LEARN 2 EARN LABS

Training Syllabus

[6 months Duration]

Java Full Stack Development

Join us for Better Career and Job Guarantee

[Live Online & Offline Classes Available]

LEARN-2-EARN LABS TRAINING INSTITUTE, AGRA

Website: www.LearntoearnLabs.com | Contact : +91-9548868337

Java Full Stack Training Program – Six Months Duration

The **Java Full Stack Training Program** is a **six-months,** intensive, hands-on course designed to equip learners with the skills necessary to build dynamic, scalable, and secure web applications using Java and modern full-stack development tools. With the increasing demand for Java-based enterprise solutions, this program ensures that participants gain practical experience in designing, developing, and deploying web applications that align with industry standards.

This program covers both front-end and back-end technologies, ensuring that learners are well-versed in every aspect of web development. The curriculum is structured to provide a step-by-step learning process, starting from Java programming fundamentals, progressing to advanced object-oriented concepts, databases, web frameworks, and full-stack development with Servlets, JSP, JDBC, MySQL, and front-end technologies like HTML, CSS, and JavaScript.

About the Program

This training program is meticulously structured to transform absolute beginners into professional Java full-stack developers who can confidently build fully functional, scalable, and high-performing web applications. The program follows a project-based learning approach, ensuring that students build real-world applications as they progress.

Participants will start with core Java programming concepts before transitioning into web development, database integration, API development, and server-side technologies. By the end of the training, they will have the expertise to develop and manage complete web applications, including front-end UI, backend API, authentication, state management, and performance optimization.

Who is a Java Full Stack Developer?

A Java Full Stack Developer is a highly skilled professional responsible for designing, developing, and maintaining end-to-end web applications using a combination of front-end, back-end, APIs, webservices, microservices, database operations, etc.. They are proficient in a range of technologies, including Spring, Spring Boot, Microservices, Hibernate, ORM frameworks, REST APIs, JSP, Servlets, JDBC, and various front-end technologies & tools.

Their role includes:

- **Back-End Development:** Writing efficient, secure, and scalable server-side logic using Spring Framework, Spring Boot, and Microservices architecture.
- **Front-End Development:** Creating responsive and interactive user interfaces using HTML, CSS, JavaScript, and Java-based front-end frameworks.
- **Database Management:** Managing relational databases like MySQL, PostgreSQL, or Oracle with JPA, Hibernate, and JDBC for seamless data storage and retrieval.
- **Microservices Architecture:** Designing and implementing scalable and loosely coupled services for enterprise applications.
- **RESTful API Development:** Building secure and efficient REST APIs for communication between front-end and back-end systems.
- **Authentication & Security:** Implementing authentication mechanisms like **JWT, OAuth2, and Spring Security** to protect applications from vulnerabilities.
- **Deployment & DevOps:** Managing CI/CD pipelines, Docker, Kubernetes, and cloud platforms for deploying and maintaining applications efficiently.
- **Performance Optimization:** Enhancing application speed, database queries, and server response times for seamless user experiences.
- **Testing & Debugging:** Writing unit tests with JUnit, Mockito, and integration tests to ensure application reliability.

A Java Full Stack Developer is proficient in handling both client-side and server-side development, making them highly valuable in modern software development. With expertise in cutting-edge Java frameworks and technologies, they play a crucial role in building high-performance web applications for various industries. With expertise in Java full-stack development, developers can create **robust, secure, and scalable applications**, making them an essential asset to companies looking to build enterprise solutions.

What Makes This Program Unique?

This training program stands out due to its comprehensive curriculum, practical learning approach, and industry relevance. Below are some key highlights that make this program unique:

- **Comprehensive Curriculum:** The course covers Java programming, object-oriented concepts, Servlets, JSP, JDBC, MySQL, RESTful APIs, front-end development, and performance optimization.
- **Hands-On Learning:** Unlike theoretical courses, this program follows a project-based approach, where students build real-world Java applications.

