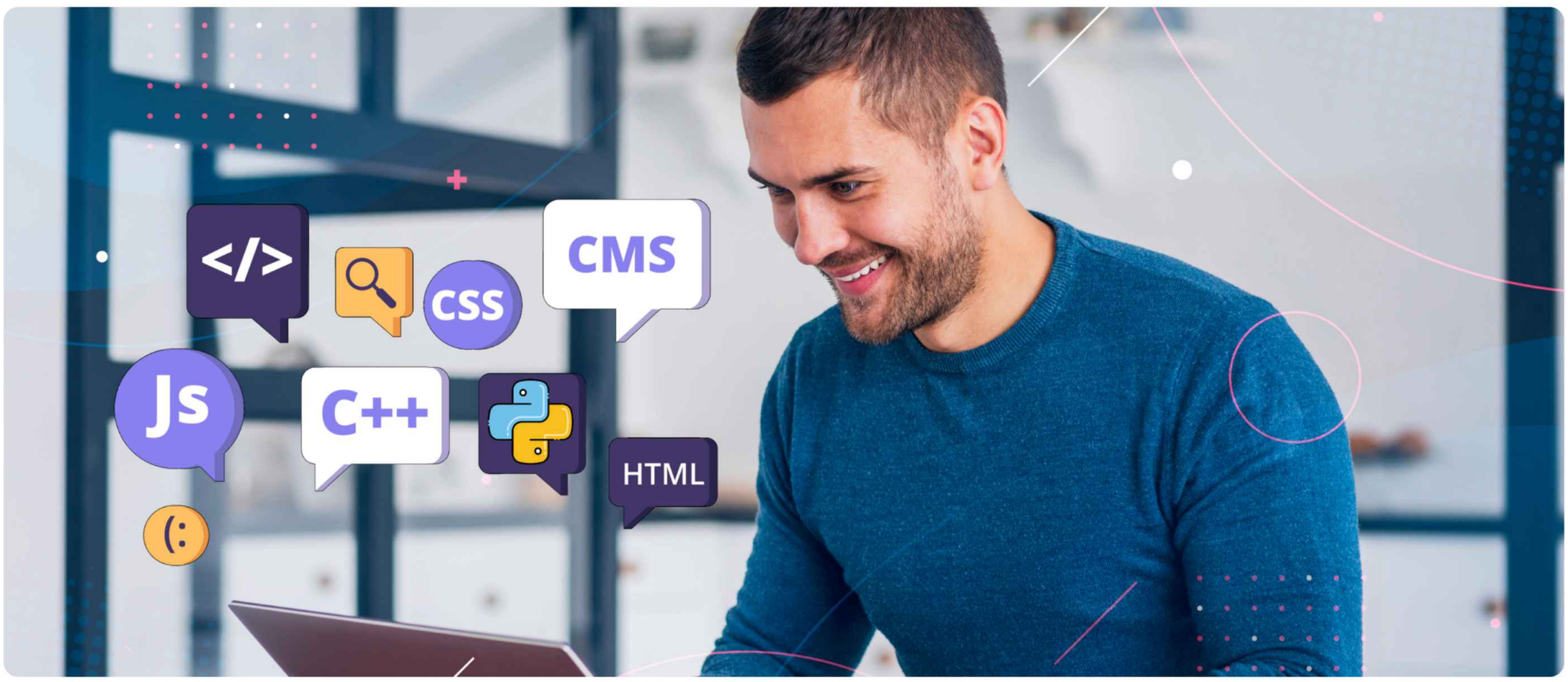
Advance your skills by mastering both front-end,back-end Python frameworks and data layer technologies.

Data Enthusiasts

Python is key to excelling in Data Science and Machine Learning.

Professionals & Freelancers

Add Python to your skill set to unlock new freelance and remote work opportunities.



+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Our Alumni Work At

At Medha EduTech, we take immense pride in the success of our students who have gone on to build thriving careers with leading companies across the globe. Our alumni have secured prestigious roles in top organizations, including **Google, Microsoft, Amazon, Deloitte, Infosys, Wipro, TCS, Cognizant, Capgemini, Accenture, Byju's, Zomato, Swiggy,** and many more.

By fostering a strong foundation in both technical and professional skills, we ensure our students are job-ready and capable of making an impact in the dynamic world of technology. Join Medha EduTech and step into a future filled with possibilities!

Companies Hiring Full Stack Python Developers

Google | Microsoft | amazon | IBM

facebook | accenture | TATA CONSULTANCY SERVICES | cognizant

verizon | Adobe | McKinsey&Company | Deloitte.

SAMSUNG | genpact | Fidelity INVESTMENTS | BARCLAYS

Gartner | Tech Mahindra | KPMG | Capgemini

HCL | Honeywell | AON Hewitt | wipro

Thomson Reuters | MAERSK | Mu Sigma DO THE MATH | CISCO

pwc | citigroup | EY | DELL

AMERICAN EXPRESS | CSC COMMON SERVICE CENTER | इसरो isro | HAVELLS

In-Demand Roles

Full Stack Developer

Back-End Developer

Automation Engineer

Finance & Trading Analyst

Data Scientist /Data Analyst

DevOps Engineer

& more

Industries Benefiting from Full Stack Python

IT & Software Development

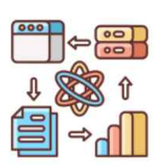
FinTech (Financial Technology)

E-commerce & Retail

Healthcare & Medical Tech

Media & Entertainment

Automotive & Transportation



Data Science & AI

Cybersecurity

Government & Public Services

EdTech (Education Technology)

