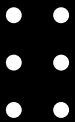


# MERN STACK

# Development

Course Outline

2024



+92 343 7491252

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

[info@upskillservices.com](mailto:info@upskillservices.com)

### Course Objective:

- Enable participants to develop a complete web application from the scratch that includes Front-end, Backend and Data-exchange technologies.
- Build strong foundations (ex: OOPS) in entry level engineers / working professionals thereby making them job ready as per latest industry requirements. Enable them to learn new technologies by applying foundation paradigms.
- By the end of the program participants will be become an industry-ready engineer who can be readily deployed in a project.

### Duration:

The duration of this bootcamp is **3 months**.

### Delivery Method:

Live virtual classes and intervention session (Assignment / projects, Live doubt clearing and evaluation).

### Learning Outcomes:

By the end of this Course, students will be able to

- Understand the fundamentals of web development, including HTML, CSS, and JavaScript.
- Develop front-end web applications using HTML, CSS, and JavaScript.
- Develop back-end web applications using a programming language such as PHP, Python, Flask, Django Java etc.
- Deploy web applications to a production environment.
- Work effectively as part of a team to develop web applications.
- Communicate effectively with both technical and non-technical audiences about web development projects.
- Identify and solve problems in a timely and efficient manner.
- Work as Freelancer, Entrepreneur etc.

## Front-end (HTML, CSS3, JAVASCRIPT)

### HTML:

- Introduction HTML
- HTML Basics
- HTML Elements
- HTML5 Semantic
- HTML Attributes
- HTML Headings
- HTML Paragraph
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Computer Code
- HTML Comments & Colors
- HTML CSS, Links and Images
- HTML Lists
- HTML Blocks
- HTML Classes
- HTML Layout
- HTML Responsive
- HTML iframes
- HTML JavaScript
- HTML Head
- HTML Entities and URI Code
- HTML Symbols and XHTML
- HTML Charset and Forms

### CSS3:

- Introduction CSS3
- CSS3 Syntax
- CSS3 How To
- CSS3 Colors
- CSS3 Backgrounds
- CSS3 Borders
- CSS Padding
- CSS Height/Width
- CSS3 Gradients
- CSS3 Shadows
- CSS3 Text
- CSS3 Fonts
- CSS3 2D Transforms
- CSS3 3D Transforms
- CSS3 Links
- CSS3 Lists
- CSS3 Tables
- CSS3 Box Model
- CSS3 Outline
- CSS3 Display
- CSS3 Max-width
- CSS3 Position
- CSS3 Float
- CSS3 Inline-block
- CSS3 Align
- CSS3 Combinators
- CSS3 Navigation Bar
- CSS3 Dropdowns
- CSS3 Tooltips
- CSS3 Images
- CSS3 Attr Selectors
- CSS3 Forms
- CSS3 Counters
- CSS3 Animations
- CSS3 Buttons
- CSS3 Pagination
- CSS3 Multiple Columns
- CSS3 User Interface
- CSS3 Box Sizing
- CSS3 Filters
- CSS3 Media Queries
- CSS3 Responsive

## Bootstrap:

- Introduction to Bootstrap
- Bootstrap Basics
- Bootstrap Grids
- Bootstrap Themes
- Bootstrap Layout
- Bootstrap Columns
- Bootstrap Typography
- Bootstrap Images
- Bootstrap Tables
- Bootstrap Figure
- Bootstrap Form
- Bootstrap Components
- Bootstrap Navbar
- Bootstrap Nav & tabs
- Bootstrap slides
- Bootstrap cards
- Bootstrap Buttons
- Bootstrap Utilities

## JavaScript:

- History of JavaScript
- Advantages
- Limitations
- Script element
- Creating your first JavaScript program
- Coding convention
- Setting up development environment (Vs Code)

## Types and Statements

- Keywords in JavaScript
- Overview of Data types
- Primitive Data types
- Non-primitive Data types
- Conditional statements
- I/O in JavaScript
- Loops

## Operators

- Introduction to operators
- Operator precedence and associativity
- Deep dive into operators
- Arithmetic
- Comparison
- Ternary
- Logical
- Language
- Bitwise

## Functions - Level I

- Introduction to functions
- Function definition
- Passing values
- Returning values
- Robust parameter handling
- Local and global variables

