

Contact: 9084516525



ExtraByte Technologies  
Private Limited

ExtraByte Academy, A unit of ExtraByte Technologies Pvt Ltd registered under companies act, 2013. ExtraByte Academy provides educational programs of Coding, Web Designing and Web Development.

We create confident developers who think beyond industrial jobs and march their ideas into self-created entrepreneurship ventures. Skill and innovative thinking give developers the confidence to transform their ideas into real life products.

We aim to accomplish our mission by providing the right people with great skills and complete dedicated team of professionals who are working hard to build expert manpower to fulfill our aim of contributing to the nation by producing competent IT professionals.

## **Full Stack Web Dev And Data Science Bootcamp (Course Detail)**

---

**Course Duration** : 2 Year

**Class Hours** : 1000 Hours (2 Hour/Day)

**Total Projects** : 200+ Projects

**What You Achieve/Goal** : Creating Web application using web based Languages.

**Focus On Practical** : **Total 200+ major or minor projects.** We focus on practical.

### **About Course**

A full-stack web development course covering languages and frameworks like Tailwind, SASS, DSA, React, Next.js, Laravel, Node.js, Flask, ORM, Data Science, and GitHub offers a comprehensive journey into the world of modern web development. Students will master both front-end and back-end technologies, starting with the styling and layout techniques of Tailwind and SASS, which enhance user interface design. They'll gain expertise in React and Next.js for building dynamic, high-performance web applications, while also diving into back-end frameworks such as Laravel, Node.js, and Flask for developing scalable, server-side applications. Additionally, they'll learn about Object-Relational Mapping (ORM) to manage databases efficiently, and explore Data Science for building robust platforms. The course also emphasizes Data Structures and Algorithms (DSA), equipping students with the problem-solving skills needed for coding challenges and efficient app performance. Throughout the program, GitHub will be utilized for version control, ensuring students are well-versed in collaborative development practices. This all-encompassing approach prepares individuals to tackle real-world full-stack development projects with confidence.

## Full Stack Web Dev And Data Science Bootcamp : Index

Chapter Number	Module	Chapter Number	Module
<b>Module 1</b>	Element Of Computing System	<b>Module 22</b>	NextJS
<b>Module 2</b>	Operating System & Browsers	<b>Module 23</b>	Ninth Live Project With NextJS
<b>Module 3</b>	HTML5	<b>Module 24</b>	Adobe Photoshop
<b>Module 4</b>	CSS3 And Interactive Design	<b>Module 25</b>	Adobe Illustrator And Logo Designing
<b>Module 5</b>	First Live Project With HTML5 & CSS3	<b>Module 26</b>	PHP
<b>Module 6</b>	Bootstrap & Responsive Websites	<b>Module 27</b>	Database With MySQL/MongoDB
<b>Module 7</b>	Second Live Project With Bootstrap	<b>Module 28</b>	Tenth Live Project With PHP & Database
<b>Module 8</b>	Tailwind CSS	<b>Module 29</b>	Laravel Framework
<b>Module 9</b>	Third Live Project With Tailwind CSS	<b>Module 30</b>	Eleventh Live Project With Laravel
<b>Module 10</b>	SASS	<b>Module 31</b>	NodeJS Design Patterns
<b>Module 11</b>	Fourth Live Project With SASS	<b>Module 32</b>	Twelfth Live Project With NodeJS
<b>Module 12</b>	JavaScript Core	<b>Module 33</b>	Python Core
<b>Module 13</b>	JavaScript Advanced	<b>Module 34</b>	Python Advanced
<b>Module 14</b>	JQuery & XML	<b>Module 35</b>	Thirteenth Live Project With Python
<b>Module 15</b>	Fifth Live Project With JavaScript & JQuery	<b>Module 36</b>	Flask
<b>Module 16</b>	Typescript	<b>Module 37</b>	ORM(Object Relational Mapping)
<b>Module 17</b>	Sixth Live Project With Typescript	<b>Module 38</b>	Fourteenth Live Project With Flask
<b>Module 18</b>	DSA With JavaScript	<b>Module 39</b>	Github
<b>Module 19</b>	Seventh Live Project With DSA JavaScript	<b>Module 40</b>	Data Science
<b>Module 20</b>	React JS	<b>Module 41</b>	Fifteenth Live Project With Data Science
<b>Module 21</b>	Eighth Live Project With ReactJS	<b>Module 42</b>	Final Project

**Additional Module : Job Interview Preparation**

# **Module 1 : Element Of Computing System**

## **Unit 1: Knowing Computer -**

What is Computer, Basic Applications of Computer; Components of Computer System, Central Processing Unit (CPU), VDU, Keyboard and Mouse, Other input/output Devices, Computer Memory, Concepts of Hardware and Software; Concept of Computing, Data and Information; Applications of IECT; Connecting keyboard, mouse, monitor and printer to CPU and checking power supply

## **Unit 2: Communications and collaboration -**

Basics of electronic mail; Getting an email account; Sending and receiving emails; Accessing sent emails; Using Emails

## **Unit 3: Number Systems -**

Base or radix; Positional number system, Popular number systems – Decimal, Binary, Number systems – Octal and Hexadecimal, Conversion-From one number system to another, Concept of binary addition, Concept of subtraction, Complements in binary number systems-1s Complement 2s Complement, Applications of Complements, BCD numbers – concept

## **Unit 4: Boolean Algebra and Gate Networks -**

Logic gates- AND, OR – Truth tables and graphical representation, Logic gates- NOT, NAND and NOR Truth tables and graphical representation, Basic laws of Boolean Algebra, Simplification of Expressions, De Morgans theorems, Dual expressions, Canonical expressions, expressions using NAND/NOR Gates, Don't care conditions, XOR and its applications

# **Module 2 : Operating System & Browsers**

## **Unit 1: Introduction to Operating Systems -**

What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure

## **Unit 2: Introduction to Internet, WWW and Web Browsers -**

Basic of Computer networks; LAN, WAN; Concept of Internet; Applications of Internet; connecting to internet; What is ISP; Knowing the Internet; Basics of internet connectivity related troubleshooting, World Wide Web; Web Browsing softwares, Search Engines; Understanding URL; Domain name; IP Address; Using e-governance website

## **Unit 3: Processes and Threads -**

Processes, threads, interprocess communication, scheduling, IPC problems

## **Unit 4: Memory Management -**

No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation

## **Unit 5: File Systems -**

Files, directories, file system implementation, file-system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system

## **Unit 6: Input-Output -**

Principles of I/O hardware, Principles of I/O software, I/O software layers, disks, clocks, user interfaces: keyboard, mouse, monitor, thin clients, power management

## **Unit 7: Deadlocks -**

Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, issues

## **Unit 8: Virtualization and Cloud -**

History, requirements for virtualization, type 1 and 2 hypervisors, techniques for efficient virtualization, hypervisor microkernels, memory virtualization, I/O virtualization, Virtual appliances, virtual machines on multicore CPUs, Clouds

## **Unit 9: Multiple Processor Systems -**

Multiprocessors, multicomputers, distributed systems

## **Unit 10: Case Study on LINUX and ANDROID -**

History of Unix and Linux, Linux Overview, Processes in Linux, Memory management in Linux, I/O in Linux, Linux file system, security in Linux. Android

## **Unit 11: Case Study on Windows -**

History of windows through Windows 10, programming windows, system structure, processes and threads in windows, memory management, caching in windows, I/O in windows, Windows NT file system, Windows power management, Security in windows.

# **Module 3 : HTML5**

## **Unit 1: Web Programming Introduction -**

Web Development Introduction

## **Unit 2: HTML-Introduction -**

History of HTML, What you need to do to get going and make your first HTML page, What are HTML Tags and Attributes?, HTML Tag vs. Element, HTML Attributes, How to differentiate HTML Document Versions

## **Unit 3: HTML-Basic Formatting Tags -**

HTML Basic Tags, HTML Formatting Tags, HTML Color Coding

## **Unit 4: HTML-Grouping Using Div Span -**

Div and Span Tags for Grouping

## **Unit 5 : HTML-Images -**

Image and Image Mapping

## **Unit 6 : HTML-Hyperlink -**

URL - Uniform Resource Locator, URL Encoding

### **Unit 7: HTML-Table -**

<table>, <th>, <tr>, <td>, <caption>, <thead>, <tbody>, <tfoot>, <colgroup>, <col>

### **Unit 8: HTML-Iframe -**

Attributes Using, Iframe as the Target

### **Unit 9: HTML-Form -**

<input>, <textarea>, <button>, <select>, <label>

### **Unit 10: HTML-Headers -**

Title, Base, Link, Style s, Script, Meta

### **Unit 11: HTML-Miscellaneous -**

HTML Meta Tag, XHTML, HTML Depreciated Tags & Attributes

## **Module 4 : CSS3 And Interactive Design**

### **Unit 12: CSS3-Introduction -**

Benefits of CSS, CSS Versions History, CSS Syntax, External Style Sheet using <link>, Multiple Style Sheets, Value Lengths and Percentages

### **Unit 13: CSS3-Syntax -**

CSS Syntax, single Style Sheets, Multiple Style Sheets, Value Lengths and Percentages

### **Unit 14: CSS3-Selectors -**

ID Selectors, Class Selectors, Grouping Selectors, Universal Selector, Descendant / Child Selectors, Attribute Selectors, CSS – Pseudo Classes

### **Unit 15: CSS3-Color Background Cursor -**

background-image, background-repeat, background-position, CSS Cursor

### **Unit 16: CSS3-Text Fonts -**

Color, background-color, text-decoration, text-align, vertical-align, text-indent, text-transform, white-space, letter-spacing, word-spacing, line-height, font-family, font-size, font-style, font-variant, font-weight

### **Unit 17: CSS3-Lists Tables -**

list-style-type, list-style-position, list-style-image, list-style, CSS Tables, border, width & height, text-align, vertical-align, padding, color

### **Unit 18: CSS3-Box Model -**

Borders & Outline, Margin & Padding, Height and width, CSS Dimensions

### **Unit 19: CSS3-Display Positioning:**

CSS Visibility, CSS Display, CSS Scrollbars, CSS Positioning, Static Positioning, Fixed Positioning, Relative Positioning, Absolute Positioning, CSS Layers with Z-Index

## **Unit 20: CSS Floats -**

The float Property, The clear Property, The clearfix Hack

## **Unit 21: CSS3-Animation**

# **Module 5 : First Live Project With HTML5 & CSS3**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS. The trainers enable to students to face the real challenges during the process of making web application.

