Full Stack Web Developer
Professional Program

|| About BIT!

A Perfect Platform for Career Transformation… For more than Two-decade, BIT has been a multi-disciplinary education and training Institute in Vadodara, Gujarat meant for Individuals, Professionals, and Corporate. To be at par with the current scenario of the industry, we aim to enhance and upgrade the skills and compatibility of an aspirant. BIT provides an Offline and Online Training platform presenting 500+ courses through Classroom-based and Virtual Class Room Training conducted by expert instructors. We have designed our curriculum considering the rapidly growing demands in the fields associated with Programming, Database, Networking, Data Science, Artificial Intelligence, Robotics, Ethical Hacking, Web Development, Graphic Designing, Software Development, Accounting, Engineering Designing Courses, and many more Computer Courses and Languages.

|| About Full Stack Web Developer

Full Stack Developers are developers that design complete apps and websites. These developers work on all facets of development, from frontend, to backend, to database and even debugging and testing. In short, the developer must understand the app through and through. Frontend developers are more sought after because of their expertise of not in one but multiple technologies. They can handle all aspects of development, and it can result in a more seamlessly created product.

However, resources to become a Full Stack Developer are limited and only offer sections or just random technologies that you must learn and combine on your own. So, we introduced this course that compiles all technologies in perfect sync to help developers transition from a simple developer to become a Full Stack Developer.

Starting from frontend development, the learner will slowly progress to become to other aspects of development including backend, database, debugging, version control and other essential technologies that are helpful for a developer.

Our Full Stack Course breaks down the fundamentals of each technology that you would require to become a Full Stack Developer. The entire course is broken down into five separate segments: Frontend, Backend, Database, Other Essential Technologies and Debugging/Version Control. Each section includes multiple technologies to help you gain more experience as a developer.

- Learn core concepts of both the front end and backend programming course
- Get familiar with the latest web developer technologies and ecosystems
- Learn all about sql and nosql databses
- Learn complete web development process
- Be able to use frontend web development skills to build website : HTML ,CSS,JAVASCRIPT
- Build a Todo List using frontend web development skills
- Build an interactive quiz app with frontend web development skills

Lecture	Content	Date	Remark
Lecture-1-5	• **Introduction to web development and web design** • Introduction to web development • Introduction to Web Design		
Lab-1-5	• **Practical**		
Lecture-6-18	• **HTML and HTML 5** • Introduction to HTML • What is an Element • HTML page structure • HTML metadata • HTML Doctype • Try it make index page • Element Formatting HTML • HTML Headings • Self-Closing Tags • HTML Comments • Preformatted Text • HTML Entities • HTML text formatting • More HTML text formatting • Text Formatting Sample Code • Practice text formatting • Divs and Spans to organize content • HTML5 Semantic Elements • Placeholder Content • Hyperlinks Anchor Tag • Try it Build a Page • HTML sample site code example • Adding Images to your Page • HTML lists • Sample Code HTML lists • HTML tables with Style • HTML tables • HTML Forms • Sample Tables • Introduction HTML5 • Create a Contact Form • HTML Form options • Form Submission • More input types HTML5 • Source Code sample input type • More with HTML5 Forms • Sample Code sign up form HTML • HTML form fieldset • HTML5 Graphics		

	• HTML5 Advance Features • Practical Exercise		
Lab-6-18	• **Practical**		
Lecture-19-30	• **CSS and CSS 3** ○ Introduction to CSS ○ CSS section resources and Source Code ○ What is CSS ○ Basics of CSS ○ CSS Inline Style ○ Source Code inline Styles ○ CSS Style Tag Styling ○ CSS selection example ○ Creating and using CSS file ○ Attach Style CSS file ○ CSS selection and classes ○ CSS element selection Class id and tag ○ CSS Colors. ○ CSS Color Selection and options example code ○ CSS Background Property ○ CSS Background Color style example ○ CSS Display Property ○ Measures and Properties CSS ○ CSS Divs and Spans ○ Source Code examples CSS divs and spans and styling ○ Styling Text CSS ○ CSS font Family ○ Google Fonts CSS ○ CSS float elements ○ CSS Floats Images ○ CSS Box model source example ○ CSS Box Model ○ CSS outline source code examples ○ CSS Float examples source ○ CSS position ○ CSS position example source code ○ CSS Outline ○ CSS Overflow and MaxWidth. ○ CSS and Tables ○ CSS for tables ○ CSS Element Alignments ○ CSS Pseudo Classes ○ Source Code ○ CSS fun with Hyperlinks ○ CSS Display Property ○ CSS3 – Border Image ○ CSS3 – Multi Background		