- **Industry-Relevant Skills:** Learn cutting-edge technologies, including JSP, JDBC, Servlets, RESTful APIs, and advanced Java concepts.
- **Expert Mentorship:** Students receive personalized mentorship and guidance from industry professionals.
- **Performance Optimization:** Learn how to debug, optimize, and improve Java web application performance.
- **Real-World Application Development:** Work on end-to-end projects that mirror real-world business needs.
- **Backend Development Expertise:** Build robust backends using JDBC, MySQL, and Servlets to create a full-stack web application experience.

Career Options After Completion

By completing this program, learners will **acquire the necessary skills** to enter the software development industry confidently. Possible career opportunities include:

- **Java Full Stack Developer** – Develop and maintain Java-based web applications.
- **Back-End Developer** – Specialize in building and managing server-side applications.
- **Front-End Developer** – Design and implement user interfaces for Java applications.
- **Software Engineer** – Work in enterprise-level application development teams.
- **Web Application Developer** – Build and deploy scalable web applications.
- **Freelance Java Developer** – Work on independent Java-based projects.
- **Java Application Support Engineer** – Provide support, troubleshooting, and maintenance for Java applications.
- **Enterprise Java Developer** – Develop large-scale enterprise applications using Java technologies.

Who Can Join This Program?

This program is suitable for:

- **Beginners** who want to start their career in full-stack development.
- **Computer Science students or graduates** looking for hands-on Java development experience.
- **Web developers** who want to transition into Java full-stack development.
- **Professionals** who want to upgrade their skills in modern Java frameworks.
- **Entrepreneurs** looking to build scalable web applications.

Advantages of the Program

The Java Full Stack Training Program provides learners with a **structured learning experience** that ensures they can confidently develop and manage web applications. Key advantages include:

- **Mastering Front-End & Back-End Technologies:** Develop expertise in Java, JSP, Servlets, MySQL, REST APIs, and front-end frameworks.
- **Hands-on Projects:** Work on live projects such as e-commerce platforms, enterprise web applications, and real-time web applications.
- **Learn Performance Optimization:** Discover techniques for debugging, improving code efficiency, and reducing response times.
- **Practical Knowledge of API Development:** Work with RESTful APIs, authentication mechanisms, and secure transactions.
- **Portfolio Development:** Build a portfolio of real-world projects to showcase to potential employers or clients.

How Much Dedication is Required?

This is an intensive program that requires a **dedicated learning commitment**. Students are expected to:

- **Dedicate 12-15 hours per week** for learning, assignments, and projects.
- **Complete real-world projects** to enhance their practical experience.
- **Actively participate in coding challenges and discussions** to reinforce learning.
- **Develop a disciplined learning routine** to keep up with assignments and coding exercises.
- **Engage in self-study and research** to stay updated with the latest Java frameworks and best practices.

Future Scope

The demand for Java full-stack developers is constantly growing as businesses continue to build enterprise applications using Java-based technologies. Java professionals are highly sought after, and this program prepares learners to be competitive in the job market.

With expertise in Java, learners can work in companies using Servlets, JSP, JDBC, and database-driven applications. Java is used by major corporations like **Google, Amazon, Microsoft, Netflix, IBM, etc.**, making it a valuable skill in the industry.

Practical Experience to Boost Your Career

During this program, students will work on multiple hands-on projects, including:

- **Enterprise Web Application** – Build and manage web applications with user authentication and database interactions.
- **E-Commerce Platform** – Implement shopping carts, payment gateways, and product listings.
- **Real-Time Data Application** – Create applications with live data updates and user interactions.

These real-world projects will help build an impressive portfolio, making learners more employable.

Transform Your Career Today

The Java Full Stack Training Program provides a structured and immersive learning experience that enables learners to **become professional full-stack developers**. Whether you aim to work in a top tech company or develop your own applications, this program will equip you with the necessary skills and knowledge.