# **Module 6 : Bootstrap & Responsive Websites**

## **Unit 1: Overview of Bootstrap -**

Introduction of Bootstrap, Syntax of Bootstrap, Container and Container-fluid, Connectivity of Bootstrap in page

## **Unit 2: Bootstrap Component -**

Jumbotron, Button, Grid, Table, Form, Alert, Wells, Badge and label, Panels, Pagination, Pager, Image, Glyphicon, Carousel, Progress Bar, List Group, Dropdown, Collapse

## **Unit 3: Bootstrap Advance Component -**

Tabs/Pill, Navbar, Input Types, Modals, Popover, Scrollspy

## **Unit 4: Bootstrap Utilities -**

Bootstrap Border, Bootstrap Clearfix, Bootstrap Close Icons, Bootstrap Colors, Display Flexbox, Display Property, Image Replacement, Invisible Content, Bootstrap Position, Responsive helpers, Screen Readers, Bootstrap sizing, Bootstrap spacing, Bootstrap Typography

# **Module 7 : Second Live Project With Bootstrap**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS and Bootstrap. The trainers enable to students to face the real challenges during the process of making web application.

# **Module 8 : Tailwind CSS**

## **Unit 1: Introduction to Tailwind CSS -**

Why Use Tailwind CSS?, Key Advantages of Tailwind CSS, Why Choose Tailwind Over Other CSS Frameworks?, Installing and Using Tailwind CSS in a Project, Using Tailwind CSS via CDN

## **Unit 2: Tailwind CSS Layout-**

Tailwind CSS Container, Tailwind CSS Box Sizing, Tailwind CSS Display, Tailwind CSS Float, Tailwind CSS Clear, Tailwind CSS Object Fit, Tailwind CSS Object Position, Tailwind CSS Overflow, Tailwind CSS overscroll Behavior, Tailwind CSS Position, Tailwind CSS Top/Right/Bottom/Left, Tailwind CSS Visibility, Tailwind CSS Z-index

### **Unit 3: Tailwind CSS Flexbox -**

Tailwind CSS Flex Direction, Tailwind CSS Flex Wrap, Tailwind CSS Flex, Tailwind CSS Flex Grow, Tailwind CSS Flex Shrink, Tailwind CSS Order

### **Unit 4: Tailwind CSS Grid -**

Tailwind CSS Grid Template Columns, Tailwind CSS Grid Column Start / End, Tailwind CSS Grid Template Rows, Tailwind CSS Grid Row Start / End, Tailwind CSS Grid Auto Flow, Tailwind CSS Grid Auto Columns, Tailwind CSS Grid Auto Rows, Tailwind CSS Gap

### **Unit 5: Tailwind CSS Alignment -**

Tailwind CSS Justify Content, Tailwind CSS Justify Items, Tailwind CSS Justify Self, Tailwind CSS Align Content, Tailwind CSS Align Items, Tailwind CSS Align Self, Tailwind CSS Place Content, Tailwind CSS Place Items, Tailwind CSS Place Self

### **Unit 6: Tailwind CSS Spacing -**

Tailwind CSS Padding, Tailwind CSS Margin, Tailwind CSS Space Between

### **Unit 7: Tailwind CSS Sizing -**

Tailwind CSS Width, Tailwind CSS Min-Width, Tailwind CSS Max-Width, Tailwind CSS Height, Tailwind CSS Min-Height, Tailwind CSS Max-Height

### **Unit 8: Tailwind CSS Typography-**

Tailwind CSS Font Family, Tailwind CSS Font Size, Tailwind CSS Font Smoothing, Tailwind CSS Font Style, Tailwind CSS Font Weight, Tailwind CSS Font Variant Numeric, Tailwind CSS Letter Spacing, Tailwind CSS Line Height, Tailwind CSS List Style Type, Tailwind CSS Placeholder Color, Tailwind CSS Placeholder Opacity, Tailwind CSS Text Alignment, Tailwind CSS Text Color, Tailwind CSS Text Opacity, Tailwind CSS Text Decoration, Tailwind CSS Text Transform, Tailwind CSS Vertical Alignment, Tailwind CSS Whitespace, Tailwind CSS Word Break

### **Unit 9: Tailwind CSS Backgrounds -**

Tailwind CSS Background Image, Tailwind CSS Background Clip, Tailwind CSS Background Color, Tailwind CSS Background Opacity, Tailwind CSS Background Position, Tailwind CSS Background Repeat, Tailwind CSS Background Size, Tailwind CSS Gradient Color Stops

### **Unit 10: Tailwind CSS Borders -**

Tailwind CSS Border Radius, Tailwind CSS Border Width, Tailwind CSS Border Color, Tailwind CSS Border Opacity, Tailwind CSS Border Style, Tailwind CSS Divide Width, Tailwind CSS Divide Color, Tailwind CSS Divide Opacity, Tailwind CSS Divide Style, Tailwind CSS Ring Width, Tailwind CSS Ring Color, Tailwind CSS Ring Opacity, Tailwind CSS Ring Offset Width, Tailwind CSS Ring Offset Color

### **Unit 11: Tailwind CSS Effects -**

Tailwind CSS Box Shadow, Tailwind CSS Opacity

### **Unit 12: Tailwind Filters -**

Tailwind CSS Filter, Tailwind CSS Blur, Tailwind CSS Brightness, Tailwind CSS Contrast, Tailwind CSS Drop Shadow, Tailwind CSS Grayscale, Tailwind CSS Hue Rotate, Tailwind CSS Invert, Tailwind CSS Saturate, Tailwind CSS Sepia, Tailwind CSS Backdrop Filter, Tailwind CSS Backdrop Blur, Tailwind CSS Backdrop Brightness, Tailwind CSS Backdrop Contrast, Tailwind CSS Backdrop Grayscale, Tailwind CSS

Backdrop Hue Rotate, Tailwind CSS Backdrop Invert, Tailwind CSS Backdrop Opacity, Tailwind CSS Backdrop Saturate, Tailwind CSS Backdrop Sepia

### **Unit 13: Tailwind CSS Tables-**

Tailwind CSS Border Collapse, Tailwind CSS Table Layout

### **Unit 14: Tailwind CSS Transitions and Animation -**

Tailwind CSS Transition Property, Tailwind CSS Transition Duration, Tailwind CSS Transition Timing Function, Tailwind CSS Transition Delay

### **Unit 15: Tailwind CSS Transforms-**

Tailwind CSS Transform, Tailwind CSS Transform Origin, Tailwind CSS Scale, Tailwind CSS Rotate, Tailwind CSS Translate

### **Unit 16: Tailwind CSS Interactivity-**

Tailwind CSS Appearance, Tailwind CSS Cursor, Tailwind CSS Outline, Tailwind CSS Pointer Events, Tailwind CSS Resize, Tailwind CSS User Select

### **Unit 17: Tailwind CSS SVG-**

Tailwind CSS Fill, Tailwind CSS Stroke, Tailwind CSS Stroke Width

## **Module 9 : Third Live Project With Tailwind CSS**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML and Tailwind CSS. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 10 : SASS**

### **Unit 1: Introduction to SASS -**

What is SASS? (Overview of CSS preprocessors), Advantages of using SASS over regular CSS, Setting up the development environment (installing SASS with Node.js, using a task runner like Gulp or Webpack), Understanding the SASS file structure (.scss vs .sass).

### **Unit 2: Variables -**

Defining and using variables in SASS, Storing colors, fonts, measurements, and other reusable values, Nested variables and scope.

### **Unit 3: Nesting -**

How to nest CSS rules inside one another in SASS, Avoiding deep nesting (best practices), Parent selectors (&) and using them effectively.

### **Unit 4: Partials and Imports -**

Breaking your CSS into smaller, modular files (partials), The @import rule to include partials, Using @use and @forward for better import management in newer SASS versions.



## **Unit 5: Mixins -**

Defining mixins for reusable CSS rules, Passing arguments to mixins, Default values for mixin arguments, Conditional logic within mixins.

## **Unit 6: Inheritance and Extends -**

Using the @extend directive to inherit styles, How inheritance works in SASS and its potential pitfalls, The difference between @extend and mixins.

## **Unit 7: Functions -**

Creating and using custom functions in SASS, Built-in SASS functions (e.g., lighten(), darken(), rgb(), etc.), Returning values from functions.

## **Unit 8: Loops and Control Directives -**

Using @for, @each, and @while loops, Control structures like @if and @else, Iterating over lists, maps, and ranges.

## **Unit 9: Lists and Maps -**

Working with lists (arrays) in SASS, Accessing list items and modifying them, Using maps (key-value pairs) for more complex data storage, Iterating over maps.

## **Unit 10: Math and Operations -**

Basic mathematical operations in SASS (addition, subtraction, multiplication, division), Calculating values dynamically (e.g., calculating widths, margins, etc.), Using functions like mod(), percentage(), and ceil() to manipulate data.

