



AiNS Computer Institute

# FULL STACK PYTHON

JOIN  
AiNS

"Empowering Innovation: Unleash the  
Full Potential of Python Across the  
Stack!"



**+91 8805845720**

[ainscomputerinstitute.com](http://ainscomputerinstitute.com)



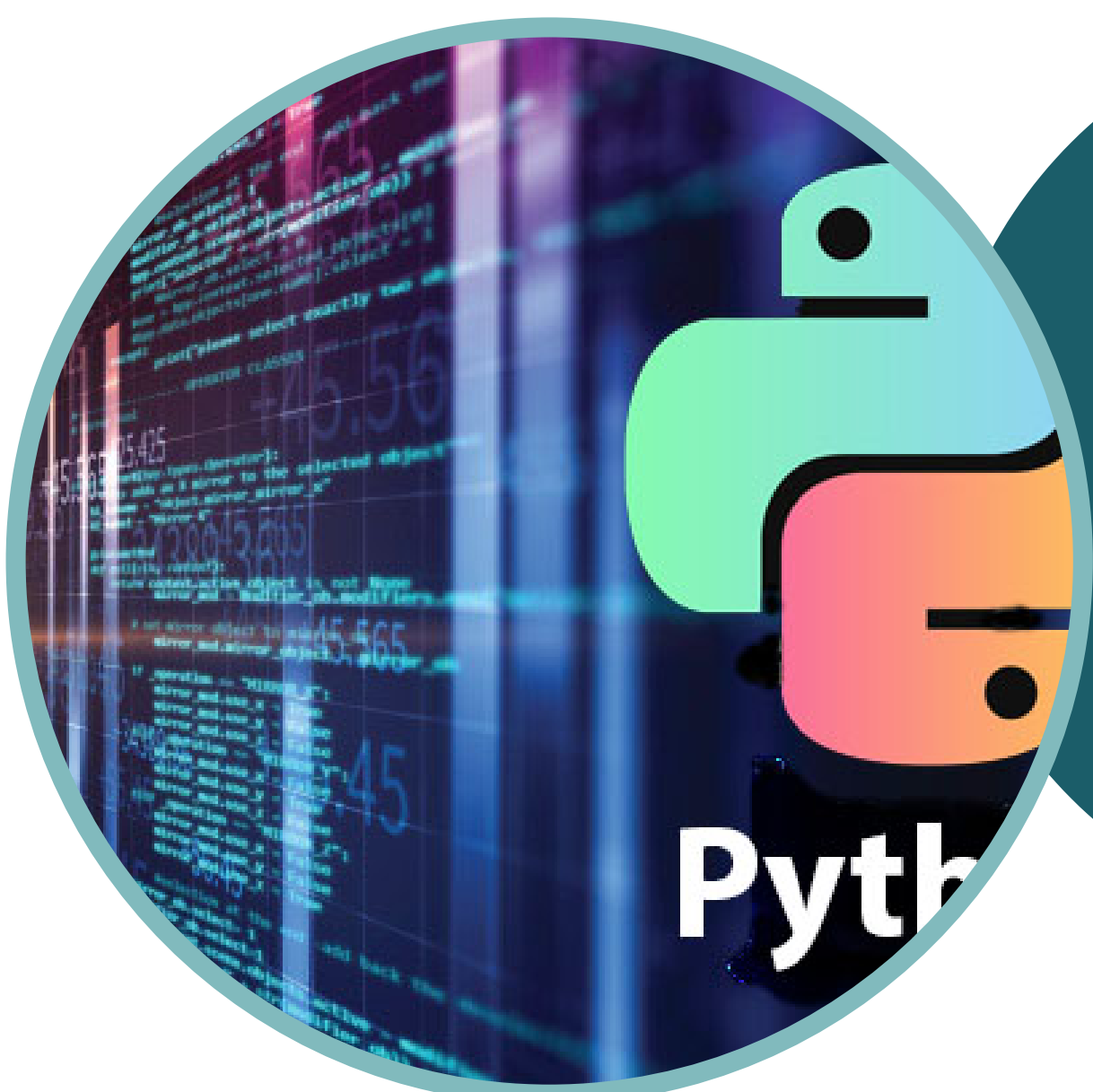
# FULL STACK PYTHON?

Full Stack Python refers to the use of the Python programming language across all layers of a web application, including both the front-end (client-side) and back-end (server-side) development.