By mastering Java full-stack development, you gain the ability to develop scalable, secure, and high-performing applications. This program not only prepares you for job-ready roles but also helps you build confidence in solving real-world challenges.

Your Journey to Success Starts Here!

Take your first step towards a rewarding career in full-stack development with the Java Full Stack Training Program.

Join today and become a highly skilled, job-ready developer in just six months!

Training Content (Syllabus)

Module 1 - Introduction to Java : Introduction to Java and its features, JDK, JRE, and JVM - Differences & Setup, Java Virtual Machine (JVM) Architecture, Java Compilation and Execution Process, Writing and Running a Simple Java Program, Variables, Data Types, and Type Casting, Java Operators: Arithmetic, Logical, Relational, Bitwise, Control Flow Statements: if-else, switch-case, Loops: for, while, do-while, enhanced for (forEach), Methods: Declaration, Calling, and Overloading, Method Parameters: Pass by Value vs Pass by Reference, Introduction to Arrays: Single & Multidimensional, Java String Handling and String Methods, StringBuilder and StringBuffer - Performance Comparison, Java Memory Management (Heap & Stack), Static vs Non-Static Methods and Variables, Java Naming Conventions & Coding Standards, Java Packages: Importing & Organizing Code, Understanding Classpath & Module System, Java IDEs Setup: Eclipse, IntelliJ IDEA, NetBeans, Writing a Simple Java Application with Console I/O, Scanner Class for User Input Handling, Understanding Primitive vs Wrapper Classes, Java Math Class and Random Numbers, Best Practices for Writing Efficient Java Code.

Module 2 - Object Oriented Concepts : What is OOP? Principles & Benefits, Classes & Objects: Creating and Using Objects, Constructors: Default, Parameterized, Copy Constructor, this Keyword & Method Chaining, Static vs Instance Members, Encapsulation: Getters & Setters, Inheritance: Types (Single, Multilevel, Hierarchical), Method Overriding vs Overloading, super Keyword in Inheritance, final Keyword: Variables, Methods, and Classes, Abstraction: Abstract Classes vs Interfaces, Interface Implementation & Default Methods, Multiple Inheritance in Java (Using Interfaces), Polymorphism: Compile-time vs Runtime, Nested & Inner Classes, Anonymous Inner Classes, Java Lambda Expressions & Functional Interfaces, Java Enum & Custom Enum Methods, Annotations: @Override, @Deprecated, @FunctionalInterface, Understanding Object Class & toString() Method, Cloning Objects in Java, Garbage Collection & finalize() Method, Comparing Objects using equals() and hashCode(), Best Practices for Object-Oriented Programming.

Module 3 - Exception Handling : What are Exceptions?, Difference Between Checked and Unchecked Exceptions, Java Exception Hierarchy, Handling Exceptions using try-catch, Multiple catch Blocks, finally Block: When and How to Use It, throw Keyword: Throwing Custom Exceptions, throws Keyword: Declaring Exceptions, Creating Custom Exceptions, Exception Propagation in Java, Stack Trace Analysis, Handling NullPointerException, Handling ArithmeticException (Divide by Zero), Handling ArrayIndexOutOfBoundsException, Handling ClassNotFoundException, Handling FileNotFoundException, Handling NumberFormatException, Handling IOException & SQLException, Using Assertions for Debugging, Multi-Catch Exception Handling, Try-With-Resources for Automatic Resource Management, Logging Exception Details

using Java Logger, Rethrowing Exceptions in Java, Using Custom Exception Messages, Best Practices for Exception Handling.

Module 4 - I/O & File Handling : Introduction to Java I/O Streams, Byte Streams vs Character Streams, File Class & File Operations, Creating and Deleting Files in Java, Reading and Writing Text Files, BufferedReader & BufferedWriter, FileReader & FileWriter, InputStreamReader & OutputStreamWriter, Reading Console Input Using Scanner Class, FileInputStream & FileOutputStream, Copying File Content Using Streams, Random Access File Handling, Serialization & Deserialization, ObjectInputStream & ObjectOutputStream, Using Properties File in Java, ZIP File Handling in Java, Reading and Writing JSON Files, Reading and Writing CSV Files, Handling Large Files Using Streams, Understanding Charset and Encoding, Working with NIO Package for Fast File Handling, Java 8 File I/O Enhancements (Files API), Writing a Simple File Handling Utility Class, Exception Handling in File Operations, Best Practices for Java File Handling.