## Course Outline

- Functions as objects

### Functions- Level II

- Function invocation patterns
- Recursion functions
- Generator functions
- Arrow functions

### Arrays and Strings

- Introduction to Arrays
- Array declaration
- Array access methods

### Regular Expressions

- Introduction to RegExp
- Regular expression usage
- Modifiers
- RegExp patterns

### Objects in JavaScript

- Introduction to objects
- Type of objects in JavaScript
- Creating objects
- Object methods

### Event handling

- JavaScript events
- Event handler
- Event flow

### Document Object Model (DOM)

- Introduction to DOM
- Types of DOM
- DOM standards and methods
- Manipulating documents using DOM

### Browser Object Model (BOM)

- Function constructor

- Variadic functions
- JavaScript scopes
- Function closures

- Multi-dimensional arrays
- String properties
- String access methods

- RegExp methods
- String methods for RegExp
- Type conversion in JavaScript

- Constructor function
- Prototype in JavaScript
- Inheritance using prototype chain

- Event bubbling and capturing
- Event listeners
- Event types

- Handling images
- Table manipulation
- Animation
- Node and Node-list handling

- Introduction to BOM
- DOM vs BOM differences
- Window object and methods
- BOM navigator
- BOM history

## Form Handling

- Introduction to forms
- Form processing
- Forms object
- Accessing data from forms

## Debugging Techniques

- JavaScript Errors
- Error handling mechanisms
- Introduction to Google Chrome debugger
- Deep dive into debugger window

## jQuery:

- What Is jQuery?
- jQuery Selectors
- jQuery Introduction
- jQuery Syntax

- BOM location
- BOM timer
- Introduction to Cookies
- Session and persistent cookies

- Form validation
- Additional features in forms
- Validation APIs

- Introduction to Breakpoints
- Changing variable values in runtime
- Avoiding Mistake

- jQuery Events
- jQuery Effects
- jQuery HTML
- jQuery AJAX

## MERN Stack – Course Modules

### MongoDB

#### Module Objectives:

- Building strong expertise in Document Oriented Non-Relational Database Management System
- Implement frontend and backend scenarios to read, write and update data stored in MongoDB

#### Overview:

This course provides you hands-on experience and exposure to implement frontend and backend scenarios to read, write and update data using MongoDB. This course builds strong foundation for web application development based on client-server architecture.

#### Detailed course contents

##### CRUD Operations Using MongoDB

- Installing MongoDB
- Connecting to MongoDB
- Schemas
- Models
- Saving a Document
- Querying Documents
- Comparison Query Operators
- Logical Query Operators
- Regular Expressions
- Counting
- Pagination
- Updating Documents- Query First
- Updating a Document- Update First
- Removing Documents
- Mongoose - Data Validation

##### Mongoose - Data Validation

- Validation
- Built-In Validators
- Custom Validators
- Async Validators
- Validation Errors
- SchemaType Options

##### Mongoose- Modelling Relationships Between vConnected Data

- Referencing Documents
- Population
- Embedding Documents
- Using an Array of Sub-documents
- Transactions
- Object ID

- Validating Object ID's

### Authentication and Authorization

- Introduction
- Creating the User Model
- Registering Users
- Using Lodash
- Hashing Passwords

### Handling and Logging Errors Deployment

- Introduction
- Preparing the App for Production
- Getting Started With Heroku
- Preparing the App for Deployment
- Adding the Code to a Git Repository
- Deploying to Heroku
- Viewing Logs
- Setting Environment Variables
- MongoDB in the Cloud
- Authenticating Users
- Testing the Authentication
- JSON Web Tokens
- Generating Authentication Tokens
- Storing Secrets in Environment Variables
- Setting Response Headers
- Encapsulating Logic in Mongoose Models
- Authorization Middleware
- Protecting Routes
- Getting the Current User
- Logging Out Users
- Role Based Authorization
- Testing the Authorization

## Express JS

### Module Objectives:

Building Strong expertise on Express framework to develop responsive web application Implement MVC and responsive design to provide unified and intuitive user experience across PC, tablet and Mobile Phone

### Overview:

This course provides you strong foundation on Express framework driven by hands-on experience and exposure to develop responsive web application using MVC design pattern. During the course, you should be able to builds strong background about Express and Node.js architecture, usage of services and applying Node.js concept to develop frontend application supported by backend web servers.



