



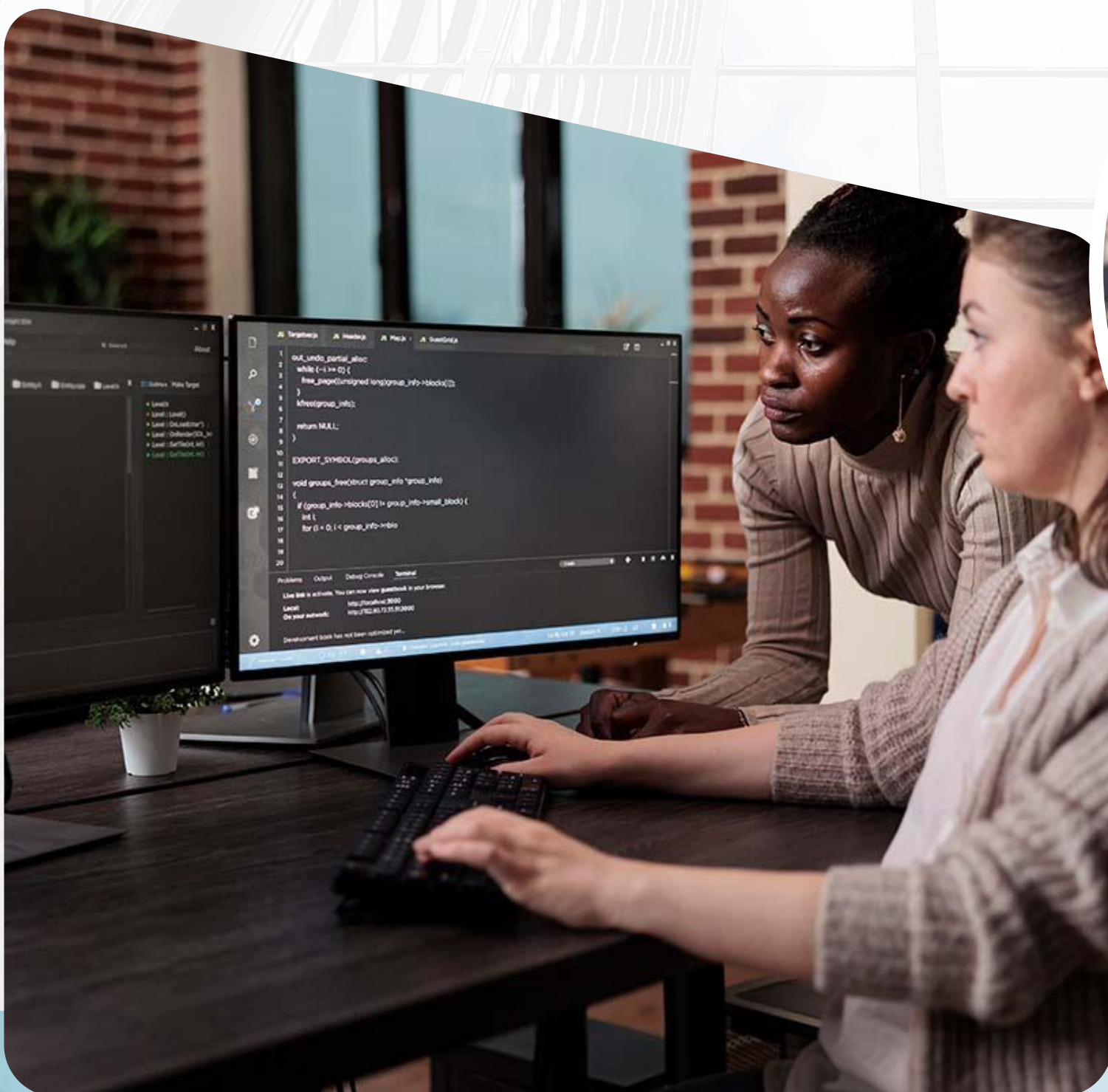
AiNS Computer Institute

FULL STACK JAVA



"Empowering End-to-End Solutions:
Full Stack Java for Seamless
Development"

JOIN
AiNS



+91 8805845720

ainscomputerinstitute.com

FULL STACK JAVA?