Module 5 - Multithreading & Concurrency : What is Multithreading? Benefits & Use Cases, Thread Lifecycle & States, Creating Threads Using Thread Class, Creating Threads Using Runnable Interface, Difference Between start() and run() Method, Thread Synchronization Techniques, synchronized Keyword & synchronized Blocks, Inter-thread Communication (wait(), notify(), notifyAll()), Deadlock: Causes & Prevention, Using ReentrantLock for Synchronization, Thread Priorities & ThreadGroup, ThreadPool Executor Framework, Callable vs Runnable Interface, Future & CompletableFuture in Java 8, ForkJoin Framework for Parallel Processing, Timer and TimerTask for Scheduling Tasks, Atomic Variables for Lock-Free Synchronization, Java Memory Model & Volatile Keyword, Semaphore & CountdownLatch for Thread Synchronization, CyclicBarrier & Phaser for Coordinating Threads, Parallel Streams in Java 8, Working with Concurrent Collections, Handling InterruptedException, Writing a Multithreaded Java Program, Best Practices for Multithreading.

Module 6 - Collection Framework & Generics : Introduction to Collection Framework, Differences Between Arrays & Collections, Collection Interface & Its Subinterfaces, Iterable, Collection, List, Set, Map Overview; List Implementations: ArrayList, LinkedList, Vector, Stack; Set Implementations: HashSet, LinkedHashSet, TreeSet; Map Implementations: HashMap, LinkedHashMap, TreeMap, Hashtable; Queue & Deque Implementations: PriorityQueue, ArrayDeque, How HashMap Works Internally?, LinkedHashMap vs TreeMap vs HashMap, Sorting Collections using Comparable & Comparator, Synchronization of Collections (Collections.synchronizedList), Fail-Fast vs Fail-Safe Iterators, ConcurrentHashMap vs SynchronizedMap, Performance Comparison of Different Collections, Introduction to Generics, Creating Generic Classes & Methods, Type Parameters in Generics, Wildcards (? extends, ? super), Bounded Type Parameters in Generics, Generics & Type Erasure in

Java, Working with Streams & Collections Together, Using Collections in Real-World Applications, Implementing Custom Collection Classes, Best Practices for Using Collection Framework.

Module 7 - Regular Expressions : Introduction to Regular Expressions, Understanding Pattern & Matcher Classes, Basic Regex Patterns & Syntax, Matching Single Characters & Character Sets, Quantifiers (*, +, ?, {n}), Anchors (^ for start, $ for end), Working with Groups & Capturing Groups, Matching Special Characters (., \d, \s, \w, etc.), Replacing Text Using Regex, Splitting Strings Using Regex, Validation Using Regular Expressions, Email Validation Using Regex, Phone Number Validation Using Regex, Extracting Data from Strings Using Regex, Regex for Date Format Validation, Using Lookaheads & Lookbehinds, Case-Insensitive Matching with Flags, Nested & Complex Regex Patterns, Optimizing Regex Performance, Using Regex in Java Streams & Collections, Finding and Replacing Multiple Patterns, Writing Efficient Regex for Large Text Processing, Using Java's Pattern.compile() & Matcher API, Testing Regex with Online Tools, Best Practices for Writing Maintainable Regex.