## **Unit 11: SASS in the Real World -**

Organizing SASS projects (folder structures, naming conventions), Best practices for using SASS in larger projects, Performance considerations (minification, compilation, etc.), Integration with build tools (Webpack, Gulp, or Grunt).

## **Unit 12: Advanced Topics -**

Media queries in SASS (e.g., creating reusable breakpoints), Advanced mixins (mixins with complex logic, dynamic arguments), Creating design systems with SASS (modular, reusable components), Using SASS with frameworks (e.g., Bootstrap, Foundation).

## **Module 11 : Fourth Live Project With SASS**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS and SASS. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 12 : JavaScript Core**

### **Unit 1: Fundamentals of JavaScript Code -**

JavaScript methods (such as alerts), Variables, The importance of quotes, Numbers vs. strings, Concatenation

## **Unit 2: Reusing Code with Functions -**

Defining functions, Calling functions, Defining parameters & passing arguments

## **Unit 3: Simple Accordion with JavaScript -**

Targeting elements by ID, Hiding & showing elements with JavaScript

## **Unit 4: If Statements: Clearing Form Fields -**

Testing code in the JavaScript Console, Getting & setting properties, Using if statements, Reshowing text hints in empty form fields

## **Unit 5: Introduction to JavaScript Objects & the DOM -**

Intro to objects, The global object, Breaking open & manipulating objects

## **Unit 6: Dynamically Changing Content with Custom Objects -**

Checking the functionality of the select menu, Getting the chosen value, Dynamically changing the state name value, Dynamically changing the rest of the values

## **Unit 7: Introduction to Arrays & the Math Object -**

Creating an array, Editing an array, Creating an array of welcome headings, The Math object, Using the Math object to pick random headlines

## **Unit 8: Introduction to For Loops -**

Creating a for loop, Using the for loop to set dropdown menus, Clearing the To menu, Optional bonus: refining the menu selection experience

## **Unit 9: Exploring JavaScript Selectors -**

Intro to JavaScript selectors, Selecting multiple elements & elements without IDs, Getting a specific list item, Getting elements by class name, Getting multiple items using `querySelectorAll()`, Getting a single item using `querySelector()`, Chaining selectors together, Targeting elements by data attribute

## **Unit 10: HTML Forms And HTML DOM**

## **Unit 11: Cookies -**

Working with cookies

## **Unit 12: Working with Objects and Classes -**

Working with Objects, Call method in JavaScript, Inheritance in JavaScript using prototype

# **Module 13 : JavaScript Advanced**

## **Unit 1: OOP's -**

Class, object, Inheritance, static

## **Unit 2: Errors and Exceptions -**

SyntaxErrors, Runtime Errors , Logical Errors, The try...catch...finally Statement , The throw Statement , The onerror( ) Method

### **Unit 3: Form Validation -**

Basic Form Validation, Data Format Validation

### **Unit 4: Animation -**

Manual Animation, Automated Animation, Rollover with a Mouse Event

### **Unit 5: Multimedia -**

Checking for Plug-Ins, Controlling Multimedia

### **Unit 6: Debugging -**

Error Messages in IE, Error Messages in Firefox or Mozilla Notifications, How to Debug a Script, Useful Tips for Developers

## **Module 14 : JQuery & XML**

### **JQuery Course :**

#### **Unit 1: Introduction -**

What You Should Already Know, What is jQuery ?, Adding the jQuery Library to Your Pages, Basic jQuery Example, Downloading jQuery, Alternatives to Downloading, jQuery Syntax, The Document Ready Function, How to use Custom Scripts?, Using Multiple Libraries, jQuery – noConflict() Method

#### **Unit 2: JQuery – Basics -**

String, Numbers, Boolean, Objects, Arrays, Functions, Arguments, Scope, Built-in Functions

#### **Unit 3: JQuery – Selectors -**

jQuery – CSS Element Selector and ID Selector, jQuery – CSS Element Class Selector and Universal Selector, jQuery – CSS Multiple Elements E/F/G Selector, jQuery Callback Functions

#### **Unit 4: JQuery – DOM Attributes -**

Get Attribute Value, Set Attribute Value

#### **Unit 5: JQuery – DOM Traversing -**

Find Elements by index, Filtering out Elements, Locating Descendent Elements, JQuery DOM Traversing Methods

#### **Unit 6: JQuery – CSS Methods -**

Apply CSS Properties and Multiple CSS Properties, Setting Element Width & Height, JQuery CSS Methods

#### **Unit 7: JQuery – DOM Manipulation Methods -**

Content Manipulation, DOM Element Replacement, Removing DOM Elements, Inserting DOM elements, DOM Manipulation Methods, Binding event handlers, Removing event handlers Event Types, The Event Object and Attributes

#### **Unit 8: JQuery – Effects -**

JQuery Effect Methods/Hide and Show, jQuery Toggle, jQuery Slide – slideDown, slideUp, slideToggle, jQuery Fade – fadeIn/fadeOut/fadeTo, jQuery Custom Animations

## **Unit 9: JQuery – Ajax and JSON -**

Ajax with jQuery, Load method, jQuery get and getJSON methods, jQuery POST request, Retrieving js file, Helper methods

## **XML Course –**

### **Unit 1: XML Tutorial -**

XML Introduction, What is XML, XML Features, HTML vs XML, XML Example, XML Technologies, XML Attributes, XML Comments, XML Tree

### **Unit 2: XML Validation -**

XML Validation, XML DTD, XML CSS, XML Schema, DTD vs XSD, CDATA vs PCDATA

Unit 3: XML Advance:

XML Parsers, XML DOM, XML Database, XML Namespaces

## **Module 15 : Fifth Live Project With JavaScript & JQuery**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS, JavaScript and JQuery. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 16 : TypeScript**

### **Unit 1: Introduction to TypeScript –**

What is TypeScript? (Differences between JavaScript and TypeScript), Setting up TypeScript (Installation via npm, creating a tsconfig.json file), Compiling TypeScript to JavaScript (Using tsc command), Understanding TypeScript's type system (static typing in JavaScript), Benefits of using TypeScript in large-scale projects.

### **Unit 2: Basic Types in TypeScript –**

Primitive types (string, number, boolean, null, undefined), Type inference (TypeScript automatically inferring types), Type annotations (Explicitly declaring types for variables), Type Aliases and const assertion (as const).

### **Unit 3: Functions and Type Annotations –**

Function types and how to annotate them, Optional and default parameters, Function return types, The void and never types, Arrow functions with types.

### **Unit 4: Objects and Interfaces –**

Defining and working with object types, TypeScript interface vs. type alias, Optional and readonly properties in interfaces, Extending interfaces and inheritance, Index signatures and dynamic property names.

### **Unit 5: Arrays and Tuples –**

Array types in TypeScript (declaring array types with [] or Array<Type>), Tuples (arrays with fixed types and lengths), Destructuring arrays and tuples, Multi-dimensional arrays and nested tuples.

## **Unit 6: Enums –**

What are Enums? (Numeric and String enums), Declaring and using Enums, Reverse mapping in numeric enums, Const enums for performance optimization.

## **Unit 7: Type Assertion and Type Casting –**

Type assertion syntax (as and angle bracket < >), When and why to use type assertions, Type assertion vs. type casting in JavaScript.

## **Unit 8: Type Inference and Type Compatibility –**

How TypeScript infers types, Type compatibility and structural subtyping, Literal types and their impact on type inference, Type widening and narrowing, The unknown type and type guards.

## **Unit 9: Generics –**

What are generics and why are they useful? Defining generic functions, interfaces, and classes, Generic constraints (using extends to constrain types), Working with generic types in functions and methods, Default generic values and default constraints.

## **Unit 10: Advanced Types –**

Union types and intersection types, Literal types and type narrowing, Discriminated unions (tagged union types), Conditional types (using extends in type conditions), Mapped types and key remapping, Type guards and custom type guards.

## **Unit 11: Classes and Object-Oriented Programming (OOP) in TypeScript –**

Defining classes and creating objects, Inheritance and subclassing, Access modifiers (public, private, protected), Getters and setters, Abstract classes and methods, Interfaces with classes, Static properties and methods.

## **Unit 12: Modules and Namespaces –**

Introduction to modules in TypeScript, Using export and import statements, Working with third-party libraries using @types or DefinitelyTyped, ES6 modules in TypeScript, Namespaces (deprecated but still useful for legacy code), Using external TypeScript modules.

## **Unit 13: Asynchronous Programming –**

Working with Promise types in TypeScript, async and await syntax and types, Handling errors in asynchronous code with try/catch, Using Promise.all, Promise.race with TypeScript types, Defining custom async types and interfaces.

## **Unit 14: TypeScript with Web Frameworks –**

Using TypeScript in frameworks like React, Angular, and Node.js, Setting up TypeScript with React (JSX and TypeScript), TypeScript and Angular (using decorators, services, etc.), TypeScript in Express applications (for server-side development).

## **Unit 15: Working with TypeScript -**

Configuration, Understanding and configuring tsconfig.json, Compiler options (target, module, strict mode, etc.), TypeScript compiler flags for debugging, Using tsc with build tools (Webpack, Gulp, or Grunt).

## **Unit 16: Error Handling and Debugging –**

Common TypeScript compiler errors and how to resolve them, TypeScript's strict mode (strictNullChecks, noImplicitAny, etc.), Using tslint or eslint for code linting and enforcing coding standards, Debugging TypeScript code in editors (VS Code, WebStorm).