	○ CSS3 – Shadow ○ CSS3 – Gradients ○ CSS3 – Colors ○ CSS3 – Animation ○ CSS3 – Multi Columns ○ CSS3 – User Interface ○ CSS3 – Box Sizing ○ CSS3 – Responsive • Practical Exercise		
Lab-19-30	• **Practical**		
Lecture-31-37	• **Javascript** ○ Welcome to JavaScript ○ Section source code and resources ○ Write JavaScript ○ Update HTML element ○ Source code console example ○ JavaScript Code ○ JavaScript Variables ○ JavaScript Variables in Action ○ JavaScript Variables in Action ○ Operators and assignments ○ JavaScript Functions ○ Source Code JavaScript Functions ○ Closer look at functions ○ Source Code code functions ○ JavaScript Objects ○ JavaScript Arrays ○ Source Code Objects ○ Array Methods ○ Condition Statements JavaScript ○ Combine with Functions ○ JavaScript Switch ○ JavaScript Conditions and switch Example code ○ JavaScript Loops ○ Source Code Loops ○ Source Code Let and Const ES6 ○ JavaScript String Methods ○ Eventlisteners JavaScript ○ JavaScript Event listeners • ES6 JavaScript		
Lab-31-37	• **Practical**		
Lecture-38-42	• **JavaScript Document Object Model DOM** • Introduction to JavaScript DOM • What is the DOM • Source Code and Resources • JavaScript DOM Manipulation		

	• Source Code DOM • JavaScript DOM Selection • Element Text Manipulation • Change Classes DOM • JavaScript DOM Examples and source code • Change Styles DOM • Source Code and Examples DOM style and element attributes • Select Multiple Elements • Interactive DOM Events • JavaScript Mouse Events • KeyPress Events. • JavaScript DOM events • JavaScript DOM Create your own Elements • Event listeners to New elements • DOM Create elements • List Project Exercise Input add new Element. • Project Bonus Adding Object Value • JavaScript Project Source Code Project List • Create Dynamic List • DOM add event listeners create elements • Dynamic Values to Element Objects		
Lab-38-42	• **Practical**		
Lecture-43-53	• **jQuery** • jQuery Introduction to section • Source Code and resources • Why jQuery and What is it. • Selecting with jQuery • jQuery Fun with Styling • jQuery Source Code example • jQuery Content • jQuery text html append prepend example • jQuery Attributes • jQuery addClass removeClass html examples • Image and Form Attributes • jQuery Form and attribute examples • jQuery Click Events • jQuery text toggleClass click • Document Ready Event Listener • jQuery document ready example • jQuery Mouse Events		

	• Keyboard Events jQuery • jQuery On Event listener • jQuery Mouse Events Keyboard Events examples and source code • jQuery Effects Hide Show • jQuery hide show toggle example • jQuery Fade Effect • jQuery fadeIn, fadeout, fadeToggle • jQuery Slide Effect • jQuery Animate Effect • jQuery slideDown, slideUp, slideToggle • jQuery animate example source code • Load and CORS error • AJAX and Form data • Multiple Items in response output • Get API JSON data output to HTML page example • Select Element. • jQuery Select examples • Course jQuery Exercise • jQuery Project Source code • jQuery 3rd party plugins, Plugin Implementation, Plugin Development. • jQuery UI Accordion Widget, Tabs Widget, Tooltip Widget, Menu Widget, Widget, Datepicker Widget, jQuery Mobile.		
Lab-43-53	• **Practical**		
Lecture-54-58	• **Bootstrap** • Bootstrap - Environment Setup • Bootstrap with CSS • Bootstrap Layout Components • Bootstrap Layout Components • Bootstrap - Glyphicons • Bootstrap - Dropdowns • Bootstrap - Button Groups • Bootstrap - Button Dropdowns • Bootstrap - Input Groups • Bootstrap - Navigation Elements • Bootstrap - Navbar • Bootstrap - Breadcrumb • Bootstrap - Pagination • Bootstrap - Labels • Bootstrap - Badges • Bootstrap - Jumbotron • Bootstrap - Page Header • Bootstrap - Thumbnails • Bootstrap - Alerts		