Module 8 - Stream API : Introduction to Java Streams, Understanding Stream Pipeline & Processing, Creating Streams from Collections & Arrays, Creating Streams from Files & Strings, Intermediate Operations: filter(), map(), flatMap(), Terminal Operations: forEach(), collect(), reduce(), Sorting Streams using sorted(), Filtering & Transforming Data in Streams, Collecting Data using Collectors.toList(), toSet(), toMap(), Using findFirst(), findAny() for Element Retrieval, Checking Elements with allMatch(), anyMatch(), noneMatch(), Counting Elements with count(), Generating Streams with Stream.generate() & Stream.iterate(), Primitive Streams: IntStream, LongStream, DoubleStream, Mapping with mapToInt(), mapToLong(), mapToDouble(), Converting Streams to Arrays & Lists, Using Optional Class with Streams, Parallel Streams vs Sequential Streams, Performance Considerations of Parallel Streams, Working with Files & I/O Streams using Stream API, Implementing Custom Stream Operations, Debugging Streams Using peek(), Exception Handling in Streams, Writing Custom Collector Functions, Best Practices for Using Streams in Java.

Module 9 – MySQL : Introduction to MySQL, Creating and Managing Databases, SQL Queries: SELECT, INSERT, UPDATE, DELETE, Database Indexing and Optimization, Relationships and Foreign Keys, Using MySQL with Node.js (mysql2, Sequelize), Advanced Query Optimization, Transactions and Rollback Mechanisms, Data Normalization Techniques, Writing Complex Joins and Subqueries, MySQL Performance Tuning and Caching, Working with Stored Procedures, Triggers and Events in MySQL, MySQL User Management and Security Best Practices, Backup and Restore Strategies, Implementing Full-Text Search in MySQL, Scaling MySQL Databases (Replication, Partitioning), Hands-on Practice and Assignments.

Module 10 - JDBC with MySQL : Introduction to JDBC & Database Connectivity, JDBC Architecture & Driver Types, Steps to Connect Java with MySQL, Creating a Database Connection in Java, Executing SQL Queries Using Statement Interface, Using PreparedStatement for Parameterized Queries, Working with CallableStatement for Stored Procedures, Fetching Data Using ResultSet, Handling SQL Exceptions in Java, Batch Processing with JDBC, Understanding Connection Pooling, Using DataSource for Database Connectivity, Performing CRUD Operations Using JDBC, Transactions in JDBC: Commit & Rollback, AutoCommit in JDBC and When to Use It, Setting Fetch Size for Performance Optimization, Handling NULL Values in JDBC, Working with Date & Time in JDBC, Mapping ResultSet to Java Objects, Writing a JDBC Utility Class for Reusability, Logging SQL Queries Using JDBC Logging, Writing Secure JDBC Code to Prevent SQL Injection, Best Practices for Writing Efficient JDBC Code.

Module 11 – HTML : Introduction to HTML & History, HTML Elements and Attributes, Semantic HTML & Importance, HTML Forms and Form Elements, Input Types and Form Validation, HTML Tables and Structuring Data, HTML5 Elements (Audio, Video, Canvas, SVG), HTML Meta Tags and SEO Basics, Creating Multi-page Websites, Links and Navigation Menus, HTML5 Storage (LocalStorage, SessionStorage), HTML Accessibility Features (ARIA), Using HTML for Email Templates, HTML Best Practices and Code Formatting, HTML Performance Optimization, HTML Boilerplate and Frameworks Overview, Introduction to PWA (Progressive Web Apps), HTML with JavaScript Integration, Hands-on Practice and Assignments.

Module 12 – CSS : Introduction to CSS & CSS Syntax, Selectors, Specificity & Inheritance, Box Model and Positioning Techniques, Float, Clear and Overflow Properties, CSS Flexbox and Grid Layouts, Media Queries and Responsive Design, CSS Variables and Custom Properties, CSS Animations and Transitions, Pseudo-elements and Pseudo-classes, CSS Units (px, %, em, rem, vh, vw), Backgrounds, Borders, and Shadows, Text Styling and Typography Best Practices, CSS3 New Features and Properties, Implementing Dark and Light Modes, CSS Preprocessors (SASS, LESS - Introduction), Using Google Fonts and Custom Fonts, CSS Frameworks (Bootstrap, Tailwind - Overview), Debugging CSS Issues, Performance Optimization in CSS, Hands-on Practice and Mini Projects.