About Full Stack Python & Its Importance

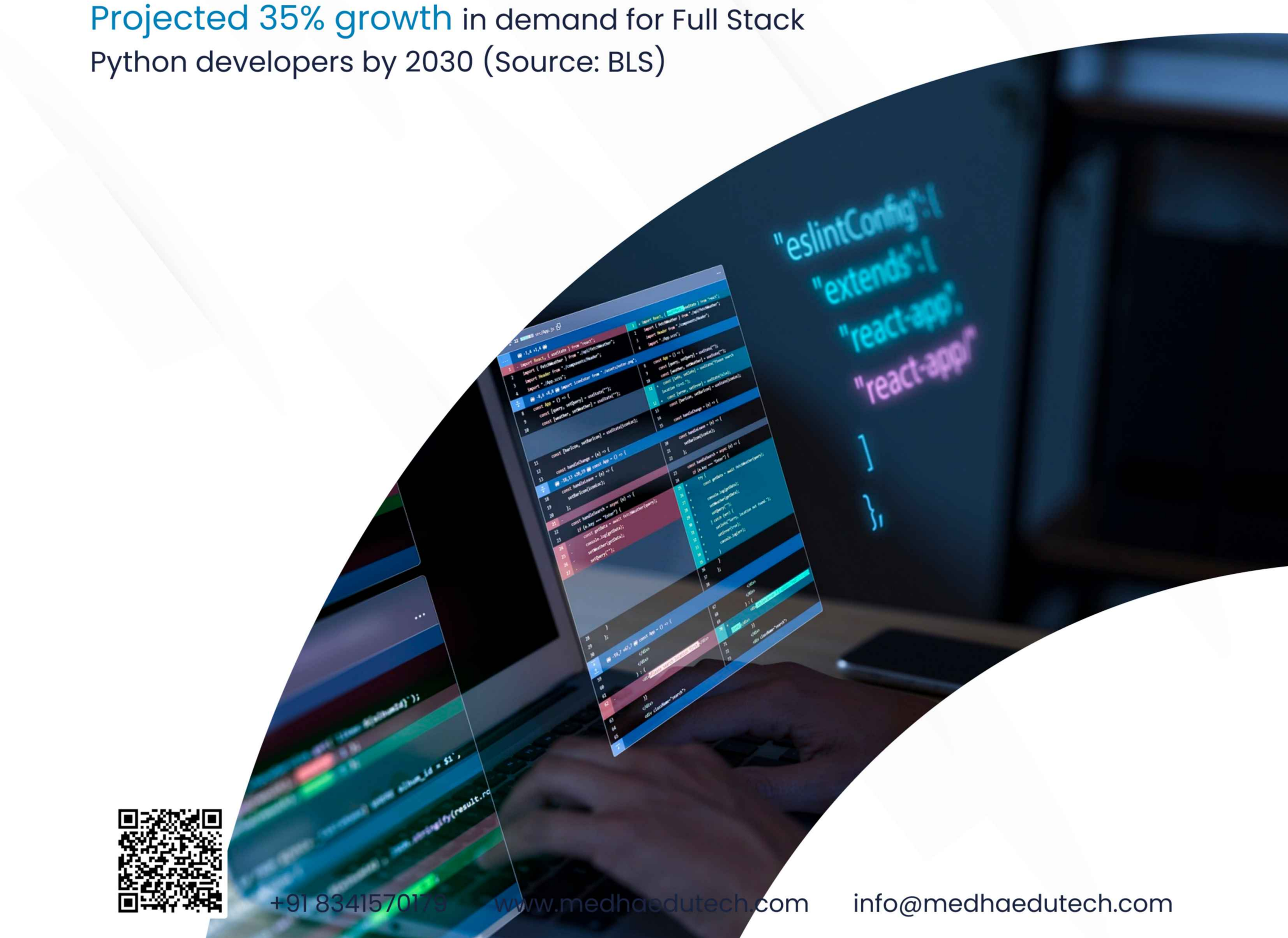
Full Stack Python is a highly versatile and in-demand skill that enables developers to work on both frontend and backend development. It integrates powerful frameworks like Django and Flask with front-end technologies, making it ideal for building scalable and dynamic web applications. With Python's simplicity and vast ecosystem, businesses across industries leverage it for web development, automation, AI, and cloud computing.

Present & Future of Full Stack Python (Stats & Trends)

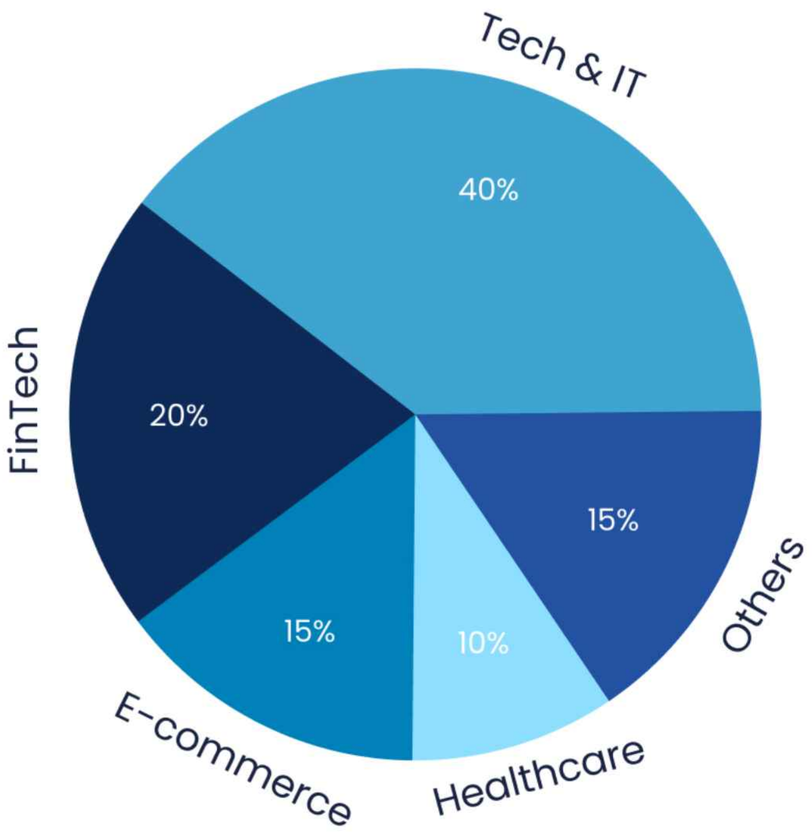
Below is a structured graphical representation of key statistics

Global Demand for Full Stack Python Developers (2024-2030)

Projected 35% growth in demand for Full Stack Python developers by 2030 (Source: BLS)