	• Bootstrap - Progress Bars • Bootstrap - Media Object • Bootstrap - List Group • Bootstrap - Panels • Bootstrap - Wells • Bootstrap Plugins, Bootstrap Demos		
Lab-54-58	• **Practical**		
Lecture-59-70	• **Angular JS** • Angular Components and Data Binding • Working of Angular Applications • Angular App Bootstrapping • Angular Components • Creating A Component Through Angular CLI • Ways to specify selectors • Template and styles • Installing bootstrap to design application • Data Binding • Types of Data Binding • Component Interaction using @Input and @Output decorator • Angular Animations • Component Life-cycle Hooks • Directives and Pipes in Angular • Understanding Angular Directives • @Component Directive • Structural Directives • Attribute Directives • Custom Directives • Pipes • Built-in Pipes • Chaining pipes • Custom pipes • PipeTransform Interface & Transform Function • Pure and Impure pipes • **Angular Services and Dependency Injection** • Angular service • Need for a service • Dependency Injection • Creating a service • Hierarchical Injector • Injecting A Service into Another Service • Observables • RxJS Library • Angular's Interaction with Backend		

	• Parts of an Http Request • HttpClient • Angular Routes and Navigation • Angular Router • Setting Up Routes • Adding Routes Using RouterLink • Wildcard and Redirecting Routes • Adding Navigation Programmatically • Passing Route Parameters • Extracting Parameters Using ActivatedRoute • Optional Route Parameters • Child Routes • Route Guards • Location Strategies • **Handling Forms in Angular** • Angular forms • Types of forms • Underlying building blocks of the form model • Template-driven vs Reactive forms • Template-driven forms • Reactive Forms • Dynamically adding data to a form • **Validating Angular Forms** • What is Form Validation? • Types of Form Validation • Built-in Validators • Form control's status and validity • Form Validation methods • CSS classes for Form control • Custom validators in Template Driven Forms • **Authentication with JWT and Security in Angular** • What is Authentication? • Authentication and authorization • Types of Authentication • Where to store tokens? • JSON Web Tokens (JWT) • Authentication in Angular application • Security threats in web application • **Testing and Application Deployment in Angular** • Testing • Why should we perform testing? • Types of testing		

	• Testing Angular application using Jasmine and Karma • Maintaining application code using Git • Version control system • Why should we use Git? • Git file workflow • Running application on production server: Nginx • Architecture of Nginx • How to configure Nginx? • Deployment of an application using Docker • Problems before containers • How containers solve the problems • What is Docker? • Docker file • Docker image • Docker containers • Docker hub • Basic Docker commands		
Lecture-71-95	• **Introduction to Web Development and React** • Building Blocks of Web Application Development • Single-page and Multi-page Applications • Different Client-side Technologies • MVC Architecture • Introduction to React • Installation of React • JSX and its use case • DOM • Virtual DOM and its working • ECMAScript • Difference between ES5 and ES6 • NPM Modules • **Components and Styling the Application Layout** • React Elements • Render Function • Components • Class Component • Component Constructor • Functional Components • Multiple Components • Props • Props with Class based Component • Props with Function based Component • States		

	• Component Lifecycle • React Events • React Forms • Different Form Concepts • Styling in React • Inline Styling • CSS Stylesheet • Building Music Shop Application using React Components • **Handling Navigation with Routes** • Routing • react-router • Features of react-router • Configuration of routing using react-router • Navigation using Links • 404 page (Not found Page) • URL Parameters • Nested Routes • Implementing styles using NavLink • Application Programming Interface • Build a REST API using json-server • API consumption in React application using Fetch method • Build a dynamic Music Store application using Routing and API connectivity • **React State Management using Redux** • Need of Redux • What is Redux? • Redux Architecture • Redux Action • Redux Reducers • Redux Store • Principles of Redux • Pros of Redux • NPM Packages required to work with Redux • More about react-redux package • **Asynchronous Programming with Saga Middleware** • Need of Async operations • Async Workflow • Action Creators • How to write Action Creators? • Handling Async Actions via Reducers • Middleware • Redux-Saga • Generators in Redux-Saga		