Full Stack Java development refers to the practice of using Java technologies to build both the front-end and back-end components of a web application. A Full Stack Java developer is proficient in both front-end and back-end technologies, enabling them to work on all aspects of web application development.

In Full Stack Java development, the front-end typically involves technologies like HTML, CSS, and JavaScript, along with frameworks/libraries like Angular, React, or Vue.js. These technologies are used to create the user interface and handle client-side interactions.

On the back-end, Java is commonly used along with frameworks like Spring Boot, Jakarta EE (formerly Java EE), or Play Framework. These frameworks provide tools and libraries for building robust and scalable server-side applications, handling tasks such as routing, data persistence, security, and integration with databases and other systems.

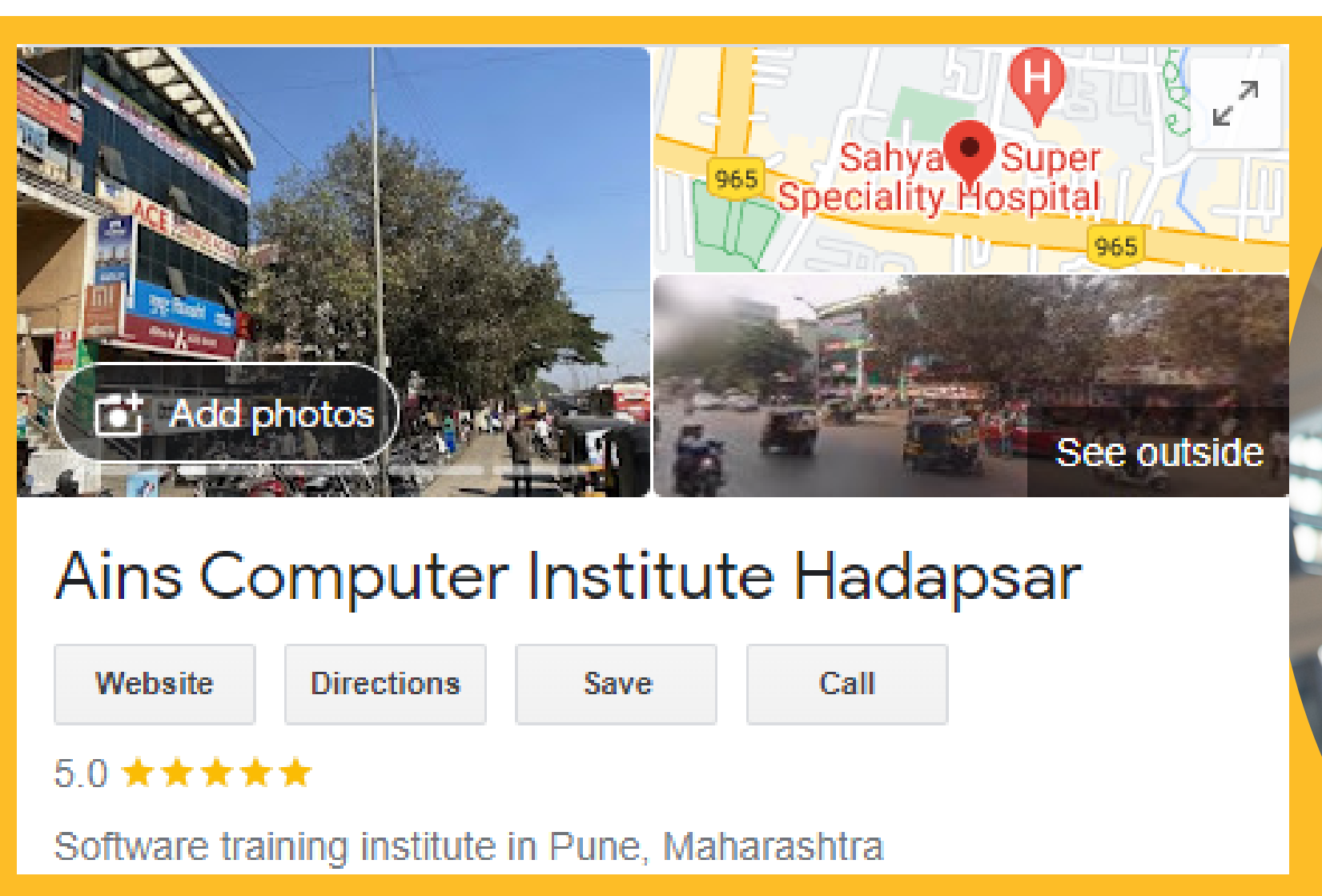
Full Stack Java developers are responsible for designing, implementing, and maintaining both the front-end and back-end components of web applications. They need to have a solid understanding of software development principles, database management, and system architecture to build efficient and effective applications.