## **Unit 17: Testing and TypeScript –**

Setting up testing environments with TypeScript (Jest, Mocha, etc.), Writing unit tests for TypeScript code, TypeScript's support for mocking and stubbing in tests, Using type definitions for testing libraries.

## **Module 17 : Sixth Live Project With TypeScript**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS and TypeScript. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 18 : DSA With JavaScript**

### **Unit 1: What is Data Structure? -**

Learn about Complexities, Learn Data Structures, Built-in Data Structures in JavaScript

### **Unit 2: Recursion in JavaScript**

### **Unit 3: Arrays -**

What is Array, Declaration of Array, Strlen(), Strcmp(), Strcat(), Initialization of Array, Use of array in function, One Dimensional Array, 2-D Array, Multidimensional Array

### **Unit 4: Build In Data Structure -**

Map in JavaScript, Set in JavaScript, Array in javascript, String in javascript, JavaScript Objects

### **Unit 5: Learn Data Structures with JavaScript**

### **Unit 6: Infix & Postfix expression –**

Infix to Postfix, Prefix to Infix, Prefix to Postfix, Postfix to Prefix, Postfix to Infix, Infix To Prefix

### **Unit 7: Linked List in JavaScript –**

What is List, Traversing an Array List, Searching in an Array List, Insertion into an array list, Deletion from an Array List, Linked List, Traversing an Linked List, Searching in an Linked List, Insertion into an Linked list, Deletion from an Linked List, Reverse Linked List, Circular Linked List, Creation an Circular Linked List, Traversing an Circular Linked List, Insertion into an Circular Linked list, Deletion from an Circular Linked List, Sorted Linked List, Double Linked List, Traversing an Doubly Linked List, Insertion into an Doubly Linked list, Deletion from an Doubly Linked List

### **Unit 8: Stack in JavaScript –**

Stack, Array implementation of Stack, Push Operation on Stack, POP operation on Stack, Linked List implementation, isEmpty, isFull, top/Peek

## **Unit 9: Queue in JavaScript –**

Queue, Array implementation of Queue, Add operation in Queue, Delete operation in Queue, Circular Queue, Add Operation in Circular Queue, Delete Operation in Circular Queue, Priority Queue, Linked List implementation of priority queue, Dequeue, Array implementation of Dequeue, Add and delete operation in dequeue

## **Unit 10: Sorting and Searching -**

What is Sorting, Efficiency Parameters, Efficiency of Sorting, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Merging, Merge Sort, Radix Sort, Quick Sort, Binary Tree Sort, Heap Sort, Binary Search

## **Unit 11: Hashing and Storage Management -**

Hashing, Hash Function for floating point numbers, Hash function for string, Collision Resolution(Open Hashing), Closed Hashing(Open Addressing), Rehashing, Extendible Hashing, Storage management, Garbage Collection, Dynamic Memory Management

## **Unit 12: Tree in JavaScript –**

Binary Tree, Binary Search Tree, AVL Tree

## **Unit 13: Graph in JavaScript –**

Graph Representation, Adjacency Matrix, Adjacency List

## **Module 19 : Seventh Live Project With DSA JavaScript**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS, DSA and JavaScript. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 20 : React JS**

### **Unit 1: Introduction to React -**

What is React?, Why React?, React version history, React 16 vs React 15, Just React – Hello World, Using create-react-app, Anatomy of react project, Running the app, Debugging first react app

### **Unit 2 : Templating using JSX -**

Working with React. createElement, Expressions, Using logical operators, Specifying attributes, Specifying children, Fragments

### **Unit 3 : About Components -**

Significance of component architecture, Types of components, Functional, Class based, Pure, Component Composition

### **Unit 4: Working with state and props -**

What is state and its significance, Read state and set state, Passing data to component using props, Validating props using propTypes, Supplying default values to props using defaultProps

## **Unit 5: Rendering lists -**

Using react key prop, Using map function to iterate on arrays to generate elements

## **Unit 6: Event handling in React -**

Understanding React event system, Understanding Synthetic event, Passing arguments to event handlers

## **Unit 7: Understanding component lifecycle and handling errors -**

Understand the lifecycle methods, Handle errors using error boundaries

## **Unit 8: Working with forms -**

Controlled components, Uncontrolled components, Understand the significance to default Value prop, Using react ref prop to get access to DOM element

## **Unit 9: Context -**

What is context, When to use context, Create Context, Context.Provider, Context.Consumer, Reading context in class

## **Unit 10: Code-Splitting -**

What is code splitting, Why do you need code splitting, React.lazy, Suspense, Route-based code splitting

## **Unit 11: Hooks -**

What are hooks, Why do you need hooks, Different types of hooks, Using state and effect hooks, Rules of hooks

## **Unit 12: Routing with react router -**

Setting up react router, Understand routing in single page applications, Working with BrowserRouter and HashRouter components, Configuring route with Route component, Using Switch component to define routing rules, Making routes dynamic using route params, Working with nested routes, Navigating to pages using Link and NavLink component, Redirect routes using Redirect Component, Using Prompt component to get consent of user for navigation, Path less Route to handle failed matches

## **Unit 13: Just Redux -**

What is redux, Why redux, Redux principles, Install and setup redux, Creating actions, reducer and store

## **Unit 14: Immutable.js -**

What is Immutable.js?, Immutable collections, Lists, Maps, Sets

## **Unit 15: React Redux -**

What is React Redux, Why React Redux, Install and setup, Presentational vs Container components, Understand high order component, Understanding mapStateToProps and mapDispatchToProps usage

## **Unit 16: Redux middleware -**

Why redux middleware, Available redux middleware choices, What is redux saga, Install and setup redux saga, Working with Saga helpers, Sagas vs promises



## **Unit 17: Unit Testing -**

Understand the significance of unit testing, Understand unit testing jargon and tools, Unit testing react components with Jest, Unit testing react components with enzyme

## **Unit 18: Webpack Primer -**

What is webpack, Why webpack, Install and setup webpack, Working with webpack configuration file, Working with loaders, Working with plugins, Setting up Hot Module Replacement

## **Unit 19: Isomorphic React -**

What is server-side rendering (SSR)?, Why SSR, Working with render To String and render To StaticMarkup methods

## **Module 21 : Eighth Live Project With ReactJS**

Extrabyte Academy is offering professional live project experience to the 2 year students with ReactJS. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 22 : NextJS**

### **Unit 1: Introduction to Next.js -**

What is Next.js? Overview of its features (SSR, SSG, API routes), Why use Next.js over regular React apps?, Setting up the Next.js development environment (Installation, project setup), Structure of a Next.js project (pages, components, public, and styles directories), Running the development server (npm run dev), Understanding the next.config.js file.

### **Unit 2: Pages and Routing -**

Introduction to Pages in Next.js, File-based routing in Next.js, Dynamic routing with file naming conventions (e.g., [id].js), Nested routes and linking between pages using the Link component, Catch-all routes for parameterized paths, Redirects and rewrites in Next.js (using next.config.js).

### **Unit 3: Static Site Generation (SSG) -**

What is Static Site Generation (SSG)?, Understanding getStaticProps() (fetching data at build time), Using getStaticPaths() for dynamic routes with SSG, Benefits and use cases for Static Site Generation, Static rendering vs. client-side rendering, Next.js with Markdown or JSON files (e.g., building a blog with SSG).

### **Unit 4: Server-Side Rendering (SSR) -**

What is Server-Side Rendering (SSR)?, Understanding getServerSideProps() (fetching data on each request), When to use SSR over SSG, Benefits of SSR in SEO and performance, Caching server-side rendered pages, Handling loading states and errors in SSR.

### **Unit 5: API Routes -**

Introduction to API Routes in Next.js, Creating API endpoints with file-based routing (/pages/api), Handling HTTP methods (GET, POST, PUT, DELETE) in API routes, Parsing request bodies and handling query parameters, Using API routes for server-side logic (e.g., authentication, form submissions), Secure API routes with authentication (JWT, sessions), Using external APIs from API routes.

## **Unit 6: Styling in Next.js -**

CSS Modules in Next.js (scoped styles for components), Global styles with globals.css, Adding third-party CSS frameworks (Bootstrap, Tailwind CSS, etc.), Using styled-components or Emotion for CSS-in-JS, Implementing CSS preprocessors like SASS with Next.js, Adding and using custom fonts, Next.js Image Optimization (using the next/image component).

## **Unit 7: Static Files and Assets -**

Understanding the public directory for static files (images, fonts, etc.), Optimizing images in Next.js (using next/image for automatic optimization), Hosting files in the public directory and accessing them in your app, Managing static assets and best practices for performance.

## **Unit 8: Client-Side Routing and Navigation -**

How client-side routing works in Next.js, Using the Link component for client-side navigation, useRouter() hook for programmatic navigation, Query parameters and shallow routing, Pre-fetching links for performance improvements.

## **Unit 9: Authentication and Authorization -**

Introduction to Authentication in Next.js, Using Next.js API routes for authentication (JWT, OAuth, session-based auth), Implementing authentication with third-party providers (e.g., NextAuth.js), Protecting routes with authentication middleware, Handling user sessions and cookies, Redirecting users based on authentication state.

## **Unit 10: State Management -**

Managing state in Next.js using React Context API, Using Redux or Recoil for complex state management, Client-side state vs server-side state, Storing and persisting state with localStorage or cookies, Server-side state with getServerSideProps() or API routes.

## **Unit 11: Performance Optimization -**

Code Splitting and Lazy Loading in Next.js, Image Optimization with the next/image component, Automatic Static Optimization, Server-Side Caching and Static File Caching, Analyzing performance with next-analytics, Best practices for reducing render blocking and improving performance.