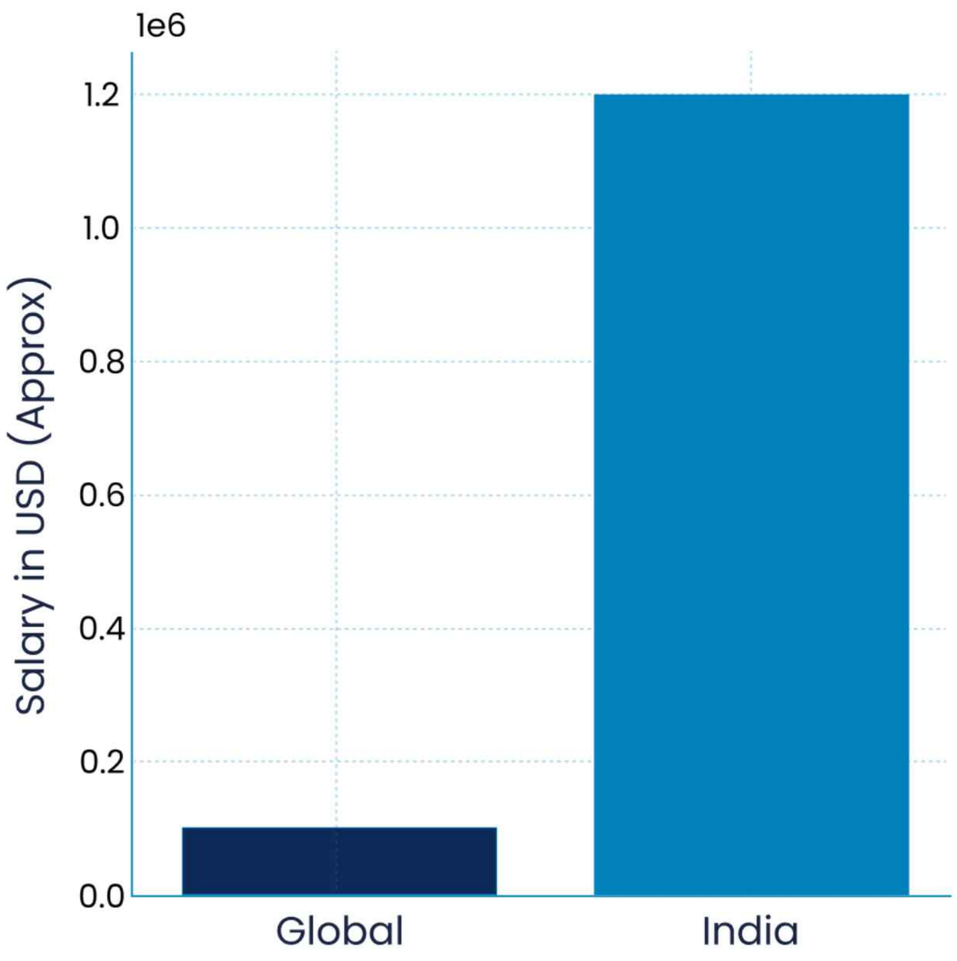
Average Salary of Full Stack Python Developers



Industry Adoption of Full Stack Python

A pie chart showing how different industries like Tech, FinTech, E-commerce, and Healthcare are leveraging Full Stack Python.

Average Salary of Full Stack Python Developers



Average Salary of Full Stack Python Developers

A bar chart comparing global salaries with those in India.

Python's Ranking in Popularity

#1 Programming Language in 2024 (Source: TIOBE Index)

Tools and Technology covered

HTML | CSS | Javascript | React Js | Python | Postman | NoSQL

Git | Vercel | Tailwindcss | Django | PostgreSQL | streamlit | AWS

supabase | Pandas | Plotly | redes | sqllite3 | Data Visualization | Cassandra

ollama | NLP | rabbit MQ | Vscode | replit | pinecone | selery

github desktop | source tree | make.com | zapier | IFTTT | datadog | Generative AI

Git | Jupyter | Google Colab | Pycharm | Visual Studio | GitHub actions | AGgrid

No SQL | Kaggle | OPENAI & API | hugging face | jinkins | docker | sonar cube

couldflare | list io | pytest | cloud watch | Aws Devops | SQL Server

Syllabus Structure – Full Stack Python with Django, ReactJS, and AWS

Python

- Introduction to Python
- Data Types and Variables
- Conditional Statements and Loops
- Functions and Lambda Expressions
- Lists, Tuples, Sets, and Dictionaries
- File Handling
- Exception Handling
- Object-Oriented Programming (OOP)
  - Classes and Objects
  - Inheritance, Polymorphism, Encapsulation
- Modules and Packages
- Python Libraries (NumPY,Panda,StreamLITetc)
- Debugging and Testing

Fast API

1. Introduction to FastAPi
   - What is FastAPI?
   - Key features and advantages over Flask/Django
   - Setting up a FastAPI project
2. Building API Endpoints
   - Defining routes with @app.get(), @app.post(), etc.
   - Path and query parameters
3. Database Integration
   - SQLAlchemy with FastAPI
   - Dependency injection for database sessions
   - CRUD operations with async database calls
4. Authentication & Authorization
   - JWT-based authentication
   - OAuth2 with password flow
   - Role-based access control
5. Production-Ready FastAPI
   - Middleware and CORS handling
   - Background tasks and WebSockets

Django

- Introduction to Framework
- Project Creation
- URL Dispatcher
- Templates and Static Files
- Models and CRUD Operations
- Forms and Form Validation
- ORM (Object-Relational Mapping)
- Admin Application
- Authentication and Authorization
- Class-Based Views
- Middleware and Signals
- Security and Session Handling
- Pagination and Caching

Syllabus Structure – Full Stack Python with Django, ReactJS, and AWS

DjangoREST

- Overview of API and REST
- CRUD Operations with API
- Serialization and Deserialization
- Functional-Based and Class-Based Views
- Authentication (Token, JWT)
- Filtering, Ordering, and Pagination
- Request and response models using Pydantic

Cloud AWS

- AWS Infrastructure Setup
- Deploying FastAPI Backend
- Database and Storage
- Deploying Frontend with Jinja2
- CI/CD and Scaling

FULL STACK DEVELOPMENT COURSE BREAKDOWN

Frontend
- HTML
- CSS
- JavaScript
- React JS
- Streamlit
- Jinja2

Backend
- Django, DRF
- FastAPI

Database
- MySQL
- PostgreSQL

Cloud
- Vercel /AWS

Version Control
- Git

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Capstone Project

AI Integration

Work on advanced projects involving GPT, Claude, Gemini, DeepSeek, and other cutting-edge AI models

Full-Stack AI Solutions

Build intelligent applications like chatbots, AI assistants, and content generators using Python and modern frameworks.

Dynamic API Integration

Learn to seamlessly connect and manage multiple AI APIs for diverse use cases.

Scalable Systems

Implement robust backend systems with Python while delivering intuitive frontend experiences.

Industry-Driven Innovations

Stay ahead with monthly project updates reflecting the latest trends and business needs.