Module 13 - JavaScript : Introduction to JavaScript, Variables (var, let, const) and Data Types, Operators and Expressions in JavaScript, Control Flow (if-else, switch-case), Loops (for, while, do-while), Functions (Named, Anonymous, Arrow Functions), Function Scope and Closures, JavaScript Objects and Prototypes, Arrays and Array Methods, String Manipulation and Regular Expressions, Events and Event Listeners, DOM Manipulation and Traversal, Forms Handling and Validation, Error Handling (try-catch, finally), JavaScript Timers (setTimeout, setInterval), JavaScript ES6+ Features

(Destructuring, Spread, Rest), JavaScript Best Practices and Coding Standards, Hands-on Practice and Assignments.

Module 14: Servlet : Introduction to Web Applications & Servlets, Difference Between CGI & Servlets, Servlet Lifecycle: init(), service(), destroy(), Writing & Deploying a Simple Servlet, HttpServlet & Handling HTTP Requests, GET vs POST Methods in Servlets, ServletConfig vs ServletContext, Request & Response Handling in Servlets, Redirecting Requests Using sendRedirect() & forward(), RequestDispatcher & Its Use Cases, Working with Servlet Filters & Interceptors, Implementing Session Management in Servlets, Cookies vs Sessions: Which to Use?, URL Rewriting for Session Management, Understanding Servlet Listeners, File Uploading Using Servlets, Exception Handling in Servlets, Logging & Debugging Servlets, Asynchronous Processing in Servlets, Writing a Custom Servlet Utility Class, Implementing Security in Servlets, Working with Servlet-Based REST APIs, Deploying Servlets in Tomcat Server, Servlet Performance Optimization, Best Practices for Writing Maintainable Servlets.

Module 15 - Java Server Pages (JSP) : Introduction to JSP & Its Importance, Difference Between JSP & Servlets, JSP Lifecycle: Compilation, Initialization, Execution, Writing & Deploying a Simple JSP Page, JSP Scripting Elements: Declarations, Scriptlets, Expressions, Understanding JSP Directives, JSP Implicit Objects & Their Usage, JSP Form Handling & Data Submission, JSP PageContext & Request Attributes, Working with JavaBeans in JSP, JSTL (JavaServer Pages Standard Tag Library), Using JSP Custom Tags for Code Reusability, JSP Expression Language (EL), Implementing MVC Architecture with JSP & Servlets, JSP Exception Handling & Error Pages, Session Management in JSP, Using JSP with Databases (JSP + JDBC), AJAX Integration with JSP for Asynchronous Data Loading, Working with JSP Filters & Interceptors, Securing JSP Applications, JSP Caching for Performance Optimization, Debugging JSP Applications, Deployment of JSP Projects in Tomcat Server, Writing Reusable JSP Templates & Components, Best Practices for Writing Scalable JSP Applications.

Module 16 – Miscellaneous : Introduction to Spring Framework, Dependency Injection (DI) & Inversion of Control (IoC), REST APIs with Spring Boot, Spring Data JPA & Hibernate ORM.

How to Join the Java Full Stack Training Program (6 Months Duration)

At **Learn2Earn Labs Training Institute**, we believe in empowering learners with the right skills to become successful Java Full Stack Developers. To ensure the best outcomes for our students, we welcome applications from individuals who meet the following criteria:

Eligibility Requirements

- **Educational Qualification:** A degree in a relevant domain (e.g., Computer Science, Computer Applications, or any other related degree program).
- **Passion for Learning:** Candidates must demonstrate a strong zeal to become a proficient Java Full Stack Developer.
- **Growth-Oriented Mindset:** We are looking for individuals who are eager to learn, adapt to new challenges, and continuously improve their skills.
- **Commitment and Dedication:** This program is intensive and requires a strong commitment to learning and practical implementation.

