NAV BHARAT
COMPUTER EDUCATION
H.O. : 13/6C, Nai Sarak, Shastri Nagar, Meerut
(M) 9837991954, 9760091954
B.O. : C-57/6 Kunj Vihar, Jagriti Vihar, Meerut
(M) 0121-2767699, 9258551962

Full Stack Web Development Using Java

HTML

- INTRODUCTION
  - Parts of an HTML Document
  - Version Information
  - Head Section
  - Meta Information
  - Favicons
  - Body Section
  - List
  - Table
  - HTML Forms
  - Anchors and Images
  - Video, Audio
- ADVANCE HTML
  - Introduction
  - HTML5 History
  - New Features and Groups
  - Structure of HTML5 Document
  - Power of HTML5 and Features
  - Semantics and Block
  - Level Elements
  - HTML5 Forms
  - HTML5 Multimedia
- HTML5 Graphics

CSS

- CORE CSS
  - Introduction
  - CSS Basics
  - CSS Syntax
  - CSS Versions
  - CSS Id and Class
  - CSS Styling

- Styling Backgrounds
- Styling Text
- Styling Fonts
- CSS Borders

- ADVANCE CSS
  - Introduction
  - CSS3 Modules
  - Position
  - Box Model
  - Background and Borders
  - Text Effects
  - 2D/3D Transformations
  - Multiple-Column Layout
  - Flex Grid
- User Interface

JavaScript

- Introduction to JavaScript
  - Syntax, Statements, Comments.
  - Syntax, Statements, Comments.
- Enabling in various browsers Popup Boxes
  - Alert, Confirm, Prompt
- Java Script (JS Basics)
  - keywords
  - identifiers
  - datatypes
  - variables and constants
  - token
- Variables Operators
  - Arithmetic
  - Assignment
  - Comparison
  - Logical
  - Conditional
- Conditional Statements
  - If
  - If...else
  - If...else If...else
  - Switch
- Loops
  - While
  - Do...while
  - For
  - For...in Statement

- Break
- Continue

- Function
  - Creating tuple
  - function declarations, definitions and calling
  - types of functions
  - anynomous functions
  - flat arrow functions or arrow functions
  - self invoked function (IIFE)
  - storage classification or variable scope (var let const and without keywords)
  - recursion
  - scope chain rule and lexical scoping
  - closure
- Objects
  - literal and keys
  - object functions
  - nested objects
  - Object Prototypes
  - setter and getter methods
- Set and Map Weak Map and weak Set
- Dates and Time
- OOPS
  - literal and keys
  - object functions
  - nested objects
  - Object Prototypes
  - setter and getter methods
- DOM Manipulation
  - DOM HTML
  - DOM CSS
  - event handling
  - time based events
- Events
  - Introduction
  - Mouse Events
  - Keyboard Events
  - Form Events
  - Document/Window Events
- Page redirection Built-in Objects
  - Introduction
  - Number
  - Strings
  - Arrays
  - Associative Array
  - Array Properties and Methods
  - Date

- Boolean
- RegEx

- Advanced JavaScript
  - Form Validations
  - Basics of Form Validation
  - Validating Radio Buttons
  - Validating Check boxes
  - Validating Select Menus
  - Validating Text areas
  - strict mode
  - closure
  - synchronous and asynchronous programming
- JS Browser Object Model (BOM)
  - JS Cookies
  - JS Window
  - JS Location
  - JS Popups
- Event propagation
  - bubbling
  - capturing
- Data fetching
  - AJAXand JSON
  - Calling API using AJAX
  - Callbacks
  - Promise
  - fetch
  - async await

ReactJS

- Typescript
  - Why Typescript
  - Basic Types
  - Class and Interfaces
  - Modules
- INTRODUCTION TO REACTJS
  - What is ReactJS?
  - What is SPA?
  - DOM vs Virtual DOM
  - Advantages & Disadvantages
  - Key Features
- ENVIRONMENTAL SETUP
  - Node | NPM
  - Installation of CLI
  - Setup Project
  - Directory Structure