## **Unit 12: Internationalization (i18n) -**

Introduction to Internationalization (i18n) in Next.js, Using Next.js's built-in i18n support, Managing translations with JSON files, Automatic language detection and routing based on user preferences, Optimizing static pages for different languages

## **Unit 13: Deploying Next.js Applications -**

Deployment strategies for Next.js, Deploying to Vercel (native platform for Next.js), Deploying to other platforms (Netlify, AWS, etc.), Optimizing the build for production (next build and next export), Handling environment variables in production (.env files), Monitoring and logging in production.

## **Unit 14: Advanced Features in Next.js -**

Incremental Static Regeneration (ISR) for dynamic content, Preview mode for previewing unpublished content, Web Vitals and performance metrics, Custom Document (\_document.js) and Custom App (\_app.js), Error boundaries and custom error pages (\_error.js), Custom Server setup in Next.js, Integrating Next.js with CMS (Content Management Systems) like WordPress or Sanity.

## **Unit 15: Testing in Next.js -**

Setting up testing with Jest and React Testing Library, Writing unit tests for components and pages, Testing API routes and server-side logic, End-to-end testing with Cypress or Playwright, Mocking data and API calls in tests.

## **Module 23 : Ninth Live Project With NextJS**

Extrabyte Academy is offering professional live project experience to the 2 year students with NextJS. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 24 : Adobe Photoshop**

### **Unit 1: Getting Acquainted with Photoshop -**

The Photoshop Environment, Palettes and the Palette Well, Creating Custom Workspaces, Opening Images, Using the File Browser, Image Magnification, Viewing Document Information, Moving the Image, Undoing Mistakes and The History Palette, Displaying Drawing Guides, Making Measurements, Adding Annotations, Setting Preferences

### **Unit 2: Basic Image Manipulation -**

Bitmap Images, Vector Images, Image Size and Resolution Settings , Scanning Images, Creating New Images, Placing Files

### **Unit 3: Color Basics -**

Color Modes and Models, Color Mode Conversion, Previewing Color Differences Between Operating Systems, Color Management, Foreground and Background Colors, Using the Color Picker, Selecting Colors with the Color Palette, Selecting Colors with the Eyedropper Tool, Selecting Colors with the Swatches Palette

### **Unit 4: Painting Tools -**

Painting Tools, The Brush Tool, Blending Modes, The Pencil Tool, The Eraser Tool, The Magic Eraser Tool, The Background Eraser Tool, Using the Art History Brush, Using the History Brush

### **Unit 5: Brush Settings -**

Using the Brushes Palette, Creating Custom Brush Tips by Selection, Creating Custom Brush Tips in the Brushes Palette, Setting Shape Dynamics, Setting Brush Scattering, Setting Brush Texture, Setting Dual Brushes, Setting Color Dynamics , Setting Other Dynamics, Miscellaneous Brush Settings, Clearing Brush Settings, Saving a Customized Brush, Saving a Customized Brush Library

### **Unit 6: Making Selections -**

Selection Basics, Making Pixel Selections, The Marquee Tools, The Lasso Tools, The Magic Wand Tool, Selecting by Color Range , Adjusting Pixel Selections, The Extract Command, Copying and Pasting Pixel Selections, Saving and Loading Selections

### **Unit 7: Filling and Stroking -**

Applying Fills, Using the Paint Bucket Tool , Using the Gradient Tool , Using the Gradient Editor , Using Patterns, Using the Pattern Maker , Stroking

## **Unit 8: Layers -**

Using Layers and Layer Sets, Creating Layers and Layer Sets, Stacking and Linking Layers, Moving Layer Content with the Move Tool, Locking Layers, Common Layer Management Tasks, Merging and Flattening Layers

## **Unit 9: Advanced Layers -**

Layer Styles, Adjustment Layers and Fill Layers, Masking Layers, Creating Clipping Groups, Creating Knockouts

## **Unit 10: Text -**

Text Basics, Entering Text, Selecting Text, Editing the Bounding Box, Creating a Type Selection, Applying Effects to Type Layers, Using the Character Palette, Checking for Spelling Errors, Using the Paragraph Palette

## **Unit 11: Drawing -**

Raster vs. Vector, Shape Layers and Shape Options, Using the Shape Tools, Using the Pen Tools, Using the Anchor Point Tools , Using the Paths Palette, Working with Paths

## **Unit 12: Using Channels and Masking -**

Using the Channels Palette, Using Channels, Spot Colors, Blending Channels and Layers , Masks, Using Alpha Channels

## **Unit 13: Manipulating Images -**

Changing the Canvas Size, Rotating and Flipping Images, Cropping Images, The Free Transform Command, The Smudge Tool, Blurring and Sharpening Images, Using the Dodge Tool and the Burn Tool, The Sponge Tool, Filters and The Filter Gallery, The Liquify Command, The Clone Stamp Tool, The Pattern Stamp Tool, The Healing Brush Tool, The Patch Tool, The Color Replacement Tool

## **Unit 14: Getting to Know the Work Area -**

Starting to work in Adobe Photoshop, Using the tools, Using the tool options bar and other palettes, Undoing actions in Photoshop, Customizing the workspace, Jumping to Adobe ImageReady, Using Photoshop Help, Using Adobe online Services, Toolbox overview

## **Unit 15: Using Adobe Bridge -**

Viewing and editing files in Adobe Bridge, Embedding information for easy identification, Searching with Adobe bridge, Using favorites and Collections, Automating routine tasks, Acquiring stock photography

## **Unit 16: Basic Photo Corrections -**

Strategy of retouching, Resolution and image size, Getting started, Straightening and cropping an image , Making automatic adjustments, Manually adjusting the tonal range, Removing a colorcast, Replacing colors in an image, Adjusting lightness with the Dodge tool, Applying the Unsharp Mask filter, Comparing automatic and manual results, Saving the image for four-color printing

## **Unit 17: Retouching and Repairing -**

Repairing areas with the Clone Stamp tool, Using the Spot Healing Brush tool, Using the Healing Brush and Patch tools, Retouching on a separate layer

## **Unit 18: Working with Selections -**

About selecting and selection tools, Selecting with the Magic Wand tool, Using the Magic Wand with other selection tools , Working with oval and circular selections, Selecting with the lasso tools, Rotating a selection, Selecting with the Magnetic Lasso tool, Cropping an image and erasing within a selection

## **Unit 19: Layer Basics -**

About Layers, Using the Layers Palette, Rearranging layers, Applying a layer style, Flattening and saving files, Photoshop for kids

## **Unit 20: Masks and Channels -**

Working with masks and channels, Creating a quick mask, Editing a quick mask, Saving a selection as a mask, Editing a mask, Loading a mask as a selection and applying an adjustment, Extracting an image, Applying a filter effect to a masked selection, Creating a gradient mask, Applying effects using a gradient mask

## **Unit 21: Correcting and Enhancing Digital Photographs -**

About Camera raw, Processing camera raw, Processing camera files, Correcting digital photographs, Editing images with a vanishing-point perspective , Correcting image distortion, Creating a PDF portfolio\

## **Unit 22: Typographic Design -**

Creating a clipping mask from type, Creating a design element from type, Using interactive formatting controls, Warping point type, Designing a paragraph of type, Warping a layer, Review questions and answers, Dancing with Type

## **Unit 23: Vector Drawing Techniques -**

About bitmap images and vector graphics, About paths and the pen tool, Using paths with artwork, Creating scalable objects for the background, Working with defined custom shapes, Importing a Smart Object

## **Unit 24: Advanced Layer Techniques -**

Creating paths to clip a layer, Creating layer sets, Creating an adjustment layer, Creating a knockout gradient layer , Importing a layer from another file, Applying layer styles, Duplicating and clipping a layer, Liquifying a layer, Creating a border layer, Flattening a layered image IIN

## **Unit 25: Vector Compositing -**

Automating a multistep mask, Setting up a four-image montage, Hand-coloring selections on a layer, Changing the color balance, Applying filters

## **Unit 26: Creating Links Within an Image -**

Slicing and image in Photoshop, Jumping to ImageReady, Creating image maps in ImageReady, Saving linked images in an HTML file

## **Unit 27: Creating Rollover Web Visuals -**

About rollovers, Creating rollover states, Creating remote rollovers, Saving the page as HTML

## **Module 25 : Adobe Illustrator And Logo Designing**

### **Unit 1: INTRODUCING ILLUSTRATOR CC -**

Apply Design Principles, Elements, and Graphics Composition, Evaluate Graphics Scalability, Navigate the User Interface, Customize the User Interface

### **Unit 2: CREATING DOCUMENTS CONTAINING BASIC SHAPES -**

Create Documents, Save Documents, Draw Basic Shapes

### **Unit 3: CREATING DOCUMENTS CONTAINING CUSTOMIZED PATHS -**

Draw Paths, Modify Paths

### **Unit 4: CREATING GRAPHICS CONTAINING CUSTOMIZED TEXT -**

Insert Text, Apply Flow Text Along a Path, Insert Bounded and Threaded Text, Apply Text Styles, Insert Typographical Characters

### **Unit 5: CUSTOMIZING OBJECTS -**

Import, Place, and Link Graphics, Alter the Appearance of Objects

### **Unit 6: CUSTOMIZING BASIC SHAPES -**

Apply Strokes and Brushes, Fill Shapes, Apply Graphics Styles, Distort Text with Text Envelopes

### **Unit 7: PREPARING DOCUMENTS FOR DEPLOYMENT -**

Update Text, Wrap Text, Hyphenate Text, Optimize Content for Print, Optimize Content for the Web, Optimize Content for PDF Documents

## **Module 26 : PHP**

### **Unit 1: Introduction to PHP -**

Evaluation of Php, Basic Syntax, Defining variable and constant, Php Data type, Operator and Expression.