On the front-end, Python can be used with frameworks like Django or Flask to generate dynamic content, handle user input, and interact with databases. These frameworks provide tools for URL routing, template rendering, and form handling, among other features.

For the back-end, Python can be used to create server-side logic, handle requests from clients, interact with databases, and perform various tasks such as authentication, authorization, and data processing. Frameworks like Django, Flask, and FastAPI are commonly used for back-end development in Python.

Full Stack Python developers are proficient in both front-end and back-end development, allowing them to build entire web applications using Python across the entire stack. This includes designing the user interface, implementing business logic, managing data storage, and ensuring the security and scalability of the application.

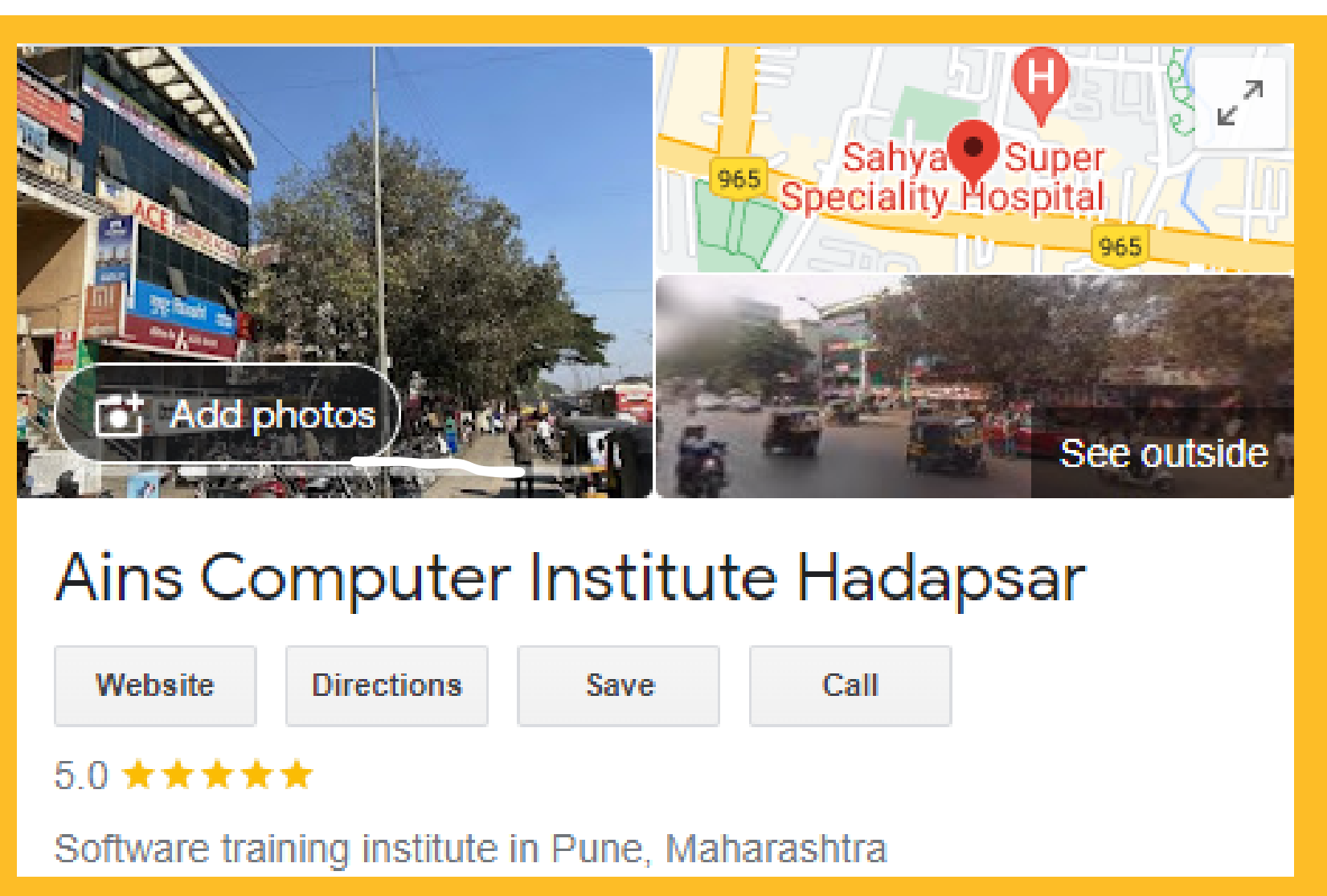




# WHY AiNS?



"AINS Computer Institute stands as a beacon of excellence in the realm of technology education, particularly in its comprehensive Full Stack Python course. Through their dedication and expertise, they have successfully cultivated a learning environment that not only imparts technical knowledge but also nurtures a deep understanding of the intricacies of Full Stack development. The institute's commitment to excellence is evident in its adept faculty, who tirelessly guide and inspire students on their journey to mastery. With state-of-the-art facilities and a curriculum designed to reflect real-world industry demands, AINS Computer Institute ensures that graduates emerge not just as proficient Full Stack Python developers but as innovators poised to make meaningful contributions in the rapidly evolving tech landscape. Their holistic approach to education, coupled with a relentless pursuit of excellence, truly sets AINS Computer Institute apart as the premier destination for those aspiring to excel in Full Stack Python development."



# BENEFITS at AiNS



- 1 **Expert Faculty**
- 2 **Industry-Relevant Curriculum**
- 3 **Career Guidance & Opportunities**
- 4 **Recognized ISO Certification**