- Code Editors
- How ReactJS Application Boot

- BASIC FEATURES OF REACTJS
  - React Concepts
  - JSX and TSX
  - Render Elements
  - Function and Class Components
  - Props and State
- Handling Events
  - Dynamic Data Rendering
  - Property Binding
- KEY FEATURES OF REACTJS
  - Conditional Rendering
  - List and Keys
  - Forms Handling
  - Forms Validations
- COMPONENT LIFE CYCLE HOOK
  - Understanding component life cycle
  - All Life cycle Hooks
- EVENT HANDLING REACT
  - Understanding React Event System
  - Passing arguments to event Handlers
- NETWORK CALL
  - Fetch
  - Axios
- CUSTOM SERVICES
  - Introduction to Services
  - Building a Service
- LOCAL DATA STORAGE
  - Local Storage
  - Session Storage
  - Cookies
- ROUTING WITH REACT ROUTER
  - Setting up React Router
  - Configuring route with Route Component
  - Making routes dynamic with Route Params
  - Working with nested routes
  - Link and NavLink
  - Redirect Routes
- UI COMPONENTS
  - Material Design
  - PrimeNG
- INTRODUCTION TO REDUX
  - Why Redux
  - Install and setup
  - Store

- Reducer
- actions
- Dispatcher
- High order Components
- map State To Props and map Dispatch To Props usage

- ADVANCE REDUX
  - Async Actions
  - Middleware
  - Redux Thunk and Redux Saga
- REACT HOOKS
  - Why We Need HOOKS.
  - Different Types Of Hooks
  - Using State And Effect Hooks
  - Usereducer , Useref Etc.
  - Custom Hooks
  - Rules Of Hooks
- THIRD PARTY MODULES
  - Social Login
  - Pagination
  - Search
  - Filter
  - JWT Token
  - File Upload
  - Many More
- REACTJS TESTING
  - Jest with Enzyme
- Develop a CRUD Application in REACTJS
- REACTJS Application Deployment
  - Build Application and Deployment

SQL

- SQL Using MySQL
  - Introduction to RDBMS
  - What is Relational Database Package
  - Difference between SQL & Database
  - Installing MySQL Server database
- SQL Basic
  - DDL: Create, Alter, Drop, etc.
  - DML: Insert, Update, Delete, etc.
  - DQL : Select
  - Auto_increment field
  - SQL Comments
  - SQL Aliases
  - Savepoint & rollback
- SQL Constraints

- Not NULL, Unique key
- Primary key, Check
- Default, Foreign key
- SQL Operators
- Arithmetic operators
- Logical operators
- Conditional operators
- Like, between, in operators

- SQL Clauses
  - Order by
  - Where
  - Limit/top
  - Group by
  - having
- SQL Joins
  - Inner Join
  - Left Join
  - Right Join
  - Full Join
- SQL View
  - creating view
  - updating view
  - fetching data from view
- SQL Functions
  - String functions
  - Aggregate functions
  - Date & time functions
- Stored Procedures & Functions
  - Understanding stored procedures and their key benefits
  - Working with stored procedures
  - Studying user-defined functions

Core Java

- INTRODUCTION TO JAVA
  - Why Java was Developed
  - Application Areas of Java
  - History of Java
  - Platform Independency in Java
  - USP of Java: Java Features
  - Sun-Oracle Deal
  - Different Java Platforms
  - Difference between JDK,JRE,JVM
  - Java Versions
  - JVM Architecture
  - Installing Java on Windows

- Understanding Path Variable: Why Set Path

- CREATING FIRST JAVA PROGRAM
  - Understanding Text Editors to Write Programs
  - How to compile java file
  - Byte Code and class file
  - How to run class file
- JAVA LANGUAGE FUNDAMENTALS
  - Identifiers
  - Keywords
  - Variables
  - Literals
  - Data Types
  - Operators
  - Comments
  - Looping Statements
  - Condition Statements
  - TypeCasting
- OOP IMPLEMENTATION (PIE)
  - Why OOP
  - OOP Concepts with Real life examples
  - Class & it's Syntax
  - Object & it's Syntax
  - Reference Variable
  - Constructors
  - Instance (Non-Static) & Static Variables
  - Instance(Non-Static) & Static Methods
  - This Keyword and it's usages
  - Object & Static Initializers (Anonymous Blocks)
  - Understanding '+' Operator
  - Inheritance& it's Syntax
  - Types of Inheritance
  - Object Class as Root of Java Class Hierarchy
  - Variable Hiding
  - Method Hiding
  - Method Overriding
  - Method Overloading
  - Super keyword and it's usages
  - Final keyword and it's usages
  - Constructor Chaining
  - Upcasting and Downcasting
  - Static &Dynamic Binding
  - Run Time Polymorphism
  - Abstract Keyword(Abstract classes and methods)
  - Understanding Interfaces
  - Implementation of Encap -sulation
  - Association with Imple -mentation