	• Saga Methods() • Major Sections of Redux-Saga • Building a Product List application using Redux-Saga Middleware • Debugging application using Redux Devtools • **React Hooks** • Caveat of JavaScript classes. • Functional components and React hooks • What are React hooks? • Basic hooks • useState() hook • How to write useState() hook when state variable is an array of objects • useEffect() hook • Fetch API data using useEffect() hook • useContext() hook • Rules to write React hooks • Additional hooks • Custom hooks • **Fetch Data using GraphQL** • What is GraphQL? • Cons of Rest API • Pros of GraphQL • Frontend backend communication using GraphQL • Type system • GraphQL datatypes • Modifiers • Schemas • GraphiQL tool • Express framework • NPM libraries to build server side of GraphQL • Build a GraphQL API • Apollo client • NPM libraries to build client side of GraphQL • How to setup Apollo client • **React Application Testing and Deployment** • Define Jest • Setup Testing environment • Add Snapshot testing • Integrate Test Reducers		

	• Create Test Components • Push Application on Git • Deploy App on Nginx • Create Docker for React Application • **Introduction to React Native** • Native Applications • React Native • React Native Elements • Expo CLI • Build a shopping cart mobile application using React Native • React Native installation and setup • Working with Styles and Layout • **Building React Native Application with API** • Native modules • Native Navigation libraries • Integration of Redux with React Native • React Native and Redux major components • Redux Thunk middleware • NPM libraries • Shopping cart application using React Native and Redux • Integration of Redux actions, store and reducers In React Native application		
Lecture-96-120	• **Introduction to Node.js** • What is Node.js? • Why Node.js? • Installing NodeJS • Node in-built packages (buffer, fs, http, os, path, util, url) • Node.js Modules • Import your own Package • Node Package Manager (NPM) • Local and Global Packages • Push code to GitHub • **File System Module and Express.js** • Get Input from Users • Pass Multiple Arguments with Yargs • File System Module • Operations associated with File System Module • JSON Data • Http Server and Client		

	• Sending and receiving events with EventEmitters • Express Framework • Run a Web Server using Express Framework • Routes • Deploy application using PM2 and Nginx • **Asynchronous Programming** • Call Stack • Callbacks, Callback Queue and Event Loop • Callback Abstraction • Callback Chaining • Promises • Promise Chaining • Request Package • Customizing HTTP Requests • Error handing with appropriate HTTP codes • Introduction to template engine (EJS) • **Integration with MongoDB and Email Servers** • Introduction to NoSQL Databases and MongoDB • Installation of MongoDB on Windows • Installation of Database GUI Viewer • Inserting Documents • Querying, Updating and Deleting Documents • Connect MongoDB and Node.js Application • Exploring SendGrid • Sending emails through Node.js application using SendGrid • **REST APIs and GraphQL** • REST API • REST API in Express • Postman • MongoDB Driver API • Express Router • Mongoose API • GraphQL • GraphQL Playground • **Building Node.js Applications using ES6** • ES6 variables • Functions with ES6 • Import and Export withES6		

	• Async/Await • Introduction to Babel • Rest API with ES6 • Browsing HTTP Requests with Fetch • Processing Query String • Creating API using ES6 • Transpilation • Building Dashboard API • Creating dashboard UI with EJS • ES6 Aside: Default Function Parameters • Data Validation and Sanitization • **User Authentication and Application Security** • Authentication • Types of Authentication • Session Vs Tokens • JSON Web Tokens • Bcrypt • Node-localstorage • **Dynamic Client-Server Interaction using Socket.IO** • Web Sockets • Web Sockets • Socket.io • Broadcasting Events • Sharing Your Location • Event Acknowledgements • Form and Button States • Rendering Messages • Working with Time and Timestamps for determining Location of Messages • Storing Users, Rendering User List, Tracking Users Joining and Leaving • Deploying the Chat Application • Redis - Building API with Redis • **Testing Node.js Applications** • Writing Tests and Assertions • Testing Asynchronous Code • Testing an Express Application • Setup and Teardown • Testing with Authentication • Advanced Assertions • Mocking Libraries • Wrapping up User Tests • Setup Task Test Suite • Testing with Task Data • **Microservices Application** • Why Microservices?		