- 5 **Real-World Data Analysis Projects**
- 6 **Access to Premium Tools and Datasets**
- 7 **Get Interview Ready & Resume Building**
- 8 **100% Job Assistance**





# Certificate of Completion

This Certificate is presented to Sarah Khan for successfully completing

his/her course in Full Stack Python by securing 98%

*We wish him/her success in life.*

**(This percentage is based on student's Practical  
& Written examination)**



9001:2015

Nasir  
Hassan  
Khan

Digitally signed by  
Nasir-Hassan Khan  
Date: 2024.03.08  
16:01:47 +05'30'

AUTHORISED SIGNATORY  
DIRECTOR





# OUR CERTIFICATION:

ISO course certification offers several advantages:

1. **International Recognition:** ISO certifications are globally recognized, demonstrating compliance with international standards and best practices.
2. **Enhanced Credibility:** Certification enhances the credibility and reputation of the organization, instilling confidence in customers, stakeholders, and partners.
3. **Improved Efficiency:** ISO standards often focus on improving processes, leading to increased efficiency, reduced waste, and cost savings.
4. **Market Access:** Certification can provide easier access to new markets, as many contracts and tenders require ISO certification as a prerequisite.
5. **Risk Management:** ISO standards help identify and mitigate risks, ensuring better management of potential threats to the organization.
6. **Customer Satisfaction:** Compliance with ISO standards often leads to higher customer satisfaction due to improved quality and consistency of products or services.
7. **Competitive Advantage:** Certification can give a competitive edge by demonstrating commitment to quality, safety, and environmental responsibility.
8. **Legal Compliance:** ISO standards often align with legal requirements, ensuring compliance and reducing the risk of penalties or legal issues.
9. **Continuous Improvement:** ISO certification encourages a culture of continuous improvement, leading to ongoing refinement of processes and performance.
10. **Employee Morale:** Employees often take pride in working for a certified organization, leading to increased morale and motivation.