- PACKAGES
  - Understanding Packages
  - Setting Class path
  - Reading Input from Keyboard
  - Access Modifiers
  - With in Package & Outside Package Implements
- NESTED TYPES
  - Static Nested Class
  - Non-static Nested Class
  - Local Class
  - Anonymous Class
  - Nested Interface
- ARRAYS
  - General Definition of Array
  - Advantages from Array
  - Arrays in Java
  - 1-d Arrays
  - 2-d Arrays
  - Jagged Arrays
  - Array of reference type
  - Operations on Arrays
  - User Define Array & Object Type
- COMMAND LINE ARGUMENTS AND WRAPPER CLASSES
  - How to read command line arguments
  - Wrapper Classes
  - Parsing of Numeric Strings
  - String representation of Primitives
- EXCEPTION HANDLING
  - Types of Runtime Errors
  - Understanding Exceptions
  - Exception Class Hierarchy
  - Try & Catch Blocks
  - Patterns of Catch Block
  - Nested Try statements
  - Throw, throws and finally
  - Creating Custom Exceptions
  - Checked & Unchecked Exceptions
  - Assertion
- WORKING WITH STRINGS
  - What is String?
  - String Class
  - Creating String Object
  - Operations on String
  - String Buffer Class and it's Methods
  - Difference between String and StringBuffer class
  - String Builder Class and it's Methods

- Difference between String Buffer and String Builder

- SWING
  - Introduction to AWT
  - Introduction to Swing Components
  - Look And Feel of Swing Components
  - MVC Architecture of Swing Components
  - Working with Image
  - Advance Swing Components
  - OptionPane, Jtree, Jtable, JTabbedPane
  - Jfile Chooser, Jcolor Choose
- Menu Components
  - JMenu
  - JMenuItem
  - JMenubar
- MULTITHREADED PROGRAM -MING
  - Multitasking: Why Concurrent Execution?
  - Multiprocessing v/s Multithreading
  - Main Thread (Default Java Thread)
  - Creating Child Threads and understanding context switching
  - Thread States
  - Thread Group
  - Thread Synchronization: Methods and Blocks
  - Inter-Thread communication
  - Daemon Threads
  - Deadlock
- I/O STREAMS
  - What is I/O?
  - Why Need Streams?
  - Byte Streams and Character Streams
  - Read/Write operations with file
  - Scanner Class
  - Object Serialization & Deserialization
  - Transient keyword
  - File Class and it's Methods
- SOCKET PROGRAMMING
  - Understanding Fundamentals of a Network
  - Socket and Server Socket Classes
  - Inet Address Class
  - Datagram Socket and Datagram Packet Classes
  - URL, URL Connection, Http URL Connection Classes
- REFLECTION
  - Understanding the Need Of Reflection
  - Getting information about class's modifiers, fields, methods, constructors and super classes
  - Finding out constant and method declaration belong to an interface
  - Creating an instance of the class whose name is not known until runtime

- Getting and setting values of an object's field if field name is unknown until runtime
- Invoking a method on an object if the method is unknown until runtime
- Invoking Private Methods

- EXTENDED & UTILITY CONCEPTS
  - Generics
  - Lambda Expression
  - Annotations
  - Object Cloning
  - Vargs
  - Static-import
  - Enum
  - Static, Default and Private Methods of Interface
  - Var Type
  - Java Modules
  - Stream API
- INTRODUCTION TO SQL (PROJECT BASED)
- DATABASE PROGRAMMING USING JDBC
  - Need Of JDBC
  - JDBC Drivers Comparable Interfaces
  - Statement, PreparedStatement, CallableStatement
  - Scrollable and Up datable Result Set
  - Batch Updates
  - Transaction
  - Metadata
  - Connection Data Base
  - Oracle
  - My SQL
- Mongo DB

Advance Java

- COLLECTIONS FRAMEWORK
  - What is Collection?
  - What is Framework?
  - Collections Framework
  - Core Interfaces
  - Collection, List, Queue, Deque
  - Set, Navigable Set, Sorted Set
  - Map, Navigable Map, Sorted Map
  - Core Classes
  - Array List, Linked List, Priority Queue, Array Deque
  - Hash Set, Linked Has Set, Tree Set,
  - Hash Map, Identity Hash Map, Weak Hash Map, Linked Hash Map, Tree Map
  - Accessing a Collection via an Iterator
  - Accessing List via List Iterator
  - Accessing a Collection via for each loop

- Working with User Defined Objects
- The Comparator and
- The Legacy classes and Interfaces.
- Enumeration, Vector ,Stack
- Hash table, Properties