### **Unit 2: Decisions and Loop -**

Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with Html

### **Unit 3: Function -**

What is a function, Define a function, Call by value and Call by reference, Recursive function, String Creating and accessing, String Searching & Replacing String, Formatting String, String Related Library function

### **Unit 4: Array -**

Anatomy of an Array, Creating index based and Associative array Accessing array, Element Looping with Index based array, Looping with associative array using each () and foreach(), Some useful Library function

### **Unit 5: Handling Html Form with PHP -**

Capturing Form, Data Dealing with Multi-value filed, and Generating File uploaded form, redirecting a form after submission

## **Unit 6: Working with file and Directories -**

Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading

## **Unit 7: Session and Cookie -**

Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session

## **Unit 8: Database Connectivity with MySql -**

Introduction to RDBMS, Connection with MySql Database, Performing basic database operation(DML) (Insert, Delete, Update, Select), Setting query parameter, Executing queryJoin (Cross joins, Inner joins, Outer Joins, Self joins.)

## **Unit 9: Exception Handling -**

Understanding Exception and error, Try, catch, throw. Error tracking and debugging

# **Module 27 : Database With MySQL/MongoDB**

## **Unit 1: MySQL Tutorial -**

MySQL Tutorial, MySQL History, MySQL Features, MySQL Data Types, Install MySQL

## **Unit 2: MySQL Database -**

Create Database, Select Database, Drop Database

## **Unit 3: Table & Views -**

CREATE Table, ALTER Table, TRUNCATE Table, DROP Table, MySQL Views

## **Unit 4: MySQL Queries -**

MySQL Queries, INSERT Record, UPDATE Record, DELETE Record, SELECT Record

## **Unit 5: MySQL Clauses -**

MySQL WHERE, MySQL DISTINCT, MySQL FROM, MySQL ORDER BY, MySQL GROUP BY, MySQL HAVING

## **Unit 6: MySQL Conditions -**

MySQL AND, MySQL OR, MySQL AND OR, MySQL LIKE, MySQL IN, MySQL NOT, MySQL IS NULL, MySQL IS NOT NULL, MySQL BETWEEN

## **Unit 7: MySQL Join -**

MySQL JOIN

## **Unit 8: Aggregate Functions -**

MySQL count(), MySQL sum(), MySQL avg(), MySQL min(), MySQL max(), MySQL first(), MySQL last()

## **Unit 9: Differences -**

MariaDB vs MySQL, PostgreSQL vs MySQL

## **Unit 10: MySQL Date/Time Functions -**

MySQL Date/Time, DATE() function, ADDDATE() function, CURDATE() function, CURRENT\_DATE() function, DATE\_ADD() function, DATE\_FORMAT() function, DATEDIFF() function, DAY() function

## **Unit 11: MSQL String Functions -**

MySQL String, String CONCAT\_WS() function, String CONCAT() function, String CHARACTER\_LENGTH() function, String ELT() function, String EXPORT\_SET() function, String FIELD() function

## **Module 28 : Tenth Live Project With PHP & Database**

Extrabyte Academy is offering professional live project experience to the 2 year students with PHP and Database. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 29 : Laravel Framework**

### **Unit 1: Laravel – Overview and Installation -**

Advantages of Laravel, Composer, Artisan, Features of Laravel

### **Unit 2: Application Structure -**

App, Console, Events, Exceptions, Http, Jobs, Listeners, Database, Public, Resources, Storage, Tests, Vendor

### **Unit 3: Configuration -**

Environment Configuration, Accessing Configuration Values, Caching of Configuration, Maintenance Mode

### **Unit 4: Routing -**

Basic Routing, Route Parameters, Named Routes

### **Unit 5: Middleware -**

Registering Middleware, Global Middleware, Route Middleware, Middleware Parameters, Terminable Middleware

### **Unit 6: Namespaces -**

Declaration of namespace

### **Unit 7: Controllers -**

Creating a Controller, Controller Middleware, Restful Resource Controllers, Implicit Controllers, Constructor Injection, Method Injection

### **Unit 8: Request, Cookie, Response -**

Retrieving the Request URI, Retrieving Input, Creating a Cookie, Retrieving a Cookie, Basic Response, Attaching Headers, Attaching Cookies, JSON Response

### **Unit 9: Views, Blade Templates -**

Passing Data to Views, Sharing Data with all Views, Steps for Creating a Blade Template Layout



## **Unit 10: Redirections, Errors and Logging -**

Redirecting to Named Routes, Redirecting to Controller Actions, Errors, Logging, Working With Database, Forms, Localization

## **Unit 11: Session, Validation -**

Accessing Session Data, Storing Session Data, Available Validation Rules in Laravel

## **Unit 12: File Uploading, Sending Email, Ajax, json() function**

## **Unit 13: Error Handling -**

Important Points, Error Log, Severity Levels, Event Handling

## **Unit 14: Facades, Authentication, Authorization -**

Facade Class Reference, Contracts, CSRF Protection, Command, Controller, Manually Authenticating Users, Difference between Authentication and Authorization, Authorization Mechanism in Laravel

## **Unit 15: Artisan Console, Encryption, Hashing -**

Introduction to Artisan, Writing Commands, Configuration, Encryption Process, Decryption Process, Basic Usage, Verification of Password against Hash

## **Module 30 : Eleventh Live Project With Laravel**

Extrabyte Academy is offering professional live project experience to the 2 year students with Laravel. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 31 : NodeJS Design Patterns**

### **Unit 1: Introduction to Node JS -**

Introduction, What is Node JS?, Advantages of Node JS, Traditional Web Server Model, Node.js Process Model

### **Unit 2: Setup Dev Environment -**

Install Node.js on Windows, Installing in mac os, Working in REPL, Node JS Console

### **Unit 3: Node JS Modules -**

Functions, Buffer, Module, Module Types, Core Modules, Local Modules, Module.Exports

### **Unit 4: Node Package Manager -**

What is NPM, Installing Packages Locally, Adding dependency in package.json, Installing packages globally, Updating packages

### **Unit 5: Creating Web Server -**

Creating web server, Handling http requests, Sending requests

### **Unit 6: File System -**

Fs.readFile, Writing a File, Writing a file asynchronously, Opening a file, Deleting a file, Other IO Operations

## **Unit 7: Debugging Node JS Application -**

Core Node JS debugger, Debugging with Visual Studio

## **Unit 8: Events -**

EventEmitter class, Returning event emitter, Inhering events

## **Unit 9: Express.JS -**

Configuring routes, Working with express

## **Unit 10: Serving Static Resources -**

Serving static files, Working with middle ware

## **Unit 11: Database connectivity -**

Connection string, Configuring, Working with select command, Updating records, Deleting records

## **Unit 12: Template Engines -**

Why Template Engine, What is Jade, What is vash

## **Module 32 : Twelfth Live Project With NodeJS**

Extrabyte Academy is offering professional live project experience to the 2 year students with NodeJS. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 33 : Python Core**

### **Unit 1: Introduction to Python Language -**

What is Python, Uses of Python Programming Language / Python Applications, Python for Software development, Python for Networking, Python for Automated Testing, Features of Python Programming Language, Implementations of Python, and Python career opportunities.

### **Unit 2: Python Language Syntax -**

Modes of Programming in Python, Interactive mode programming, Script mode programming, Creating Python program file, Python Identifiers, Python keywords, Lines and Indentation, Spilt Python statements, Join Python Statements, Writing code blocks, Comments in Python, and Quotation in Python.

### **Unit 3: Python Keywords and Identifiers -**

Python keywords or Reserved words, Python keywords define the syntax and structure of the Python language, Python keywords are case sensitive, Python literals (True, False, Null), Python Identifiers, class names, variable names, function names, method names, and Identifier naming rules.

### **Unit 4: Python Comments -**

Purpose/use of comments in Computer Programming, Comments for Understanding Python code, Python Comment Syntax, Python Single line comment, Multiline comment in Python, and writing Python comments.

## **Unit 5: Python Variables:**

What is Variable?, Declaration of Variables, Assign Values to Variables, Initialization, Reading, Variable naming restrictions, and Types of Python Variables.

## **Unit 6: Python Data Types -**

What is Data Type?, Implicit Declaration of Data Types, Python Numbers (Integers, floating-point numbers, and complex numbers), Python Strings, Python boolean data type.

## **Unit 7: Python Operators -**

Python Arithmetic, Comparison/Relational Operators, Increment Operators, Logical operators, Python Identity Operators, and Python Operators Precedence

## **Unit 8: Python Control Flow – Decision Making -**

(Decision Making / Conditional Statements in Python, Simple If Structure, if-else structure, if elif structure, and nested If Structure. Execute a block of Statements when the condition is true, Execute a block of Statements when a compound condition is true, Execute a block of Statements when the condition is true otherwise execute another block of Statements, Decide among several alternates(elif), and Execute a block of Statements when more than one condition is true (Nested if))

## **Unit 9: Python Control Flow – Looping -**

Python Control Flow Statements, Python Loop Statements. Python while loop, Python for loop, Python range(), Python Nested Loop Structures, and Inserting conditions in Loops and vice versa.

## **Unit 10: Python Control Flow – Branching -**

(Python Flow Control – Branching Statements, A branching statement is a statement that determines whether other statements will be executed. Python Branching Statements – break, continue, pass)

## **Unit 11: Python Numbers -**

Python Number data types are for storing numeric values, Python supports integers, floats, and complex numbers.