How to Apply

1. Visit our Java Full Stack Training Program (Six-Months Duration) page at **https://learntoearnlabs.com/java-full-stack-development-course/** and read the complete details about the training program.
2. Under the **Apply Now** section, fill out the form with your details.
3. Based on your submitted details, a Learn2Earn Labs representative will contact you to further process your application or query.
4. Complete any additional steps, such as interviews, assessments, or telephonic discussions, as requested by the institute representative via email or WhatsApp.

What Happens Next?

- Once your details are reviewed, eligible candidates will be contacted for the next steps, which may include an introductory session or discussion.
- After successful enrollment, you will receive all the program details, including the schedule, enrollment ID, batch ID, terms and conditions, etc., through an enrollment confirmation letter.

Take the first step toward transforming your career! If you meet the eligibility criteria and are passionate about becoming a professional Java Full Stack Developer, **this program is designed for you.** 😊

Join us today and pave the way to a successful career in **Java Full Stack Development**!

LEARN 2 EARN LABS

Best Training Institute for Graduates

Live Online & Offline Classes Available

Join us for Better Career and Package Guarantee

Top Edge Training Programs

- Full Stack Web Development
- Mobile App Development
- Front End Development
- Back End Development
- Java Full Stack
- Data Science & ML
- Digital Marketing

Guaranteed Package (In Writing)*

- **3-5 Lakhs** (With 6 Months Training Programs)
- **5-8 Lakhs** (With 12 Months Training Programs)
- **8 Lakhs+** (With 2 Years Training Programs)

* Terms & Conditions Applied

Amenities

- Digital Notes
- Live Training Sessions
- Project Assistance
- Interview Preparation
- Working Experience
- Job Recommendations
- Professional Development
- Digital Resume & Portfolio

LEARN-2-EARN LABS TRAINING INSTITUTE

Anna Icon Complex, near Kargil Petrol Pump, Sikandra-Bodla Road, Sikandra, Agra
Website : www.LearntoearnLabs.com
Call : 91-9548868337

LEARN 2 EARN LABS

Guiding Careers with Visionary Support

Institute Director(s)

Mr. Mohit Singh
M.Tech, B.Tech (C.S.E)

Mr. Mohit Singh is a professional full-stack trainer, project consultant and startup mentor. He is holding expertise in Java, Application Design, MERN Stack, DevOps, Design Thinking and User Experience Design.
He has trained thousands of students & hundreds of employed professionals. He completed his trainings in Google, Gurugram and short term projects in IIT Delhi, IIT BHU & IIT Jodhpur.
He is also recognized as Mentor with startup India (MAARG), Punjab Startup, startup Uttarakhand, Mumbai State Innovation Society, Atal Innovation Mission, etc. in the area of education & utility services.

https://www.linkedin.com/in/mohit9pages/

Dr. Shubhendra Gupta
Phd, B.Ed, M.Sc (Physics)

Dr. Shubhendra Gupta is an experienced digital marketer, Business Consultant and startup mentor with a demonstrated history of working in the education and services industry.
He use to train students & working professionals for getting better job opportunities and train business owners in generating profits or leads. His areas of interest are Digital Marketing, Business Development, Data Analysis, Strategic Planning, Market Research & Reality, User Testing, Website design, etc.
He is also recognized as Mentor with Startup Hubs & Innovation Labs in the area of education, brand building & business consultation.

https://www.linkedin.com/in/dmshubhendra/

Institute Vision

To be an institute that provides a transformative learning to produce highly skilled & competent professionals and to create leaders and innovators for society and industry.

LEARN-2-EARN LABS TRAINING INSTITUTE

F-4, First Floor, Anna Icon Complex,
near Kargil Petrol Pump, Sikandra-Bodla Road,
Sikandra, Agra, Uttar Pradesh, India

Website : www.LearntoearnLabs.com

Call : 91-9548868337