value

adding



# DEMAND:

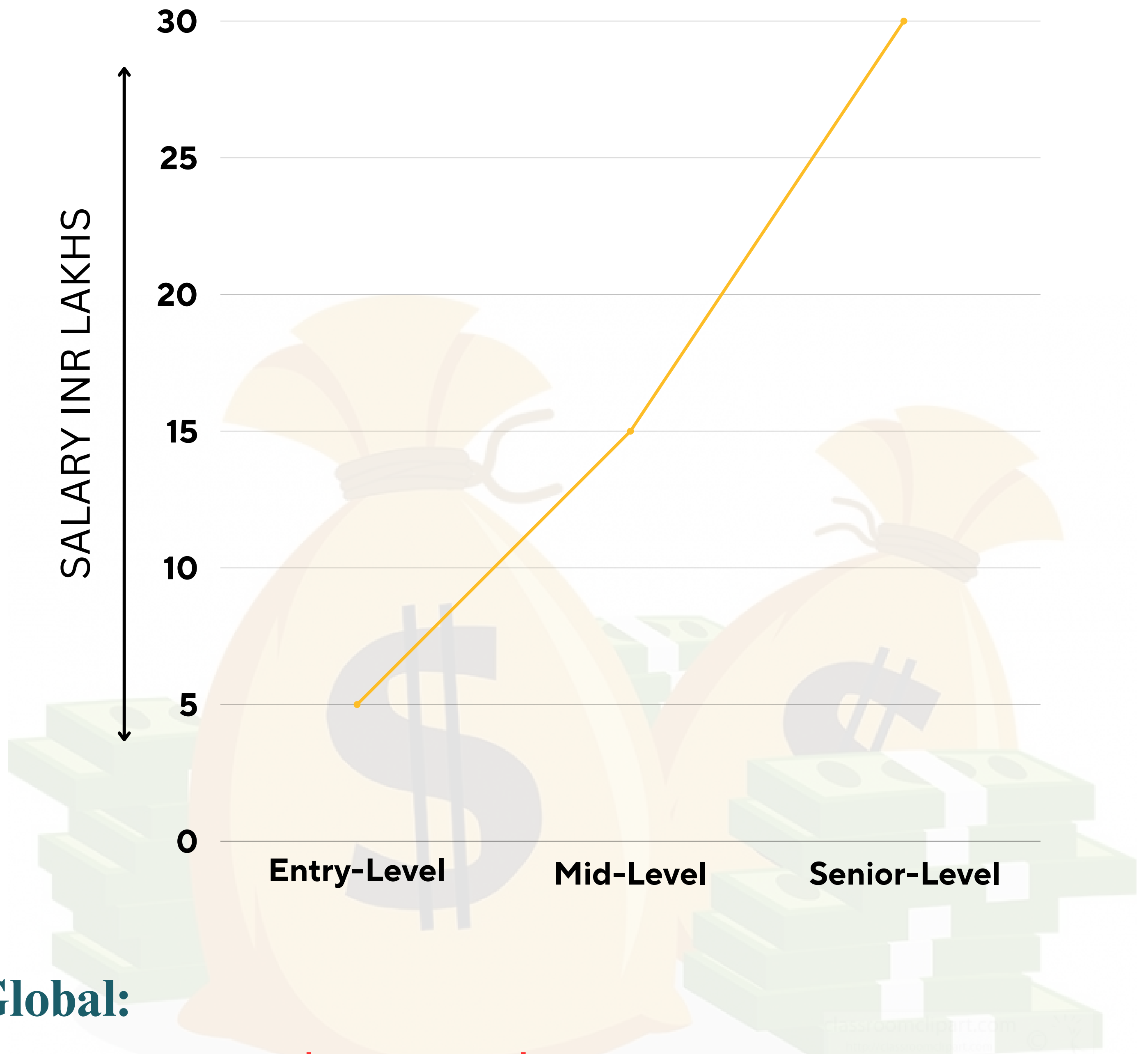
The demand for Full Stack Python developers in India and globally remains high. Python is a versatile and widely used programming language, making it popular for both front-end and back-end development in web applications. Here are some factors contributing to the demand for Full Stack Python developers:

- 1. Versatility:** Python's versatility allows developers to work on both front-end and back-end development, making Full Stack Python developers highly sought after for their ability to handle various aspects of web application development.
- 2. Popularity of Python:** Python has consistently ranked among the top programming languages in various indices, including the TIOBE index and the Stack Overflow Developer Survey. Its popularity contributes to the demand for developers proficient in Full Stack Python development.
- 3. Frameworks and Libraries:** Python has a rich ecosystem of frameworks and libraries for web development, including Django, Flask, and Pyramid for back-end development, and libraries like React, Angular, and Vue.js for front-end development. Proficiency in these tools further enhances the demand for Full Stack Python developers.
- 4. Growth of Web Development:** With the increasing digitization of businesses and the growing importance of web presence, the demand for skilled web developers, including Full Stack Python developers, continues to rise both in India and globally.
- 5. Startup Ecosystem:** Python's ease of use and rapid development capabilities make it a popular choice among startups for building Minimum Viable Products (MVPs) and scaling web applications. As the startup ecosystem continues to expand, so does the demand for Full Stack Python developers.





# SALARY IN INDIA:



## Global:

- **Entry-Level: \$60,000 - \$100,000 per year**
- **Mid-Level: \$80,000 - \$120,000 per year**
- **Senior-Level: \$100,000 - \$150,000+ per year**



# TOP COMPANIES HIRING FULL STACK PYTHON:



and many more...