## **Unit 12: Python Strings -**

String is a sequence of characters written in single quotes or in double quotes or in three double quotes. The string may have Alphabets, Numbers, and Special Characters. Operations on Strings, Finding String length, Concatenating Strings, Print a String multiple times

## **Unit 13: Python Lists -**

Python Data Structures, Create Python Lists, Update Python Lists, Delete Elements from Python Lists, and Built-in Functions & Built-in Methods for Python Lists.

## **Unit 14: Python Tuples -**

Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets

## **Unit 15: Python Sets -**

A Python set is a collection that is unordered and unindexed. In Python sets are written with curly brackets.

## **Unit 16: Python Dictionaries -**

A dictionary is a collection that is unordered, changeable, and indexed. In Python dictionaries are written with curly brackets, and they have keys and values

## **Unit 17: Python Arrays -**

Array is a container that can hold a fixed number of items and these items should be of the same type. Python does not have built-in support for Arrays, but Python Lists can be used instead

## **Unit 18: Python user-defined Functions -**

In all programming and scripting languages, a function is a block of program statements that can be used repetitively in a program. It saves the time of a developer. In Python concept of function is the same as in other languages. There are some built-in functions that are part of Python. Besides that, we can define functions according to our needs

## **Unit 19: Python Built-in Functions -**

Python has several functions that are readily available for use. These functions are called built-in functions

## **Unit 20: Python – Modules -**

A module allows you to logically organize your Python code. Simply, a module is a file consisting of Python code. A module can define functions, classes, and variables. A module can also include runnable code

## **Unit 21: Python User Input -**

Python user input from the keyboard can be read using the input() built-in function. The input from the user is read as a string and can be assigned to a variable

## **Unit 22: Python File Handling -**

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files

## **Unit 23: Python Exceptions Handling -**

Python provides us with a way to handle the Exception so that the other part of the code can be executed without any disruption. However, if we do not handle the exception, the interpreter doesn't execute all the code that exists after that

## **Unit 24: Regular Expressions -**

A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern. Python has a built-in package called re, which can be used to work with Regular Expressions. The re module offers a set of functions that allows us to search a string for a match. RegEx Functions, Metacharacters, and Special Sequences.

## **Unit 25: Class & Object**

Basic concepts of object programming, A short journey from procedural to object approach, Properties, Methods

## **Module 34 : Python Advanced**

### **Unit 1: OOPS -**

Class and object, Attributes, Methods, Overloading, Overriding, Data hiding

### **Unit 2: Inheritance -**

Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hybrid Inheritance, Hierarchical Inheritance, IS-A Relationship and HAS-A Relationship

### **Unit 3: Polymorphism -**

Duck Type Philosophy, Method Overloading, Operator Overloading, Constructor Overloading, Method Overriding, Constructor Overriding

### **Unit 4: Database -**

Introduction, Connections, Executing queries, Transactions, Handling error

### **Unit 5: Networking -**

Socket, Socket Module, Methods, Client and server, Internet modules

### **Unit 6: Multithreading -**

Introduction Multitasking, isAlive, Enumerate, Thread Class, Demonstrate, Running Parallel Program, Operating Based Multithreading

### **Unit 7: GUI Programming -**

Introduction, Components and events, An example GUI, The Root Component, Adding a button, Entry Widgets, Text Widgets, Check Button

### **Unit 8: Data Science With Python -**

Introduction, Setting up with environment, Basic data types, Numpy arrays, Exploring Pandas, Pandas Series and Data Frames, Statistics with Pandas, DataFrame

## **Module 35 : Thirteenth Live Project With Python**

Extrabyte Academy is offering professional live project experience to the 2 year students with Python. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 36 : Flask**

### **Unit 1: Introduction to Flask -**

What is Flask? Overview and history of Flask, Installing Flask and setting up the development environment, Flask directory structure (app structure, templates, static files), Running the Flask development server, Introduction to routing in Flask (URL routing and mapping functions to URLs), Introduction to Flask templates (Jinja2 templating engine), Basic Flask application structure (creating a simple Flask app).

## **Unit 2: Flask Routing and Views –**

Defining routes in Flask using the `@app.route()` decorator, HTTP Methods (GET, POST, PUT, DELETE), Dynamic URL routing with parameters in Flask routes (e.g., `/user/<username>`), Handling different HTTP methods in a single route, Redirects and error handling (404 pages, custom error pages), Rendering HTML templates using `render_template()`, Passing variables from views to templates, URL building with `url_for()`.

## **Unit 3: Flask Templates (Jinja2) -**

Understanding Jinja2 templating in Flask, Template inheritance and base templates, Using conditionals and loops in templates, Template filters and custom filters, Template macros (reusable template components), Escaping content and preventing XSS attacks in templates, Partials and including other templates in a main template.

## **Unit 4: Handling Forms and User Input -**

Creating and processing forms in Flask, Handling form data using `request.form` and `request.args`, Managing GET and POST requests, Form validation and handling errors, CSRF protection in Flask forms, Using Flask-WTF (Flask extension for handling forms), Handling file uploads in Flask, Flash messages and feedback to users.

## **Unit 5: Flask Redirects and Error Handling -**

Working with HTTP redirects (using `redirect()` and `url_for()`), Handling HTTP errors (404, 500, etc.) with custom error pages, Flask's `abort()` function and custom error handling, Using `try/except` blocks for error handling in routes, Global error handling with `@app.errorhandler()`, Logging and debugging in Flask applications.

## **Unit 6: Flask Session and Cookies -**

Introduction to sessions in Flask (server-side session storage), Setting and getting session variables, Flask's secure cookie system (storing data in cookies), Expiring session data and using `session.permanent`, Using cookies for maintaining user preferences or authentication, Flask's session vs. `request.cookies`.

## **Unit 7: Flask Database Integration -**

Introduction to databases in Flask (SQL vs NoSQL), Setting up SQLite with Flask (using `flask_sqlalchemy`), Introduction to Flask-SQLAlchemy (ORM for Flask), Defining database models (tables, columns, relationships), CRUD operations (Create, Read, Update, Delete) with Flask-SQLAlchemy, Querying databases in Flask using SQLAlchemy methods, Migrations with Flask-Migrate, Connecting Flask with other databases (PostgreSQL, MySQL).

## **Unit 8: Authentication and Authorization -**

Introduction to authentication in Flask (user login/logout system), Flask-Login: Managing user sessions and authentication, Creating user registration and login forms, Handling password encryption with Werkzeug or Flask-Bcrypt, User authorization (role-based access control), Flask-Principal and Flask-User for user permissions, Protecting routes from unauthorized access, Managing user sessions and preventing session fixation attacks.

## **Unit 9: Flask Blueprints and Modular Applications -**

Introduction to Flask Blueprints (creating modular components), Organizing large Flask applications using Blueprints, Registering and including Blueprints in the main application, Organizing views, templates, and static files by Blueprint, Using Flask's Blueprint for reusable components (auth, admin, etc.).

## **Unit 10: Flask Testing and Debugging -**

Introduction to testing Flask applications, Writing unit tests for Flask routes and views (using unittest or pytest), Flask's test client and simulating requests, Testing database interactions with Flask-SQLAlchemy, Mocking and patching for testing external dependencies, Flask's debugger and logging for easier troubleshooting, Test coverage and continuous integration with Flask.

## **Module 37 : ORM(Object Relational Mapping)**

### **Unit 1: Introduction to ORM and SQLAlchemy -**

What is ORM? Benefits and drawbacks of using ORM over raw SQL, Overview of SQLAlchemy (ORM and Core), How SQLAlchemy integrates with Flask, Setting up a Flask project with SQLAlchemy, Installing and configuring Flask-SQLAlchemy, Basic configuration of a database URI (SQLite, PostgreSQL, MySQL), The relationship between Flask, SQLAlchemy, and the database.

### **Unit 2: Setting Up and Configuring SQLAlchemy in Flask -**

Installation of Flask-SQLAlchemy, Configuring Flask to use SQLAlchemy with the app's config dictionary, Defining the SQLALCHEMY\_DATABASE\_URI and SQLALCHEMY\_TRACK\_MODIFICATIONS, Creating the db object and initializing it with the Flask app, Creating and setting up the database, Basic application structure: Flask app, models, and routes.

### **Unit 3: Defining Models with SQLAlchemy -**

What are models in SQLAlchemy? Creating the first model, Defining tables using Python classes (model definitions), Columns and data types in SQLAlchemy (e.g., Integer, String, Date, Float, etc.), Using primary keys and foreign keys, Relationships: One-to-One, One-to-Many, Many-to-Many, Declaring relationships with relationship() and backref(), Working with nullable, default, and unique constraints in model fields, Using \_\_repr\_\_ method for better debugging and printability of model instances.

### **Unit 4: Performing CRUD Operations with SQLAlchemy -**

Basic CRUD operations in SQLAlchemy: Create: Inserting records using db.session.add() and db.session.commit(), Read: Querying data with query.filter\_by(), query.filter(), query.all(), query.first(), and query.get(), Update: Modifying records using session.merge() and committing with session.commit(), Delete: Removing records with session.delete() and committing the deletion, Querying with filters, sorting, and limiting results, Using SQLAlchemy's powerful filter(), like(), in\_(), between(), and order\_by() functions, Handling database sessions with db.session.

### **Unit 5: Relationships Between Models -**

Understanding and implementing One-to-Many relationships (e.g., Author to Books), Implementing Many-to-One relationships (reverse of One-to-Many), Implementing Many-to-Many relationships (e.g., Students and Courses), Defining relationships with relationship(), backref(), and foreign\_key(), Deleting and cascading changes across related models (e.g., delete all books when an author is deleted).