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

Python 3.x

1. Basic understanding of Python
2. Python and Other programming languages
3. Difference between python2.x and 3.x
4. Python Environment
   a. Windows
   b. Linux
   c. Mac os
5. Python first program
6. Python Internals
7. Compilers v/s interpreters
8. Different interpreters and IDE's
   a. Interpreters
      i. Python
      ii. I python
      iii. I python Notebook (Jupyter)
      iv. Bpython
      v. Bpython Notebook
   b. IDE's
      i. Spyder
      ii. Pycharm
      iii. Vscode
      iv. Notepad++
      v. Sublime text
9. Data types and variables
   a. Object reference
10. Python Keywords
11. Operators
    a. Relational
    b. Comparison
12. Statements of python
13. Working with Numbers
14. Python Escape sequence characters
    a. 16 bit Unicode
    b. 32 bit Unicode
    c. String literals
15. Working with strings
    a. Indexing
    b. Slicing
16. Type Casting
    a. % with positional
    b. {} with index
17. Type Conversion
18. Conditional Statements
    a. If …
    b. If … else..
    c. If …. Else… if.. else …
    d. If … elif…else
    e. Nested Functions
19. Control Statements
    a. Pass
    b. Continue
    c. Break
20. Loops
    a. For (Finite loop)
    b. While (Infinite loop)
21. Python Sequences
    a. List
       i. List creation
       ii. Indexing
       iii. Slicing
       iv. list methods

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

b. Tuple
  i. Tuple creation
  ii. Tuple to list
  iii. List to tuple
  iv. Indexing
  v. Slicing
  vi. Packing and unpacking
  vii. Tuple methods

c. Dictionary
  i. Dictionary creation
  ii. Dict with list values
  iii. Dict with tuple values
  iv. Dict with dict values
  v. Dict with comprehensions
  vi. Dictionary methods

d. Set
  i. Sets creation
  ii. Union , union all
  iii. Intersection
  iv. Minus
  v. Set methods

22. Copy
  a. Soft copy
  b. Shallow copy
  c. Deep copy
  d. Copy of elements
  e. Copy of objects
  f. Copy of sub-objects

23. Working with Functions
  a. Over view of Function
  b. Function creation
  c. Return v/s print
  d. parameters
  e. Arguments
  f. Default arguments
  g. Positional arguments
  h. Keyword arguments
  i. Variables
  j. Local variables
  k. Locals()
  l. Global variables
  m. Globals()
  n. Calling local variables globally
  o. Function object
  p. * , ** , *args , **kwargs
  q. Nested Functions

24. Working with Python Built-in Functions
  a. Len, max, range, min, sum
  b. Ascii, bin, ord
  c. Locals(), globals()
  d. Setattr(), getattr(), delattr()
  e. Oct(),hex(),complex()
  f. divmode
  g. any, all
  h. frozenset()

25. Working with Recursions
  a. Calling function it self

26. Working with Exceptions
  a. Try
  b. Except
  c. Else
  d. Finally
  e. Built-in exceptions

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

f. User defined exception
  i. exception creation
  ii. Class inheritance from
  Base class

27. Working with Regular Expressions
    a. Match
    b. Search
    c. Compile
    d. Split
    e. Group
    f. Full match
    g. Sub
    h. Find all
    i. Finditer
    j. escape
    k. Special characters
28. Working with Modules
    a. Creating modules
    b. Importing module
    c. Access module functions
29. Working with Files
    a. Creating files
    b. Data migration
    c. Database to files
    d. Read, read line , read lines
    e. Write, write lines
    f. Append
    g. Seek
    h. Tell
    i. Flush
    j. Writable, readable, seekable

30. Working with CSV Files
    a. Import csv
    b. Csvreader
    c. Csvwriter
    d. csv
31. Working with Excel Files
    a. Import xlrd
    b. Import openpyxl
32. Working with Databases
    a. Oracle
    b. PostgreSQL
33. User Defined Packages
    a. __init__.py
    b. Sub-packages
    c. Modules
    d. Importing UD Packages
34. class
    a. Class attributes
    b. Methods
    c. Special methods
    d. Method object for a class
    e. Class method
    f. Meta class
35. oops
    a. Inheritance
      i. Single
      ii. Multiline
      iii. Multiple
      iv. Hierarchical
      v. Hybrid
  b. Polymorphism
    i. Method Overloading

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

1. pythonlangutil
2. @overload
3. @signature

ii. Method Overriding

1. Super()
2. Constructor overriding

iii. Operator overloading

c. Encapsulation

i. Public attributes
ii. Private attributes
iii. Public methods
iv. Special methods
v. Private methods

d. Data Abstraction

i. Abstract class
ii. Abstract methods

36. @class method
    a. Object with class method
37. @Static method
    a. Object method creation
38. Python stack
    a. Put
    b. Get
    c. Put_nowait
    d. Get_nowait
39. Python debugging
    a. Next
    b. Where(w)
    c. Continue (c)
    d. exit
40. Python list comprehension
    a. Comprehension to create list
    b. Comprehension to create generator
41. Collections
    a. Deque
    b. Queue
        i. Fifo queue
        ii. Lifo Queue
        iii. Priority Queue
    c. DefaultDict
    d. ordereddict
    e. Namedtuple
    f. counter
42. Multithreading
    a. Thread creation
    b. Thread start
    c. Thread join
43. Multi-Processing
    a. Process creation
    b. Process start
    c. Process join
    d. Sub-Process
44. Different Python Functions
    a. Enumerate
    b. Lambda
    c. Map
    d. Filter
    e. Reduce
    f. Iterator
    g. Yield
    h. Generator
    i. Closures
    j. Decorators

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

45. Different Python packages (Overview)
    a. os
    b. math
    c. cmd
    d. random
    e. operator
    f. Sys
    g. Itertools
    h. Date time
    i. Time
    j. Calendar
    k. Py_compile
    l. Compileall

46. Logging
    a. Logging levels
        i. Not set - 0
        ii. Debug - 10
        iii. Info – 20
        iv. Warning – 30
        v. Error – 40
        vi. Critical – 50
    b. Logger
        i. Stream handler
        ii. File handler
        iii. SMTP handler

Fast API

1. Introduction to FastAPI
   - What is FastAPI?
   - Why choose FastAPI over Flask/Django?
   - Key features: Type hints, async support, automatic documentation
   - Installing FastAPI and Uvicorn
   - Setting up a FastAPI project
   - Running the first FastAPI app (hello world)

2. Understanding FastAPI Routing
   - Defining routes using @app.get(), @app.post(), @app.put(), @app.delete()
   - Path parameters and type validation
   - Query parameters and defaults
   - Request bodies with Pydantic models
   - Handling form data and file uploads
   - Response models for structured API responses

3. Dependency Injection in FastAPI
   - What are dependencies?
   - Creating and using dependencies

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

- Common use cases: database sessions, authentication, logging
- Using Depends() for cleaner code

4. Data Handling with Pydantic
   - Introduction to Pydantic models
   - Field validation and constraints (min_length, max_length, regex)
   - Nested models and custom validation
   - Data serialization and response customization
   - Enforcing data types and handling missing fields
5. Database Integration
   - Introduction to SQLAlchemy
   - Setting up an async database with Databases library
   - Connecting FastAPI to PostgreSQL/MySQL
   - Creating tables and defining ORM models
   - CRUD operations (Create, Read, Update, Delete)
   - Using Alembic for database migrations
6. Authentication & Authorization
   - Introduction to authentication in FastAPI
   - Using OAuth2 with password flow
   - Implementing JWT authentication (fastapi.security)
   - Protecting routes with Depends()
   - Role-based access control (RBAC)
   - API key authentication
7. Background Tasks & WebSockets
   - Running background tasks with BackgroundTasks
   - Sending emails or processing data asynchronously
   - WebSockets for real-time communication
   - Building a simple chat application
8. Middleware, CORS, and Security
   - What is middleware in FastAPI?
   - Logging and request processing with middleware
   - Handling CORS (CORSMiddleware)