	• What is Microservices? • Why Docker? • What is Docker? • Terminologies in Docker • Child Processes • Types of child process		
Lecture-121-1	• **Introduction to MongoDB - Architecture and Installation** • Understanding the basic concepts of a Database • Database categories: What is NoSQL? Why NoSQL? Benefit over RDBMS • Types of NoSQL Database, and NoSQL vs. SQL Comparison, ACID & Base Property • CAP Theorem, implementing NoSQL and what is MongoDB? • Overview of MongoDB, Design Goals for MongoDB Server and Database, MongoDB tools • Understanding the following: Collection, Documents and Key/ Values, etc., • Introduction to JSON and BSON documents • Environment setup (live Hands-on) and using various MongoDB tools available in the MongoDB Package • • Case study discussion • **Schema Design and Data Modelling** • Data Modelling Concepts • Why Data Modelling? Data Modelling Approach • Analogy between RDBMS & MongoDB Data Model, MongoDB Data Model (Embedding & Linking) • Challenges for Data Modelling in MongoDB • Data Model Examples and Patterns • Model Relationships between Documents • Model Tree Structures • Model Specific Application Contexts • Use Case discussion of Data modeling • **CRUD Operations** • MongoDB Development Architecture • MongoDB Production Architecture • MongoDB CRUD Introduction, MongoDB CRUD Concepts		

	• MongoDB CRUD Concerns (Read & Write Operations) – • Concern Levels, Journaling, etc., • Cursor Query Optimizations, Query Behavior in MongoDB • Distributed Read & Write Queries • MongoDB Datatypes • **Indexing and Aggregation Framework** • Index Introduction, Index Concepts, Index Types, Index Properties • Index Creation and Indexing Reference • Introduction to Aggregation • Approach to Aggregation • Types of Aggregation (Pipeline, MapReduce & Single Purpose) • Performance Tuning. • **MongoDB Administration** • Administration concepts in MongoDB • Monitoring issues related to Database • Monitoring at Server, Database, Collection level, and various Monitoring tools related to MongoDB • Database Profiling, Locks, Memory Usage, No of connections, page fault etc., • Backup and Recovery Methods for MongoDB • Export and Import of Data to and from MongoDB • Run time configuration of MongoDB • Production notes/ best practices • Data Managements in MongoDB (Capped Collections/ Expired data from TTL), Hands on Administrative Tasks. • **Scalability and Availability** • Introduction to Replication (High Availability) • Concepts around Replication • What is Replica Set and Master Slave Replication? • Type of Replication in MongoDB • How to setup a replicated cluster & managing replica sets etc., • Introduction to Sharding (Horizontal Scaling), • Concepts around Sharding, what is shards, Key, • Config Server, Query Router etc. • How to setup a Sharding,		

	• Type of Sharding (Hash Based, Range Based etc.), and Managing Shards. • **MongoDB Security** • Security Introduction • Security Concepts • Integration of MongoDB with Jaspersoft • Integration of MongoDB with Pentaho • Integration of MongoDB with Hadoop/Hive • Integration of MongoDB with Java • Integration of MongoDB with GUI Tool Robomongo • Case Study MongoDB and Java • **Application Engineering and MongoDB Tools** • MongoDB Package Components • Configuration File Options • MongoDB Limits and Thresholds • Connection String URI Format/ Integration of any compatible tool with MongoDB API and Drivers for MongoDB • MMS (MongoDB Monitoring Service) • HTTP and Rest Interface • Integration of MongoDB with Hadoop and Data Migration MongoDB with Hadoop (MongoDB to Hive) • Integration with R • **MongoDB on the Cloud** • Overview of MongoDB Cloud products • Using Cloud Manager to monitor MongoDB deployments • Introduction to MongoDB Stitch • MongoDB Cloud Atlas • MongoDB Cloud Manager • Working with MongoDB Ops Manager • **Diagnostics and Fixes** • Overview of tools • MongoDB Diagnostic Tools • Diagnostics Commands • MongoDB Deployment • Setup & Configuration, Scalability, Management & Security • Slow Queries • • Connectivity		
Lab-151-180	• **Project Development**		

Course Includes:

Learn anytime anywhere.... We believe in quality.... Learn Online / Offline (Vadodara-Gujarat-India)

- 180 Hrs Instructor Led Training
- 180 Hrs Self-Paced Learning
- 50 Hrs Project work & Exercises
- Real-life Projects
- Certification and Job Assistance
- Free Access to Workshop & Webinar

Upon the completion of the Classroom Training, The BIT Certification is awarded upon successfully completing the Project after reviewed by experts

Upon the completion of the Online Training, The BIT Certification is awarded upon successfully completing the Project after reviewed by experts.

Website:	**Contact:**	**Email:**
www.bitbaroda.com **www.bitonlinelearn.com**	**9327219987**	**info@bitbaroda.com**