### **Unit 6: Handling Migrations with Flask-Migrate -**

Introduction to database migrations (why and when to use migrations), Installing and setting up Flask-Migrate, Understanding Alembic (used by Flask-Migrate for migrations), Initializing the migration environment (flask db init), Creating migrations (flask db migrate), Applying migrations (flask db upgrade), Downgrading migrations (flask db downgrade), Managing and resolving conflicts in migration files.

## **Unit 7: Querying with SQLAlchemy -**

Introduction to SQLAlchemy's ORM Query API, Using `.filter()` and `.filter_by()` for condition-based queries, Using `join()` to perform SQL JOIN operations (Inner Join, Left Join, etc.), Advanced querying with `subquery()`, `exists()`, and `union()`, Pagination of query results using `.paginate()`, Aggregating data with `count()`, `sum()`, `avg()`, `min()`, `max()`, Ordering query results with `.order_by()`, Limiting query results and skipping rows with `.limit()` and `.offset()`.

## **Unit 8: Transactions and Rollbacks -**

Introduction to database transactions, Manual transaction control using `db.session.begin()`, `db.session.commit()`, and `db.session.rollback()`, Using Flask-SQLAlchemy to manage the session and transactions, Handling transaction errors and exceptions in Flask applications, Committing multiple operations as a single transaction, Implementing atomic operations in Flask with SQLAlchemy.

## **Unit 9: Data Validation and Integrity -**

Using SQLAlchemy's built-in validation methods, Implementing custom validation logic in models, Handling database integrity errors (e.g., unique constraints, foreign key violations), Using Flask-WTF for form handling and validating model data, Implementing field constraints (e.g., `nullable=False`, `unique=True`), Handling validation errors and rolling back transactions.

## **Unit 10: Flask-SQLAlchemy Best Practices -**

Understanding the lifecycle of SQLAlchemy sessions in Flask, Optimizing database queries (e.g., selecting only required columns, lazy loading vs eager loading), Using indexes to improve query performance, Efficient use of database relationships (avoid N+1 queries), Working with database connection pooling, Keeping models decoupled from application logic, Organizing models, schemas, and views into different modules, Using Flask's `before_request` and `after_request` for session management.

## **Unit 11: Flask-SQLAlchemy with Flask-Admin -**

Introduction to Flask-Admin for building administrative interfaces, Integrating Flask-SQLAlchemy with Flask-Admin, Creating custom views for models using Flask-Admin, Customizing the Flask-Admin interface (form layouts, column filters, etc.), Adding security and access control to the admin interface.

# **Module 38 : Fourteenth Live Project With Flask**

Extrabyte Academy is offering professional live project experience to the 2 year students with Python with flask. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 39 : Github**

### **Unit 1: Introduction to Version Control and Git -**

What is version control and why it's important, Introduction to Git (distributed version control system), Installing Git and setting up a local repository, Basic Git concepts: commits, branches, and merging, The role of GitHub as a cloud-based platform for Git repositories, Setting up a GitHub account and configuring Git locally, Basic Git commands: `git init`, `git clone`, `git status`, `git add`, `git commit`, `git push`, `git pull`.



## **Unit 2: Basic GitHub Usage -**

GitHub overview: what it offers beyond version control, Creating a repository on GitHub, Understanding repositories, branches commits, and pull requests, Cloning a GitHub repository to your local machine, Pushing changes to GitHub from your local repository, Pulling updates from GitHub to your local machine, Creating, editing, and deleting files directly from GitHub's web interface, Forking a repository and cloning it to work on your own version, GitHub's file viewer and its features (blame, history, raw view, etc.).

## **Unit 3: Branching and Merging -**

Introduction to branching in Git and why it's useful, Creating and switching branches (git branch, git checkout, git switch), Working with multiple branches in GitHub repositories, Merging branches and resolving conflicts (git merge), Reverting and undoing commits (git revert, git reset), Rebase vs. merge: when to use each, Pull requests (PRs) as a mechanism for merging code in GitHub, Reviewing and approving pull requests in GitHub.

## **Unit 4: Collaborating with GitHub -**

Introduction to collaboration in GitHub (clone, commit, push, pull requests), Forking a repository and making changes in your own copy, Using pull requests to propose changes to a repository, Reviewing and merging pull requests on GitHub, Issues in GitHub: opening, commenting, and closing issues, Using labels, milestones, and assignees for organizing work, Writing effective commit messages, Using GitHub actions for automating workflows.

## **Unit 5: Managing GitHub Projects -**

GitHub Projects: what they are and how to use them, Organizing your repository using boards and columns, Adding tasks, cards, and tracking project progress, Using GitHub milestones for version and release planning, Creating and using project templates in GitHub, Working with issues and pull requests in the context of projects.

## **Unit 6: Advanced Git Features -**

Git Stashing (git stash) to temporarily save changes, Handling merge conflicts: identifying and resolving conflicts, Git hooks: automating Git commands with custom scripts, Git tags for marking important commits (versions/releases), Advanced Git commands: git cherry-pick, git rebase, git bisect, The git log command for viewing history and commit history graphs, Creating and using custom aliases for Git commands.

## **Unit 7: GitHub Pages and Static Websites -**

Introduction to GitHub Pages for hosting static websites, Creating and deploying GitHub Pages site from a repository, Understanding the structure of a GitHub Pages website, Configuring a custom domain for your GitHub Pages site, Using Jekyll for creating blogs or static websites on GitHub Pages.

## **Unit 8: Managing Access and Permissions on GitHub -**

GitHub teams and organizations: what they are and how they work, Managing repository collaborators and permissions, Setting up private vs. public repositories, Using access control with GitHub Actions for security and automation, Using SSH keys for secure GitHub authentication, Two-factor authentication (2FA) on GitHub for extra security.

## **Unit 9: Open Source Contributions and Best Practices -**

How to contribute to open-source projects on GitHub, Forking a project, creating branches, and submitting pull requests, Writing good pull request descriptions and providing helpful feedback, GitHub's contribution guidelines and best practices, Understanding the GitHub Flow for collaborative development, Following proper etiquette in open-source communities, How to maintain a project, accepting or rejecting pull

requests, and handling issues, Best practices for maintaining a clean Git history and avoiding “dirty” commits.

### **Unit 10: GitHub Security Features -**

Managing security alerts on GitHub repositories, Dependabot: automated dependency updates and vulnerability fixes, Scanning code for vulnerabilities with GitHub Advanced Security, Enforcing security policies for pull requests (e.g., code reviews, status checks), Enabling and using GitHub’s code scanning and secret scanning features.

## **Module 40 : Data Science**

### **Unit 1: Introduction to Data Science -**

What is Data Science?, Importance of Data Science in business and industry, Overview of the Data Science lifecycle: Problem Definition, Data Collection, Data Cleaning, Data Exploration, Modeling, Evaluation, and Deployment

### **Unit 2: Python for Data Science -**

Basics of Python: Variables, Data Types, Functions, Data Structures: Lists, Tuples, Dictionaries, Sets, Libraries for Data Science: NumPy for numerical computations, Pandas for data manipulation, Matplotlib/Seaborn for data visualization, SciPy for scientific computing

### **Unit 3: Data Wrangling & Preprocessing -**

Loading Data (CSV, Excel, databases), Cleaning Data (Handling missing data, duplicates, outliers), Data Transformation (Scaling, Normalization, Encoding Categorical Variables), Feature Engineering (Creating new features from existing data)

### **Unit 4: Exploratory Data Analysis (EDA) -**

Summary Statistics (Mean, Median, Mode, Variance, etc.), Visualizations: Histograms, Boxplots, Scatter Plots, Correlation Matrices Identifying patterns, trends, and outliers, Hypothesis Testing

### **Unit 5: Probability and Statistics for Data Science -**

Descriptive Statistics, Inferential Statistics, Probability Theory, Distributions (Normal, Binomial, Poisson, etc.), Sampling Methods and the Central Limit Theorem, Hypothesis Testing (p-value, t-tests, chi-square tests), Confidence Intervals

### **Unit 6: Machine Learning Foundations -**

Supervised Learning vs Unsupervised Learning, Introduction to ML Algorithms: Linear Regression, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees & Random Forest, Support Vector Machines (SVM), Naive Bayes, Model Evaluation: Accuracy, Precision, Recall, F1-Score, ROC-AUC

### **Unit 7: Model Evaluation and Validation -**

Overfitting and Underfitting, Cross-validation Techniques (K-fold, Stratified), Bias-Variance Tradeoff, Hyperparameter Tuning (Grid Search, Random Search)

## **Unit 8: Data Science Project -**

End-to-End Data Science Project: From data collection to model deployment, Working with real-world datasets, Building a Data Science Portfolio

## **Unit 9: Tools & Platforms -**

Jupyter Notebooks for interactive coding, Git & GitHub for version control and collaboration

## **Module 41 : Fifteenth Live Project With Data Science**

Extrabyte Academy is offering professional live project experience to the 2 year students with Data Science. The trainers enable to students to face the real challenges during the process of making web application.

## **Module 42 : Final Project**

Extrabyte Academy is offering professional live project experience to the 2 year students with HTML, CSS, JavaScript, React, PHP/NodeJS/Flask etc. The trainers enable to students to face the real challenges during the process of making web application.

## **Additional Module : Job Interview Preparation**

The **preparation for acing a tech interview** starts with a complete and worthwhile roadmap or preparation plan. It is quite obvious that until and unless you won't know what to prepare, where to prepare, what subjects hold more weightage.

Here, we're going to provide you with the required preparation plan along with the respective quality learning resources to make your interview preparation journey a bit more convenient and easier.

---

[extrabyteacademy@gmail.com](mailto:extrabyteacademy@gmail.com)

[www.extrabyteacademy.com](http://www.extrabyteacademy.com)

Contact : 9084516525

Gas Godam Road, S Bend, Haldwani