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

- Security best practices: HTTPS, headers, rate limiting
- Protecting against SQL injection and CSRF attacks

9. Testing FastAPI Applications
   - Why testing is important
   - Writing unit tests with pytest
   - Testing API endpoints with TestClient
   - Using dependency overrides for mock testing
   - Load testing with locust
10. Deploying FastAPI Applications
    - Deployment strategies: Docker, AWS, DigitalOcean, etc.
    - Using Gunicorn and Uvicorn for production
    - Setting up Nginx as a reverse proxy
    - CI/CD pipelines for FastAPI (GitHub Actions, GitLab CI)
    - Scaling FastAPI with Kubernetes
11. Advanced Topics
    - GraphQL with FastAPI (strawberry-graphql)
    - Asynchronous task queues with Celery and Redis
- Streaming responses with Streaming Response
- Handling large file uploads efficiently
- Using FastAPI with event-driven architecture (Kafka, RabbitMQ)

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

DJANGO

1. Understanding of Framework
2. Why is Framework?
3. What is Web App Framework?
4. What is MVC Architecture?
5. Understanding of Django Web App Framework
6. Pre-requisites
7. Django Setup
   - Virtual Environment
   - Python installation
   - Django installation
   - Uninstallation of Django and Python
   - VS Code
8. Extensions in VS Code for development
9. Django Project Creation
10. Understanding Django Project Directory
    - Settings.py
    - Aswi.py
    - Wsgi.py
    - Manage.py
    - Urls.py
11. Creating Application in Django Project
12. Understanding the files in Application
13. Start and Stop Development Server
14. Understanding Request and Response
15. Function-Based Views
16. Understanding of MVT Architecture
17. URL Dispatcher or URL Patterns in Project Directory
    - Path()
    - Name
    - Template
18. Creating Multiple Applications
19. Dynamic URL
20. Custom Path Converters
21. URL Dispatcher or URL Patterns in Application
    - Include
    - Namespace
    - App Registration
22. Understanding Template in Django Project
    - Render
    - HttpResponse
    - Request
23. Template Folder Structure
24. Setup Template Path in Settings.py
    - Templates Directory
    - Installed_Application
    - Template Directory
25. Setup Static Files in Settings.py
    - staticfiles_dirs
    - load static
    - static with URL

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

26. Rendering Templates with Context
27. Application-Wise Template Rendering
28. Writing Template Files
29. Injecting CSS Styles Inside Templates
30. Understanding the DTL Inside Template
    - {{ variable }}
    - {% for %}
    - Pre-defined for Loop Variables
    - {% if %}
    - Filters
    - {% include %}
31. Dynamic Template Files
32. Templates Inside Application
33. How to Use JavaScript Inside Application
34. Template Inheritance
    - Extends
    - Block and End Block
    - Base Template
    - Parent Template
    - Child Template
35. Template Inheritance with Static Files
36. Using Bootstrap in Django Templates
    - Bootstrap.js
    - jQuery.js
    - Popper.js
37. Model
    - Create Super User
    - Migrate/Create Database
    - Create Model
    - Register Model
    - Create Query Object
    - CRUD Operations
39. Django Admin Application
    - Create User
    - Create Group
    - Create Permission
    - Assign Permission
40. Forms
    - Model Form
    - Model Form Inheritance
    - Forms API
    - From Django import Forms
    - Configure Attributes
    - ID, Label, Ordering
    - Form Widgets
    - Get and Post Methods
    - CSRF
    - Get Form Data and Validate
    - Post Form Data to Server
    - HttpResponseRedirect
41. Form Fields
    - Cleaning and Validating Form Field
    - Validating Complete Form Data at a Time
    - Built-in Validators and Custom Validators

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

- Match Password and Re-enter Password Field
- Form Errors and Field Errors
- CRUD Operations with Forms

42. ORM
    - Queryset
        - All
        - Get
        - Filter
        - Query
        - Values
        - Values_list
        - Union
        - Intersection
    - Queryset API Field Lookup
    - Aggregation
    - Aggregation Functions
    - Query Objects
    - Model Inheritance
    - Model Manager
43. Model Relations
    - One-One
    - Many-to-One
    - Many-to-Many
44. Messages Framework
    - Level
    - Tag
45. CRUD Project with Function-Based Views
46. Authentication and Authorization
    - User Authentication System
    - Django.contrib.auth
    - Creating User
    - Changing Password
    - Authenticating User
    - Permissions to Users
    - Creating Groups
    - Login as User
    - Logout
    - Abstract User
    - Auth_User
    - Create Registration Form
    - UserCreationForm
    - User Model
    - Create Login Form
    - Authenticate()
    - Login()
    - Logout()
    - AuthenticationForm()
    - Is_authenticated
    - Change Password
    - User Profile
    - Admin Profile
    - Profile Changing
47. Authorization and Permissions
48. Pagination with Function-Based View
49. New Features in Django 4 Version

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

50. Cookies
51. Session Framework
52. Page Session Expiration
53. File-Based Session
54. Counting Page and Login
55. Cache and Per Site Cache
56. Cache Per View
57. Template Fragmentation
58. Low-Level Cache
59. Signals
60. Built-in Signals
61. Custom Signals
62. IP Tracking
63. Middleware
    - Built-in Middleware
    - Custom Middleware
    - Function-Based Middleware
    - Get_response()
    - Middleware Hooks
    - Template Response
    - UnderConstructionMiddleware
64. Class-Based Views
    - View
    - TemplateView
    - RedirectView
    - FormView
    - ListView
    - CreateView
    - DeleteView
    - UpdateView
    - DetailView
65. Function-Based Authentication Views
    - Login_required
    - Customization
66. Authentication Settings
67. Database Configuration
    - PostgreSQL
    - MongoDB
68. Pagination with Class-Based Views
69. Security in Django