Overall, Full Stack Java development allows developers to leverage the power of Java across the entire stack of a web application, from the user interface to the server-side logic, providing a comprehensive and cohesive approach to web development.



WHY AiNS?

AiNS Computer Institute stands out as the ideal destination for mastering Full Stack Java development. With a comprehensive curriculum designed by industry experts, our Full Stack Java course covers every aspect of Java programming, equipping students with the skills needed to excel in today's competitive tech landscape. Our experienced instructors provide personalized guidance and hands-on training, ensuring that students gain practical experience in developing robust web applications using Java technologies. Furthermore, our state-of-the-art facilities and supportive learning environment foster a dynamic atmosphere where students can thrive and unleash their full potential. By choosing AiNS Computer Institute for the Full Stack Java course, students embark on a transformative learning journey that prepares them for rewarding careers in software 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 Java by securing 98%

We wish him/her success in life.

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

Nasir
Hassan
Khan
Digitally signed
by Nasir Hassan
Khan
Date: 2024.03.07
19:41:07 +05'30'

AUTHORISED SIGNATORY
DIRECTOR



9001:2015



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

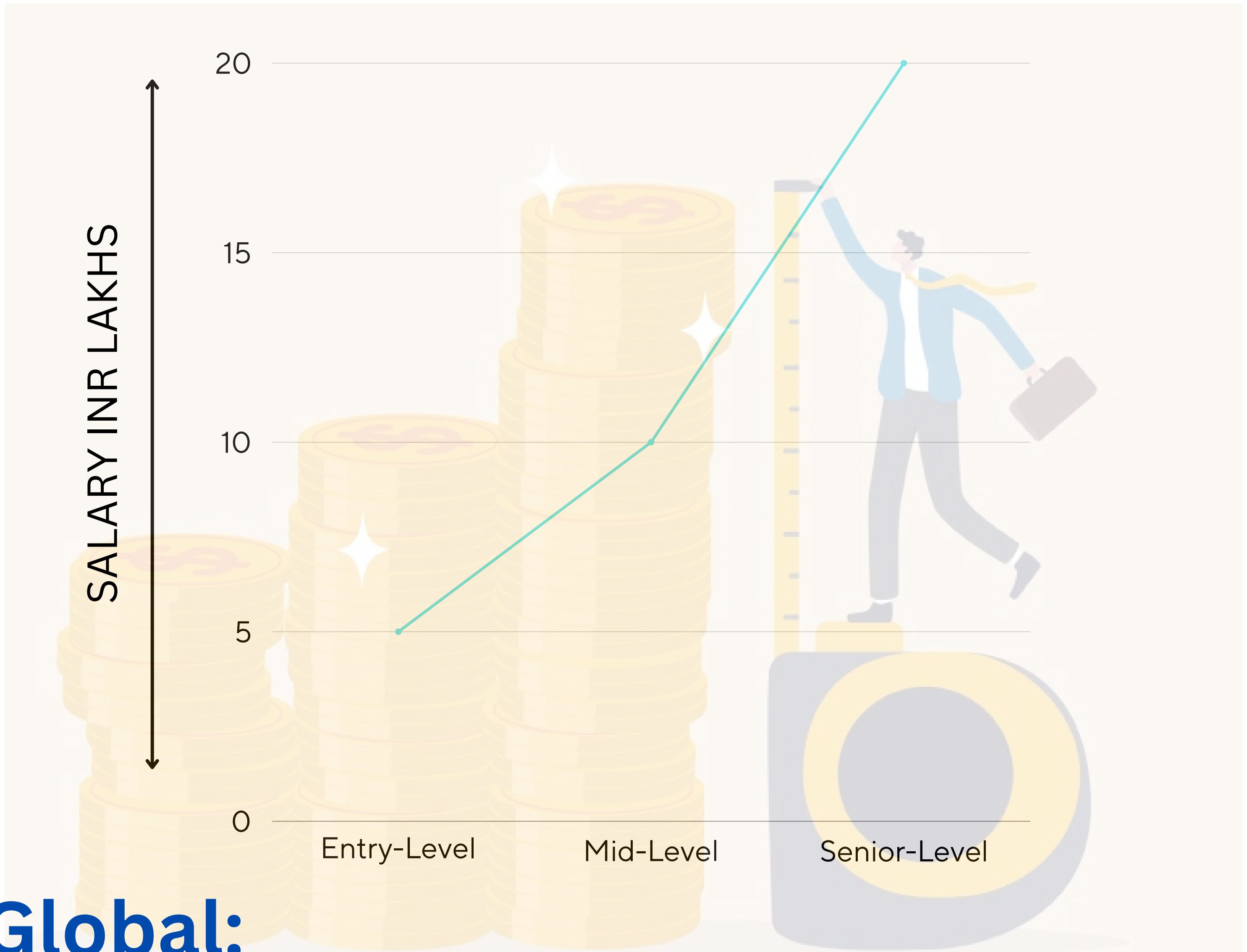
adding

DEMAND:

- **Versatility:** Full Stack Java developers possess skills in both front-end and back-end development, making them versatile and capable of handling a wide range of tasks in web application development.
- **Java's Popularity:** Java continues to be one of the most widely used programming languages in the world, especially in enterprise-level applications. Many organizations have existing Java codebases and require developers who can work with Java for both front-end and back-end development.
- **Full Stack Development Trend:** The trend towards full stack development has been growing, with many companies preferring developers who can work on all aspects of web application development rather than specializing in just front-end or back-end.
- **Startup Culture:** In India, the startup ecosystem is booming, and startups often look for Full Stack Java developers who can contribute across the entire technology stack due to limited resources and the need for versatility.
- **Remote Work Opportunities:** With the rise of remote work, Full Stack Java developers have the flexibility to work for companies across the globe, increasing their opportunities for employment.
- **Salary and Career Growth:** Due to the high demand and specialized skill set, Full Stack Java developers often command higher salaries and have opportunities for career growth and advancement.



SALARY IN INDIA:

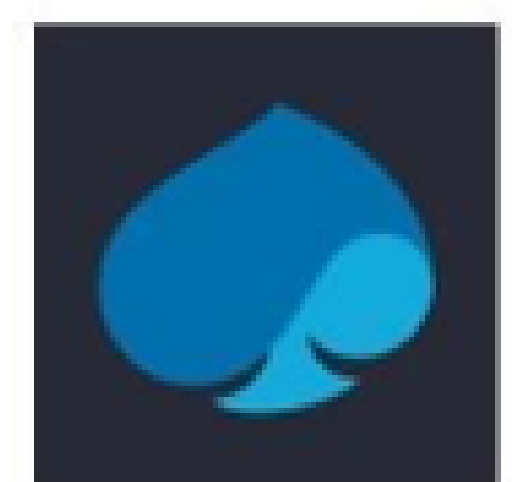


Global:

- **Entry-Level: \$50,000 - \$80,000 per year**
- **Mid-Level: \$80,000 - \$120,000 per year**
- **Senior-Level: \$120,000+ per year**



TOP COMPANIES HIRING FULL STACK JAVA:



SYLLABUS CONTENT:



CORE JAVA



**ADVANCED
JAVA**



HTML



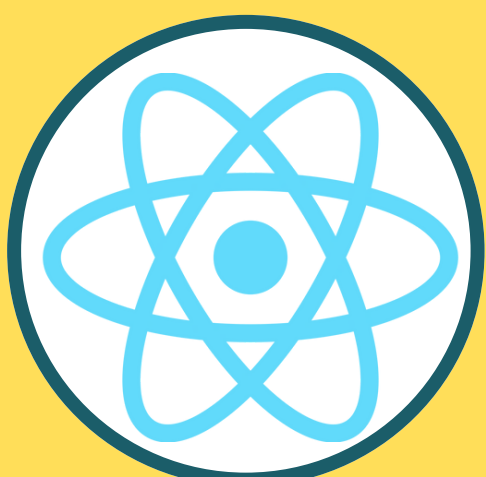
BOOTSTRAP



CSS



JAVA SCRIPT



REACT



MYSQL



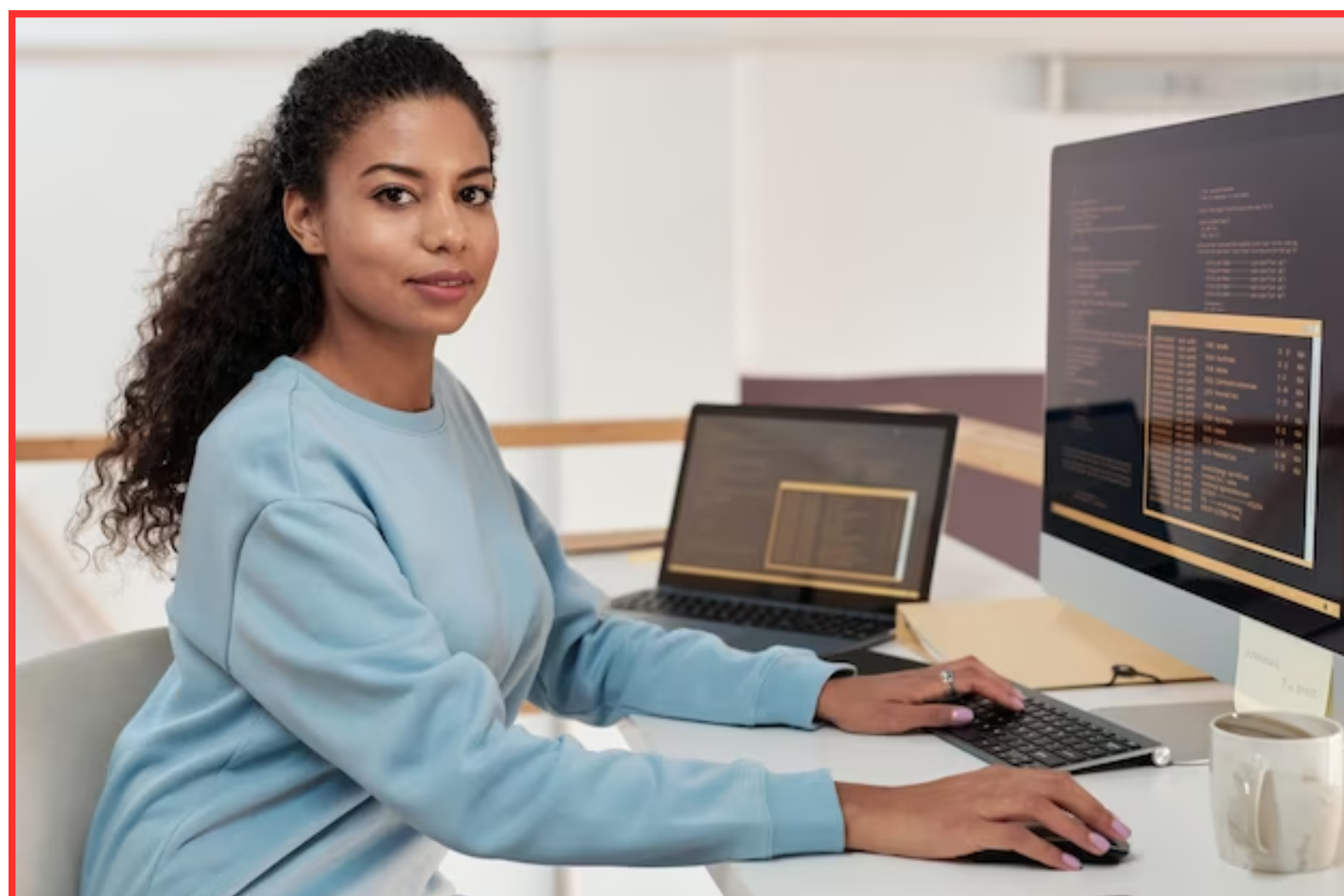
Core JAVA Syllabus

1. Introduction of Java
2. Data types in Java
3. Operators In Java
4. Control Statements in java
5. Arrays in Java
6. String Class in Java
7. StringBuffer and StringBuilder
8. Introduction to OOP
9. Classes & Objects
10. Inheritance
11. Polymorphism
12. Abstraction
13. Encapsulation
14. Packages
15. Exception Handling
16. Wrapper Classes
17. Threads
18. IO Streams & Files
19. Generics
20. Collection & MAP
21. AWT
22. Swing
23. Type Casting
24. Abstract Classes & Interfaces



Advance JAVA Syllabus

- 1. Java 8 Features**
- 2. INTRODUCTION OF advance Java**
- 3Java Database Connectivity (JDBC)**
- 4. eXtensible Markup Language (XML)**
- 5. Common Gateway Interface**
- 6. Java Servlets**
- 7. Understanding the Deployment Descriptor (DD) / web.xml**
- 8. Using HTTP Session**
- 9. Web Application Security**
- 10. Security Constraints**
- 11. Java Database Connectivity (JDBC)**
- 12. Forward Action Tag**
- 13. JavaBeans**
- 14. Java Database Connectivity (JDBC)**
- 15. eXtensible Markup Language (XML)**



36. Common Gateway Interface

37. Java Servlets

38. Forward Action Tag

39. Understanding the

Deployment Descriptor (DD) / web.xml

40. Using HTTP Session

41. Web Application Security

42. Security Constraints

43. Spring Basics

44. Spring Container

45. JavaBeans

46. Spring AOP

47. Spring Data Access

48. Spring O-R /mapping

49. Spring Web MVC Framework

50. Role of DispatcherServlet

51. Introduction Hibernate

52. Hibernate Configuration

53. Hibernate Concepts

54. Hibernate O-R Mapping

55. Introduction Hibernate

56. Manipulating and Querying

57. Hibernate Query Language

58. Criteria Queries



Web Development Syllabus

1. 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



2. 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, After ,Selection)