# SYLLABUS CONTENT:



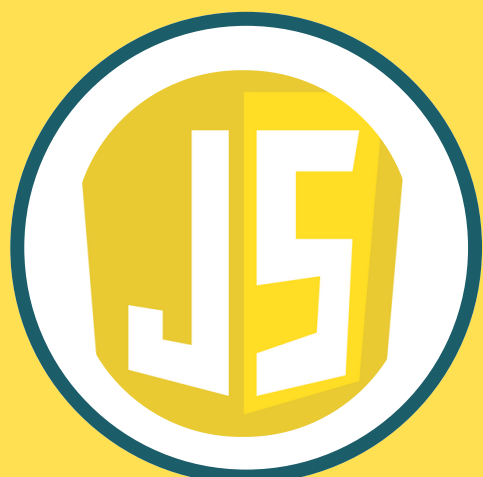
**HTML**



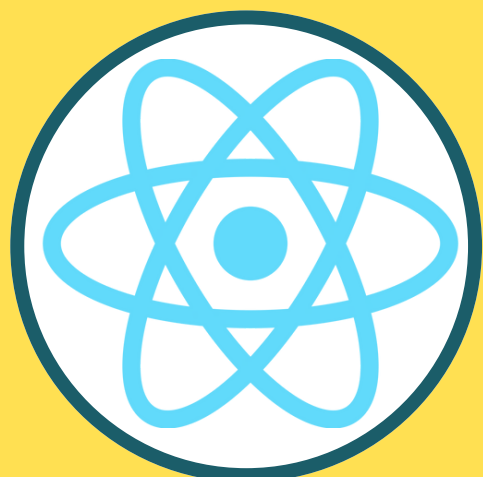
**CSS**



**BOOTSTRAP**



**JAVASCRIPT**



**REACT JS**



**PYTHON**

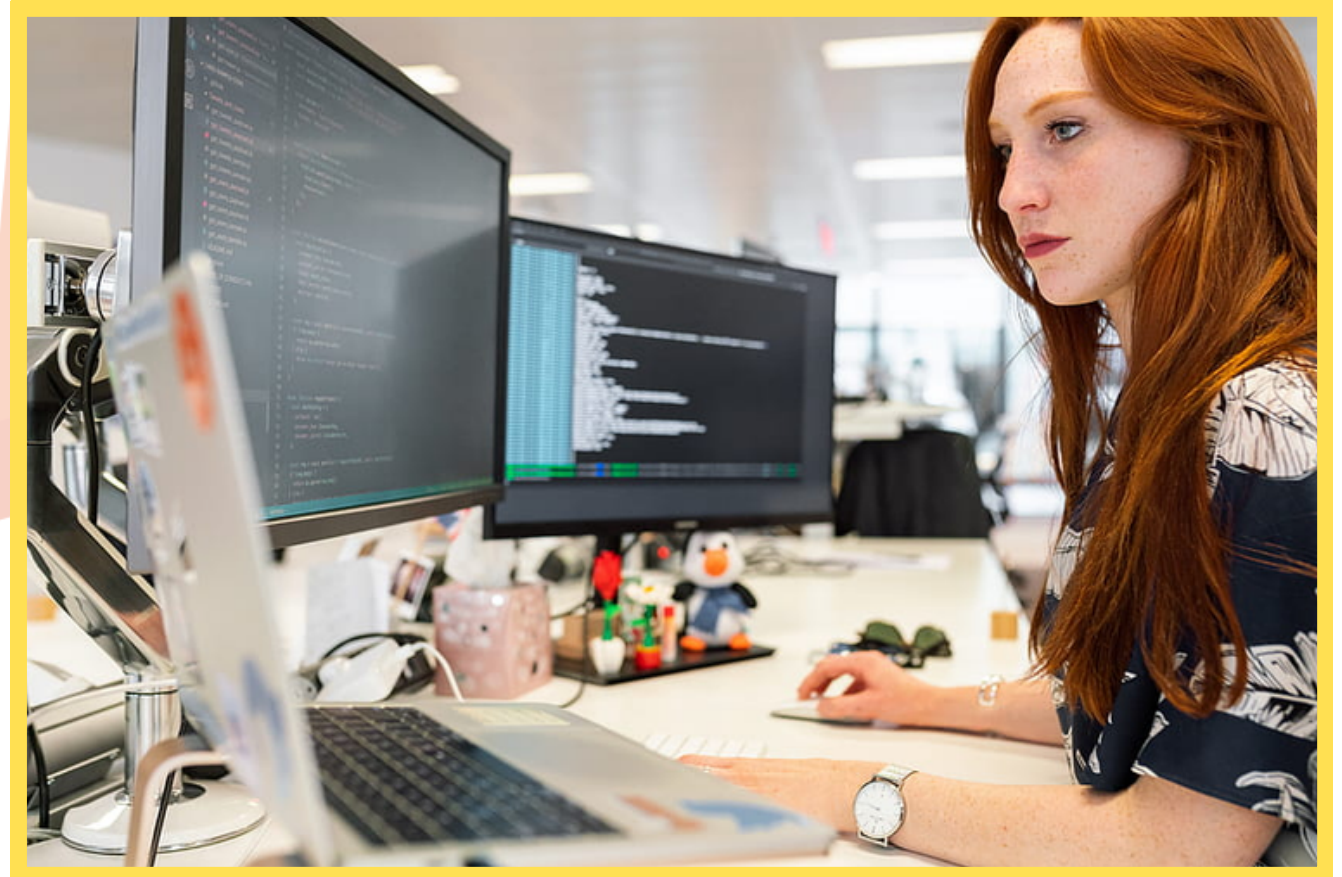


**SQL**



# HTML 4 and HTML 5

- A. Introduction of HTML
- B. Tag, Elements and Attributes
- C. Basics syntax
- D. Table
- E. List
- F. Forms
- G. Structure of HTML4 and HTML5
- H. Semantic and non-semantic tags
- I. HTML 5 Features
- J. New Input type
- K. Forms Attribute
- L. SVG
- M. Canvas
- N. Audio, Video Tag



# CSS (Cascading Style Sheet)

- A. Attributes (ID, Class, Style, Title )
- B. CSS Types (Inline, Internal, External)
- C. Box-model
- D. Display Property (Block, Inline, None)
- E. Visibility-Hidden
- F. Position Property(Static, Relative, Absolute, Fixed )
- G. Z-index Property
- H. Combinators (Descendant Selector, Child Selector, Adjacent Sibling Selector, General Sibling Selector)
- I. CSS Pseudo-classes (Link, Visited, Hover, Active)
- J. CSS Pseudo-elements (First Line, First Letter, Before, ,Selection)
- K. Static Web Page
- L. Viewport Meta tag





## **CSS 3**

- A. Background, Multiple Backgrounds**
- B. Font Related Features (online fonts)**
- C. Text-Effect and Box-Effect**
- D. Gradients-Linear and Radial**
- E. Transition**
- F. Transformation**
- G. Animation**
- H. Media Queries**

## **Bootstrap**

- A. Introduction of Bootstrap (Responsive)**
- B. Typography**
- C. Tables**
- D. Images, Buttons**