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

DjangoREST

1. Overview of API
2. Why API in Web Development?
3. What is Web API?
4. How API Works?
5. What is REST and RESTful API?
6. Understanding CRUD Operations
7. Overview of End Point
   - Base URL
   - Naming Conversion
   - End Point
   - Request
   - Response
8. RESTful Methods
   - GET for All Records
   - GET for Specific Record
   - POST to Store Data
   - PUT for Update
   - PATCH for Update
   - DELETE for Remove
9. Overview of DjangoRESTFramework
   - Setup DRF
   - Install DRF
   - Uninstall DRF
   - Freeze
10. Serialization
    - Serializer Fields
    - Complex Data
    - Python Native Data
    - JSON-dumps
    - JSON-loads
    - Serializer Class
    - @csrf_exempt
    - Deserialization
    - Create
    - Update
    - Delete
    - Model Serializer
    - JSONRenderer
    - JSONResponse
    - HttpResponse
    - Request.get
    - Request.post
    - Request.put
    - Request.delete
    - Arguments
    - Bytes-IO
11. CRUD API
    - Functional Based API View
      - @api_view
    - Class-Based API View
      - APIView
    - JSONParser
    - CSRF_Exempt
    - Method Decorator
12. Validations
    - Field Level
    - Object Level
    - Validators
    - Validation Error
13. Concrete View
    - ListAPIView

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

- CreateAPIView
- RetrieveAPIView
- UpdateAPIView
- DestroyAPIView
- ListCreateAPIView

14. ViewSet
    - List()
    - Retrieve()
    - Create()
    - Update()
    - Partial_Update()
    - Destroy()
15. ModelViewSet Class
16. Basic Authentication
17. Permission Class
18. Session Authentication
19. Authentication in Function-Based View
20. Token Authentication in DRF
21. Custom Authentication
22. JSON Web Token Authentication
    - RemoteUserAuthentication
23. Throttling
    - AnonRate Throttling
24. Filtering
25. Ordering
26. Pagination
    - Cursor Pagination
    - Limit Offset
    - Page Number
27. Relations in Serializers
28. Hyperlinked Model Serializer
29. Nested Serializer

GitHub

- Working with GitHub Repositories
- GitHub Workflow & Version Control
- Branching & Collaboration
- Advanced GitHub Features
- GitHub for AI & Machine Learning Projects
- Security & Best Practices

Cloud AWS

1. AWS Infrastructure Setup
    - Overview of AWS services needed: EC2, S3, RDS, Route 53, ACM (SSL)
    - Setting up an EC2 instance (Ubuntu/Amazon Linux)
    - Configuring security groups and firewall rules
    - Setting up an Elastic IP for a static public IP
2. Deploying FastAPI Backend
    - Installing Uvicorn, Gunicorn, and Nginx on EC2
    - Running FastAPI as a systemd service
    - Setting up a reverse proxy with Nginx
    - Using Let's Encrypt SSL for HTTPS

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

- Managing environment variables securely

3. Database and Storage
   - Setting up PostgreSQL/ MySQL on AWS RDS
   - Connecting FastAPI to RDS securely
   - Using AWS S3 for static file storage
   - Configuring FastAPI to serve static and media files
4. Deploying Frontend with Jinja2
   - Structuring FastAPI to serve Jinja2 HTML templates
   - Managing static files (CSS, JS, images) with Nginx
   - Enabling server-side rendering (SSR) with FastAPI
   - Optimizing page load speed and caching
5. CI/CD and Scaling
   - Automating deployments with GitHub Actions + AWS CodeDeploy
   - Using Docker + ECS (Elastic Container Service) for scalable hosting
   - Load balancing with AWS ALB (Application Load Balancer)
   - Implementing auto-scaling for high availability
   - Monitoring with AWS CloudWatch and Logs

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

Introduction to DSA & Python Syntax Refresher

- Arrays and Lists
- Operations: Insertion, Deletion, Traversal
- Problems: Max/Min, Reverse, Frequency count
- Strings
- Manipulation & Built-in Methods
- Problems: Palindrome, Anagram, Substrings
- Stacks & Queues
- Using Lists & collections.deque
- Applications: Expression evaluation, Balanced brackets
- Recursion Basics
- Factorial, Fibonacci, Sum of digits
- Introduction to Backtracking
- Searching & Sorting
- Binary Search, Linear Search
- Bubble, Selection, Insertion, Merge Sort
- Linked Lists
- Singly & Doubly Linked Lists
- Operations & Basic Problems
- Hashing
- Using Dictionaries & Sets
- Applications in Frequency Count & Lookups
- *Trees (Intro)*
- Binary Tree & Binary Search Tree
- Traversals: Inorder, Preorder, Postorder
- Greedy & Two Pointer Techniques
- Activity Selection, Pair Sum, Sliding Window Basics

1. HTML5 FUNDAMENTALS

Basic HTML Structure

- Document Type Declaration (DOCTYPE)
- HTML Document Structure (html, head, body)
- Meta Tags and Document Information
- Character Encoding and Language Attributes
- Page Title and Favicon Integration

HTML Elements and Tags

- Heading Tags (h1-h6) and Hierarchy
- Paragraph and Text Formatting
- Line Breaks and Horizontal Rules
- Comments in HTML
- Block vs Inline Elements
- Text Formatting and Typography
- Bold, Italic, and Emphasis Tags
- Subscript and Superscript
- Quotations and Citations
- Code and Preformatted Text
- Abbreviations and Acronyms
- Text Direction and Language

Lists and Navigation

- Unordered Lists (ul, li)
- Ordered Lists (ol, li) with Different Types
- Definition Lists (dl, dt, dd)
- Nested Lists and Complex Structures
- Navigation Menus with Lists
- Links and Hyperlinks
- Anchor Tags and href Attributes
- Internal Page Links (#anchors)
- External Links and target Attributes
- Email Links (mailto:)
- Telephone Links (tel:)
- Download Links
- Link States and Accessibility

Images and Media

- Image Tags and Attributes (src, alt, title)
- Image Formats (JPEG, PNG, WebP, SVG)
- Responsive Images with srcset
- Figure and Figcaption Elements
- Image Maps and Clickable Areas
- Lazy Loading Images
- Tables and Data Presentation

1. HTML5 FUNDAMENTALS

- Table Structure (table, tr, td, th)
- Table Headers and Scope Attributes
- Table Caption and Summary
- Column and Row Spanning
- Table Sections (thead, tbody, tfoot)
- Styling Tables for Accessibility

Forms and User Input

- Form Element and Attributes
- Input Types (text, email, password, number, etc.)
- Textarea for Multi-line Text
- Select Dropdowns and Option Groups
- Radio Buttons and Checkboxes
- File Upload Inputs
- Form Labels and Accessibility
- Fieldsets and Legends
- Form Validation Attributes
- Button Types and Functionality

HTML5 Semantic Elements

- Header, Nav, Main, Section, Article
- Aside and Footer Elements
- Time and Date Elements
- Address and Contact Information
- Details and Summary for Collapsible Content
- Mark for Highlighted Text
- Progress and Meter Elements