K. Static Web Page

L. Viewport Meta tag



3. 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**

4. 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**



5. 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

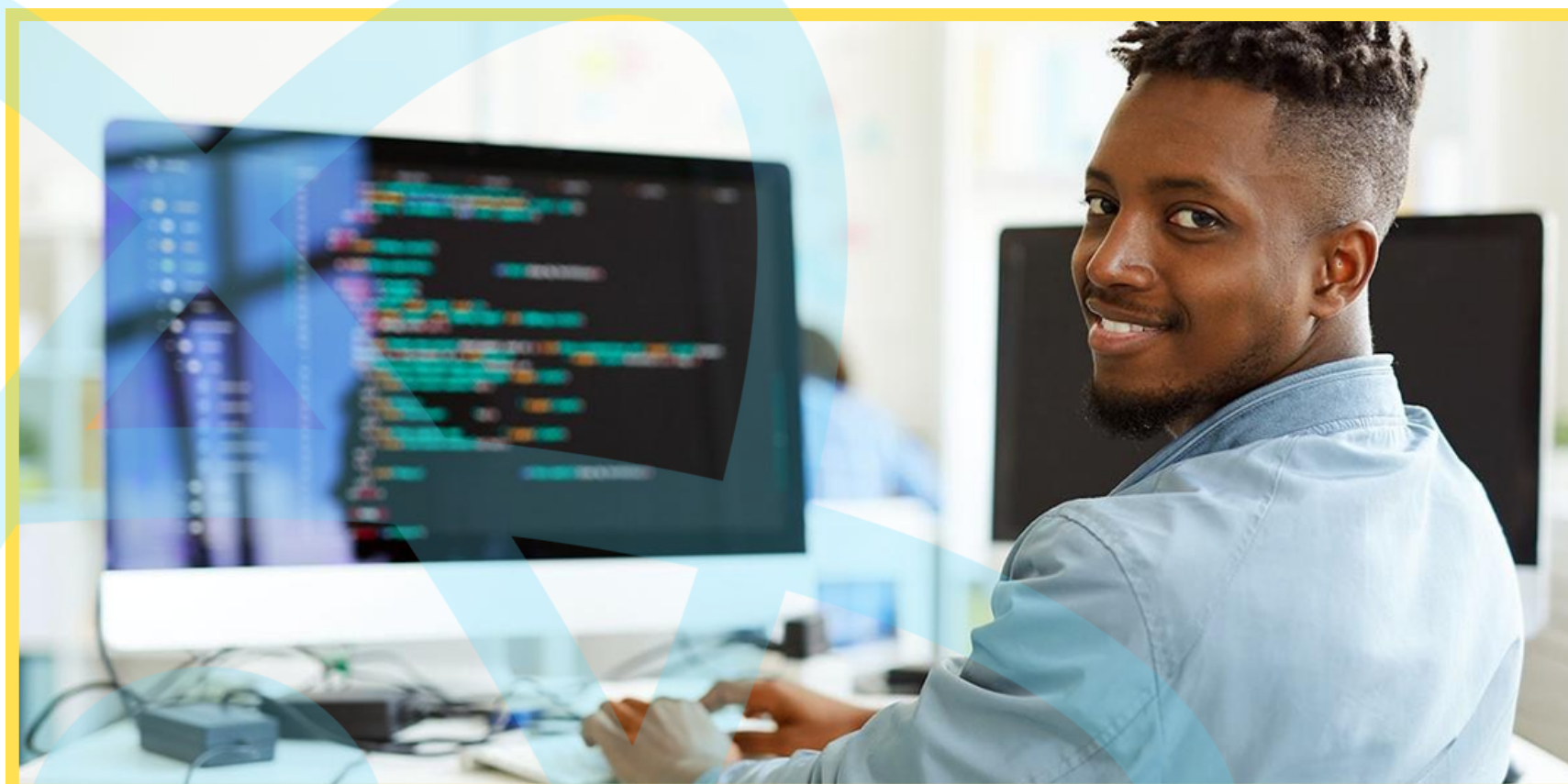
- **Controlled component**
- **Uncontrolled Component**
- **Formik**

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**



7.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 Operations**
- **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**