- E. Grid Structure-Type of columns**
- F. Forms**
- G. Jumbotron,**
- H. well**
- I. Panel**
- J. Navbar, Nav Tab**
- K. Carousel**
- L. Responsive Web Page**



# JavaScript

**A. Introduction of JavaScript**

**B. Use of JavaScript**

**C. Variables**

**D. Keywords**

**E. Data Type (Primitive, non-primitive)**

**F. JS Conditions (if, if-else)**

**G. Conditional operators & logical operators**

**H. Loops (for, while, do-while)**

**I. Switch Case**

**J. Functions**

**K. SetTimeout and set Interval Function**

**L. HTML DOM**

**M. Use the document object to access and manipulate HTML**

**N. Changing HTML Elements**

**O. Adding and Deleting Elements**

**P. Array**

**Q. Objects**

**R. How to access Objects (Dot Notation, bracket Notation)**

**S. Object Creation (Empty Object, Literal Way, Constructor Way)**

**T. Prototype**

**U. Validations**

**V. Events**





# REACT JS

## 1. ReactJS - Introduction

- React versions
- Features
- Benefits
- Applications



## 2. ReactJS - Installation

- Toolchain
- The serve static server
- Babel compiler
- Create React App toolchain

## 3. ReactJS - Architecture

- Workflow of a React application
- Architecture of the React Application

## 4. React – Creating a React

- Application
- Using CDN
- Using Create React App tool
- Files and folders
- Source code of the application
- Customize the code
- Run the application
- Using custom solution
- ReactJS - Using Rollup bundler
- Using Parcel bundler