HTML5 APIs and Features

- Audio and Video Elements
- Canvas for Graphics (Basic Introduction)
- Local Storage and Session Storages
- Geolocation API Basics
- Drag and Drop Functionality
- Web Workers Introduction

2. CSS3 COMPREHENSIVE STYLING

CSS3 COMPREHENSIVE STYLING CSS Fundamentals

- CSS Syntax and Rules
- Internal, External, and Inline Styles
- CSS Comments and Organization
- Cascade, Specificity, and Inheritance
- CSS Reset and Normalize
- Browser Default Styles

Selectors and Targeting

- Element, Class, and ID Selectors
- Descendant and Child Selectors
- Adjacent and General Sibling Selectors
- Attribute Selectors
- Pseudo-classes (:hover, :focus, :nth-child)
- Pseudo-elements (::before, ::after)
- Selector Combinators
- Selector Specificity Calculation

Typography and Text Styling

- Font Properties (family, size, weight, style)
- Web Fonts and @font-face
- Google Fonts Integration
- Text Properties (align, decoration, transform)
- Line Height and Letter Spacing
- Text Shadow and Effects
- Web Typography Best Practices
- Font Loading Performance

HTML5 APIs and Features

- Audio and Video Elements
- Canvas for Graphics (Basic Introduction)
- Local Storage and Session Storages
- Geolocation API Basics
- Drag and Drop Functionality
- Web Workers Introduction

Colors and Backgrounds

- Color Values (hex, rgb, rgba, hsl, hsla)
- Color Keywords and System Colors
- Background Properties (color, image, repeat)
- Background Size and Position
- Multiple Background Image

2. CSS3 COMPREHENSIVE STYLING

Box Model and Layout

- Understanding the Box Model
- Content, Padding, Border, Margin
- Box-sizing Property
- Display Property Values
- Width and Height Properties
- Min/Max Width and Height
- Overflow and Clipping
- Visibility and Opacity

Borders and Outlines

- Border Properties (width, style, color)
- Border Radius for Rounded Corners
- Border Images and Patterns
- Outline Properties
- Box Shadow Effects
- Multiple Shadows

Positioning and Layout

- Static, Relative, Absolute, Fixed Positioning
- Z-index and Stacking Context
- Top, Right, Bottom, Left Properties
- Sticky Positioning
- Float and Clear Properties
- Layout Methods Comparison

Flexbox Layout System

- Flex Container Properties
- Flex Direction and Wrap
- Justify-content and Align-items
- Align-self and Align-content
- Flex Item Properties (flex-grow, flex-shrink, flex-basis)
- Flex Shorthand Property
- Common Flexbox Patterns
- Flexbox vs Other Layout Methods

CSS Grid Layout

- Grid Container and Grid Items
- Grid Template Columns and Rows
- Grid Areas and Named Lines
- Grid Gap and Spacing
- Grid Auto Properties
- Grid Alignment Properties

2. CSS3 COMPREHENSIVE STYLING

Responsive Web Design

- Mobile-First Design Approach
- Media Queries and Breakpoints
- Viewport Meta Tag
- Responsive Units (em, rem, vh, vw, %)
- Flexible Images and Media
- Responsive Typography
- Container Queries (Modern CSS)
- Testing Responsive Designs

CSS Transforms and Animations

- 2D Transforms (translate, rotate, scale, skew)
- 3D Transforms and Perspective
- Transform Origin Property
- CSS Transitions
- Transition Properties and Timing
- CSS Animations and Keyframes
- Animation Properties and Control
- Performance Considerations
- Hardware Acceleration

Advanced CSS Features

- CSS Variables (Custom Properties)
- CSS Functions (calc, min, max, clamp)
- CSS Filters and Effects
- Blend Modes and Compositing
- CSS Masks and Clipping
- Counter Properties
- CSS Feature Queries (@supports)
- Print Stylesheets

CSS Preprocessor

- Introduction to Sass/SCSS
- Variables and Nesting
- Mixins and Functions
- Partials and Imports
- CSS Compilation Process
- Sass vs SCSS Syntax

CSS Frameworks and Libraries

- Bootstrap Framework
- CSS Grid Systems
- Utility-First CSS (Tailwind CSS)
- CSS Component Libraries
- When to Use Frameworks vs Custom CSS
- CSS Architecture and Organization
- CSS Methodologies (BEM, OOCSS, SMACSS)
- File Organization Strategies
- CSS Naming Conventions
- Component-Based CSS
- CSS Documentation
- Maintainable CSS Practices

3. BASIC JAVASCRIPT FOR DOM MANIPULATION

JavaScript Fundamentals for Web

- Including JavaScript in HTML
- Script Tags and External Files
- Console and Debugging Basics
- Variables and Basic Data Types
- Basic Operators and Expressions

Document Object Model (DOM)

- Understanding the DOM Tree
- Document Object and Window Object
- Selecting Elements by ID, Class, Tag
- querySelector and querySelectorAll
- Element Properties and Methods
- Node Relationships (parent, child, sibling)

DOM Manipulation Technique

- Changing Element Content (innerHTML, textContent)
- Modifying Element Attributes
- Adding and Removing CSS Classes
- Changing Inline Styles
- Creating New Elements
- Inserting and Removing Elements
- Cloning Elements

Event Handling

- Introduction to Events
- addEventListener Method
- Common Events (click, mouseover, keypress)
- Event Object and Properties
- Preventing Default Behavior
- Event Bubbling and Capturing
- Removing Event Listeners

Form Manipulation

- Accessing Form Elements
- Getting and Setting Form Values
- Form Validation with JavaScript
- Handling Form Submission
- Dynamic Form Controls

Basic Animations and Effects

- Show/Hide Elements
- Fade In/Out Effects
- Slide Effects
- Simple Image Sliders
- Interactive Buttons and Menus
- Dynamic Content Loadings

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

Introduction to ReactJS

1. ReactJS: Definition and Benefits
2. React vs. Traditional JavaScript
3. Single Page Applications (SPAs)
4. Setting up Environment (Node.js, npm, Create React App)