- DATE & TIME API
  - java.util.Date
  - java.util.Calender
  - java.sql.Date
- SYSTEM PROPERTIES & INTERNATIONALIZATION (I18N)
  - Understanding Locale
  - Resource Bundle
  - Usage of properties file
  - Fetching text from Resource Bundle
  - Displaying the text in HINDI
  - Displaying date in HindiJAVA
- EE(JAVA PLATFORM ENTERPRISE EDITION)
  - Understanding the Concept of Java EE:JEE Specification
  - Java EE Architecture
    i) Single Tier
    ii) Two Tier
    iii) Three Tier
    iv) N-Tier
  - Java EE Components
  - Web Components
  - Distributed(Business) Components
  - Java EE Containers& Servers
- WEB CONTAINER & WEB SERVER (APACHE TOMCAT)
  - EJB Container& Application Server (Web logic, Glass fish, Web sphere)
  - Java EE Services
    i) JNDI Service
    ii) Java Transaction Service
    iii) JAAS
    iv) JMS
- JAVA SERVLET
  - Introduction to web program- ming
  - File Uploading and Down -loading
  - Session Tracking & State Management
  - Cookie
  - Url Rewriting
  - Hidden F+orm Field
  - Session Object
  - Events & Listeners
  - Dependency Injection
  - Refreshing Servlet
  - Filters
- JAVA SERVER PAGES & JSTL

- JSP Architecture
- JSP Elements
- JSP Objects
- Understanding Java Beans
- Custom Tags
- Using tags of JSTL
- Expression Language

- PROJECT CLASSES
  - Front End Coding
  - FORM DESIGNING
    - i) HTML & CSS
    - ii) JAVA SCRIPT
    - iii) BOOTSTRAP
  - Back End Coding
  - DATABASE DESIGNING
  - Connecting forms to database
  - Writing Business Logic
  - Project Hosting
- DESIGN PATTERN
  - Why Design Patterns...?
  - Front Controller
  - Composite View
  - Factory Pattern
  - Singleton Pattern
  - DAO Pattern
- JAVA MAIL API
  - Email System and Protocols
  - Sending & Receiving Mails
  - Handling Attachments
- INTRODUCTION TO DISTRIBUTED PROGRAMMING
  - RMI
  - Web Services
- INTRODUCTION TO REST FULL SERVICES
  - @PathParam
  - @Path
  - @FormParam
  - @QueryParam
  - @DefaultValue
- OVERVIEW OF JPA FRAMEWORK

Spring

- INTRODUCTION
  - What is Spring?
  - Spring modules
  - Understanding dependency Injection

- Applying aspect - oriented programming
- MAVEN DEPLOYMENT
  - Maven Configuration
  - Converting Maven to Eclipse
  - Various Maven Command
- BASIC BEAN WIRING
  - Containing your Bean
  - Creating bean
  - Injecting into bean properties
  - Auto wiring
  - Controlling bean creation
  - Aspect Oriented Programming
- HITTING THE DATABASE
  - Learning Spring's data Access Philosphy
  - Configuring a data source
  - Using JDBC with Spring
  - Working with JDBC Templates
  - Using Spring's DAO Support Classes for JDBC
  - Integrating Hibernate with Spring
- Caching

Hibernate

- INTRODUCTION TO ORM
  - Need of ORM
  - Problems using JDBC Directly
  - ORM Implementation
- INTRODUCTION TO HIBERNATE
  - Hibernate Architecture
  - Hibernate configuration
  - Hibernate's Support for Other
  - Technologies
  - Installing Hibernate
  - A "Hello world" stand alone application
- CREATING PERSISTING CLASSES
  - Mapping a basic Java Class
  - Mapping a Class with Binary Data
  - Mapping a Serializable Class
  - Mapping a class with Data/ calendar attributes
  - Mapping a Read-only class
  - Mapping a class using Versioning /Time stamps
- MAPPING IN HERITENCE WITH JAVA CLASSES
  - Table-Per –class Hierarchy Mapping
  - Table - Per –subclass Hierarchy Mapping
  - Table -Per –concrete -subclass Hierarchy Mapping
  - Persistence interfaces

- WORKING WITH COLLECTIONS
  - Associations
  - Lazy initialization
  - Mapping Maps/Sorted Maps
  - Mapping Sets /Sorted Sets
  - Mapping lists
  - Mapping Arrays
  - Mapping a Bidirectional Association
- HIBERNATE CACHING
  - How caching improves performence
- First, Second level cache

SpringBoot

- Introduction to Spring Boot
- Spring Boot Annotation
- Spring Boot & JDBC Template
- Spring Boot & JPA Hibernate
- Spring Boot Rest API
- Spring Boot MVC
- Spring Boot Security
- Introduction to Micro Services