## **5. React – JSX**

- **Expressions**
- **Functions**
- **Attributes**
- **Expression in attributes**

## **6. ReactJS – Component**

- **Creating a React component**
- **Creating a class component**
- **Creating a function component**

## **7. React – Styling**

- **CSS stylesheet**
- **Inline Styling**
- **CSS Modules**

## **8. React – Properties (props)**

- **Create a component using**
- **properties**
- **Nested components**
- **Use components**
- **Component collection**

## **9. React – Event management**

- **Introduce events in Expense**
- **manager app**





## **10. React – State Management**

- **What is state?**
- **State management API**
- **Stateless component**
- **Create a stateful component**
- **Introduce state in expense manager app**
- **State management using React Hooks**
- **Create a stateful component**
- **Introducing state in expense manager app**
- **Component Life cycle**
- **Working example of life cycle API**
- **Life cycle api in Expense manager app**
- **Component life cycle using React Hooks**
- **React children property aka Containment**
- **Layout in component**
- **Sharing logic in component aka Render**
- **props**
- **Pagination**
- **Material UI**

## **11. React – Http client programming**

- **Expense Rest Api Server**
- **The fetch() api**



## 12. React – Form programming

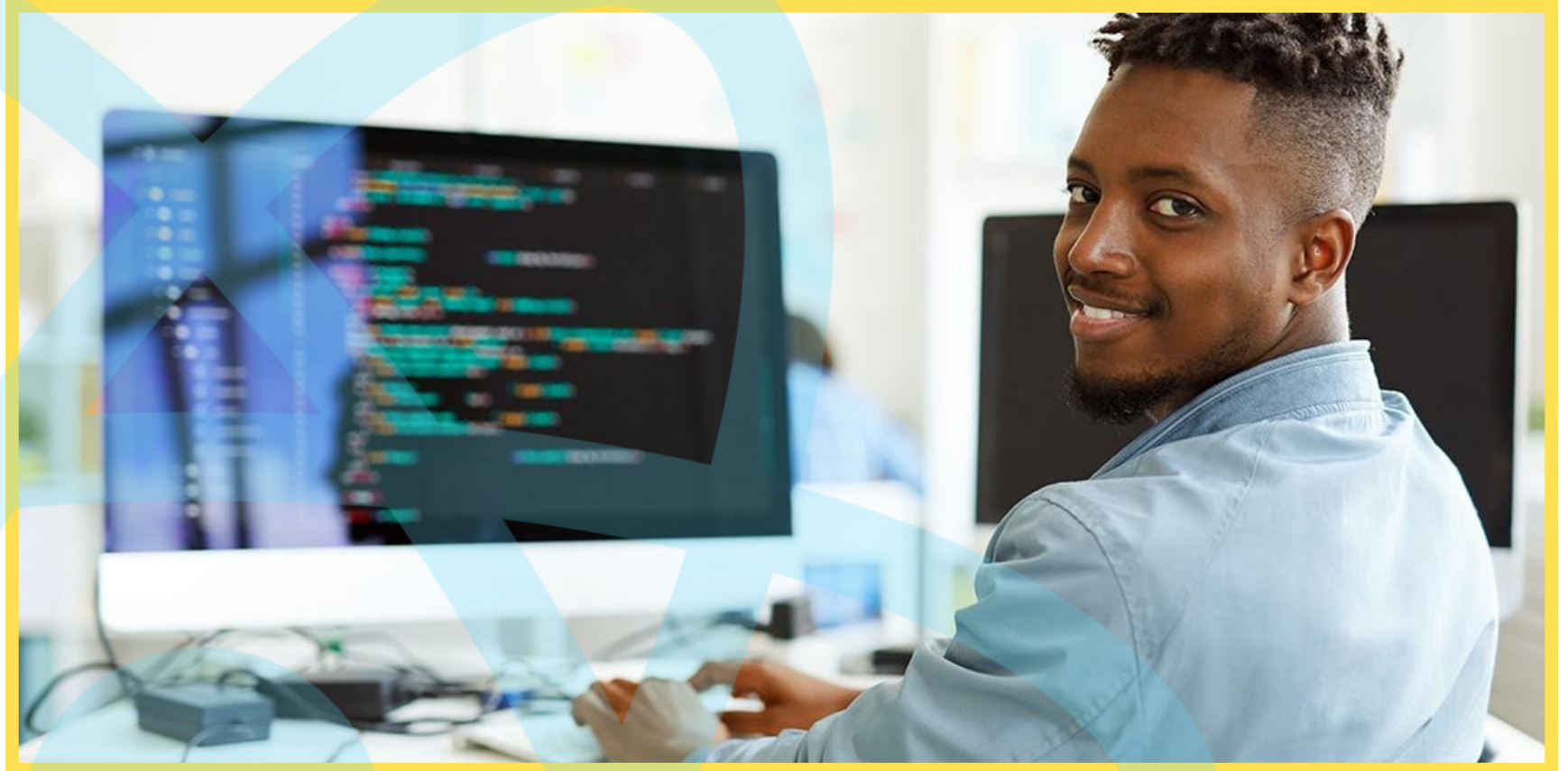
- Controlled component
- Uncontrolled Component
- Formik