Core Concepts of React

1. Components: Functional vs. Class
2. JSX Syntax
3. Props & State
4. Component Lifecycle
5. Event Handling

Advanced React Features

1. React Hooks (useState, useEffect, useContext)
2. Context API
3. React Router
4. Form Handling
5. Error Boundaries

Styling in React

1. CSS Modules
2. Styled Components
3. Responsive Design (Bootstrap, Tailwind CSS)

Working with APIs

1. RESTful APIs
2. Fetch API & Axios
3. Handling Responses & Errors
4. Python Integration (Flask/ Django)

State Management

1. Local vs. Global State
2. Redux Toolkit
3. Context API vs. Redux

Performance Optimization

1. Code Splitting
2. Memoization (React.memo, useMemo)
3. Optimization Techniques

Testing & Debugging

1. React Developer Tools
2. Unit Testing (Jest, React Testing Library)
3. Common Bug Fixes

Detailed Syllabus

Full Stack Python with Django, ReactJS, and AWS

React with Python Backend

1. Python Backend (Flask/Django)
2. Data Exchange (React & Python)
3. Full-Stack Projects

Deployment

1. Build for Production
2. Deployment (Vercel, Netlify, AWS)
3. Backend Integration

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Learning Outcomes

By the end of this course, students will be able to

1. Develop and deploy full-stack applications using Python, ReactJS, and Django.
2. Design and manage relational and NoSQL databases (MySQL, MongoDB).
3. Build and integrate RESTful APIs for seamless frontend-backend communication.
4. Deploy and manage applications in cloud environments (AWS, Azure, Google Cloud).
5. Implement authentication, authorization, and security best practices.
6. Work with version control systems (Git, GitHub) for collaborative development.
7. Optimize applications for scalability, performance, and cost efficiency.
8. Gain hands-on experience with real-world projects and industry use cases.

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Certifications

NPTEL Python Certification
by IITs

NIELIT Python
Certification - '0' Level

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Job Placement Journey

- Pre-boot sessions
- Live Classes with Mentorship
- Work on Multiple Projects
- Regular Assignments
- 1:1 Mentoring
- Soft Skills, Reasoning, Aptitude & Personality Development
- Mock Interview
- Resume Building
- Portfolio Building
- Network Opportunities
- Internships
- Skill Based Interview Schedules
- Post Placement Support

+91 8341570179 www.medhaedutech.com info@medhaedutech.com

Enrollment Process

Enquiry

Contact us via our website or call us for details about the program.

Counseling

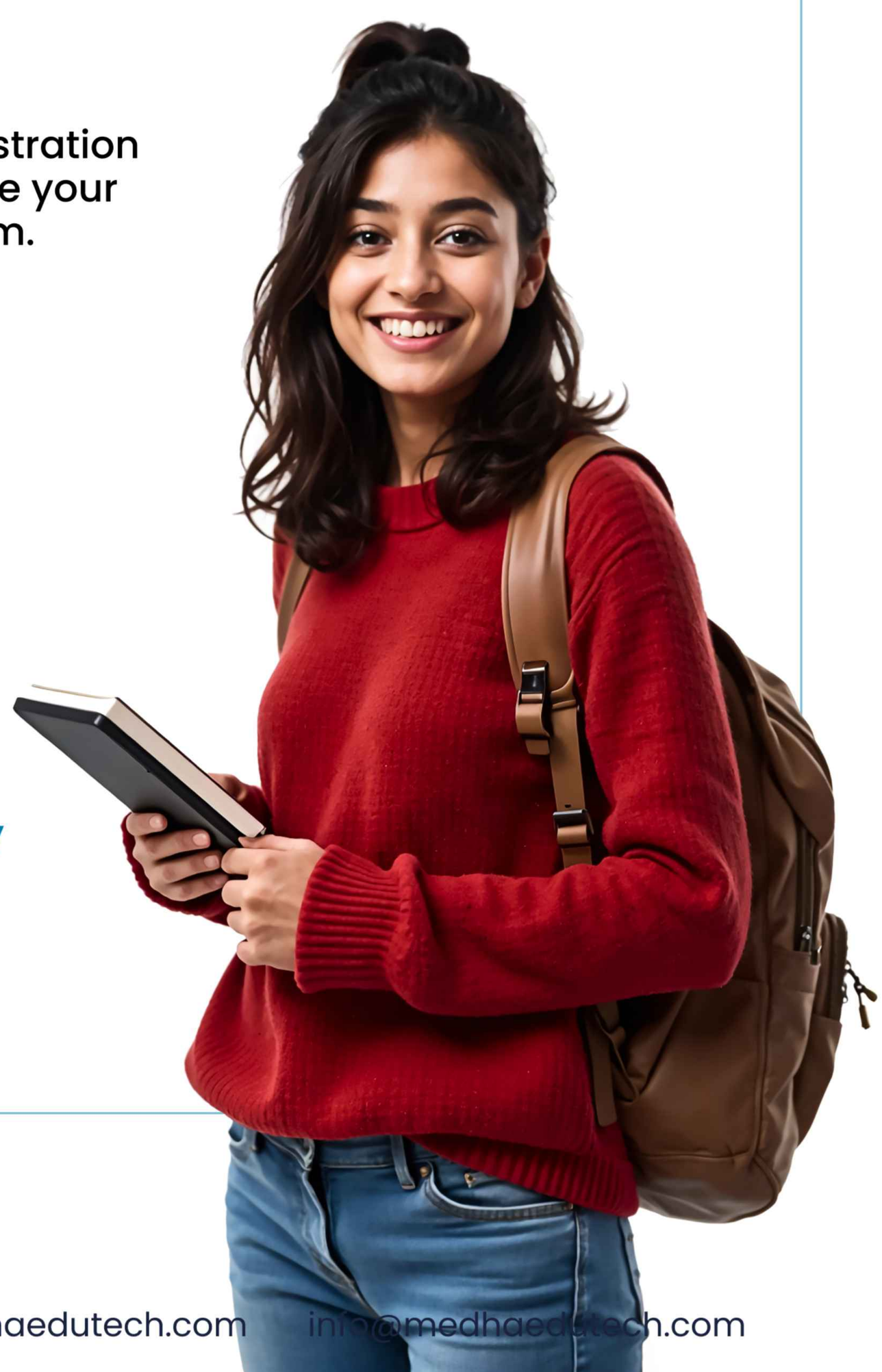
Schedule a one-on-one session with our career counselors to understand the best path for you.

Registration

Complete the registration process and secure your spot in the program.

Become Future-Ready

Enroll today and start your journey towards becoming a expert with python



MEDHA EDU TECH

educate elevate empower

Your Path to

Full Stack Python Excellence

Build complete web apps with Python, FastAPI, frontend frameworks, and live projects.

Whether you're a student, working professional, entrepreneur, or AI enthusiast, our industry-focused programs are designed to make you future-ready.

Choose from a wide range of courses

Generative AI and ML| Full Stack Java | Data Science | Multi Cloud DevOps | Cybersecurity | Digital Marketing with AI | Graphic Designing & Video Editing | Spoken English & Personality Development.

Get hands-on experience, expert mentorship, and placement assistance to kickstart your journey in tech and business. Ready to upskill? Join us today.

8341570179

info@medhaedutech.com

www.medhaedutech.com