## 13. React – Routing

- Install React Router
- Nested routing
- Creating navigation

## 14. React – Redux

- Concepts
- Redux API
- Provider component



## 15. React – Animation

- ReactJS
- React Transition Group
- Transition
- CSSTransition
- TransitionGroup

## 16. React – CLI Commands

- Creating a new application
- Selecting a template
- Installing a dependency
- Running the application

## 17. React – Building and Deployment

- Building
- Deployment





# PYTHON

## 1.1 Introduction to Python.

- Installation & Environment settings.
- Introduction to Shell.

## 1.2 Variables, Keywords, Data types and Identifiers.

- Variables
- Keywords
- Data types
- Identifiers

## 1.3 String, List, Set, Tuple and Dictionary and Slicing

- String Data types
- List Data types
- Set Data types
- Tuple Data types
- Dictionary Data types
- Slicing

## 1.4 Operators

- Arithmetic Operators
- Logical Operators
- Relational Operators
- Bitwise operators
- Assignment Operators
- Membership Operators
- Identity Operators

## 1.5 Control Statements

- Decisional Statements
- Looping Statements
- Break, Continue and Pass



## 1.6 Input and Print Statements

- Input statements
- Print statements

## 1.7 Functions or Methods

- Types of Functions
- Recursion
- Arguments
- Packing and unpacking(varargs)

## 1.8 Oops

- Class, Objects
- Inheritance
- Method Overriding
- Access Specifies

## 1.9 File Handling and Json

- Flat File Handling
- Json
- Pickle

## 1.10 Exception Handling

- Try
- Except and final
- Custom Exceptions
- Raising Exceptions
- Assertions

## 1.11 Comprehension

- List Comprehension

## 1.12 Decorators

- Method
- Class Level

## 1.13 Map, Filter and Lambda Expressions

## 1.14 Iterators and Generators





# Django & Flask Framework

1. Frameworks
2. Introduction to frameworks
3. Introduction to design patterns or Architectures
4. Installations and Creating Application
5. Configurations
6. HTML Responses, url mapping
7. MVC and MVT architecture
8. Connecting templates and HTML files
9. Rendering the context or data from backend
10. Static files [static images and css linking]
11. Bootstrap
12. Template tags and template inheritance
13. Models
14. Receiving data from front-end through forms
15. GET and POST methods
16. Django built-in forms module
17. Custom forms
18. Validations on built-in forms
19. Sending the data to DB
20. Model forms
21. Admin console
22. User Registration module
23. Login creation
24. CRUD operations
25. Integration of Mail API
26. Git
27. Hosting an application
28. Sessions and cookies
29. Flask Framework introduction



# MySQL

- MySQL Core and SQL Basics
- MySQL Server Basics
- Database models
- ER Model Overview
- Data types
- Understanding Test Database
- Basics Queries
- Removing Duplicates
- Data Filters Using Operators
- Data Sorting
- Query Design & Functions
- Grouping
- Joins
- Arithmetic and String functions
- Advanced Functions
- SET Operators
- Creating Complex Queries
- DML operations - Insert, Update & Delete
- TSQL





# Database Objects -

- Create, Alter and Drop
- Tables
- Views
- Complex Views
- Indexes
- Advanced Index
- Concepts
- Security - User
- Management Basics
- Routines
- Programming Basics
- Programming
- Concepts in PLSQL
- Cursors
- Nested Cursors
- Functions
- Procedures
- Interoperability
- between Functions &
- Procedures
- Query Optimization
- Techniques
- Project / Case Study