## Detailed course contents

### Express Framework - (Building RESTful API's)

- Introduction
- RESTful Services
- Introducing Express
- Building Your First Web Server
- Nodemon
- Environment Variables
- Route Parameters
- Handling HTTP GET Requests
- Handling HTTP POST Requests
- Calling Endpoints Using Postman
- Input Validation
- Handling HTTP PUT Requests
- Handling HTTP Delete Requests

### Express - Advanced Topics

- Middleware
- Creating Custom Middleware
- Built-In Middleware
- Third-party Middleware
- Environments
- Configuration
- Debugging

### Asynchronous JavaScript

- Synchronous Vs. Asynchronous Code
- Patterns For Dealing With Asynchronous Code
- Callbacks
- Callback Hell
- Named Functions to Rescue
- Promises
- Replacing callbacks with Promises
- Consuming Promises
- Creating Settled Promises
- Running Parallel Promises
- Async and Await

## React JS

### Module Objectives:

- Create Single Page Applications (SPA) using React
- Learn core React features and use-cases
- Integrate React Applications into Firebase backend
- Deploy React applications into popular servers

### Overview:

This course provides you building SPA using React and its core features by taking a hands-on approach. Starting with Introduction this course deep dives into core React features like components, state and event binding and rendering. Followed by core features, participants will learn to integrate React with backend services (ex: Firebase) and consuming HTTP services in forms of REST APIs. The course exposes participants with advanced features like Routing, Authentication and authorization followed by deploying into popular servers (ex: GitHub).

### Detailed course contents

#### Introduction to React

- History of React
- Key Benefits of React
- React development environment
- Creating your first React Application
- React Source code structure

#### JSX

- Introduction to JSX
- Coding in JSX
- Expressions in JSX
- Working with HTML
- Conditional Constructs

#### Components

- Introduction to components
- Why Components?
- Writing JSX code in components
- Adding CSS
- Populating Data Dynamically
- Passing data through "props"
- Multiple Components

#### State and Event Binding

- Introduction to Events
- Event Handlers
- Working with state
- Data Binding
- Controlled and Uncontrolled Components

## Rendering Lists and Conditional Contents

- Rendering lists of data
- Using stateful list
- Keys in data
- Conditional Contents
- Adding dynamic styles
- Debugging Techniques
- Understanding error message
- Code flow and warnings
- Breakpoints
- Using React Dev Tools

## Class based Components

- What and Why?
- Adding first class based component
- Working with states and events
- Component Life cycle and in action
- Class based vs functional component

## HTTP

- Introduction to HTTP
- Methods in HTTP
- Response code in HTTP
- Introduction to REST interfaces
- Characteristics of REST
- Introduction to JSON
- JSON data representation
- GET request
- Using async and await
- Handling Http errors
- useEffects()
- POST request

## Custom React Hooks

- What are custom hooks?
- Creating and using custom hooks
- Custom HTTP hooks

## Forms

- Introduction to Forms
- Working with user input and submission
- Adding validation
- Working with custom hooks

## Routing and Navigation

- Introduction to Single Page Applications
- What is Routing?
- Installing react router
- Defining and using routes
- Working with links
- Dynamic routes with Params
- Nested routes
- Redirecting the user
- Working with query Param
- Sending and getting data via HTTP

## Authentication

- What is authentication?
- Why and How?
- Authentication tokens
- Setting up
- Adding signup
- Showing feedback to the user
- Adding User login
- Managing auth state with context
- Using tokens
- Redirecting the user
- Adding logout
- Protecting frontend pages
- Persisting User authentication status
- Adding auto logout

## Deployment

- Introduction
- Deployment steps
- Adding lazy loading
- Building the code for production
- Getting started with deployment
- Handling routes and finishing deployment

## Node JS

### Module Objectives:

- Building Strong expertise on Node.js core to develop web application
- Implement web application and deployment

### Overview:

This course provides you strong foundation on Node.js driven by hands-on experience and exposure to develop NodeJS based application and deployment. During the course you should be able to builds strong background about Node.js architecture, usage of services and applying Node.js concept to develop frontend application supported by backend web servers.

### Detailed course contents

#### Overview and Architecture

- What is Node
- Node Architecture
- How Node Works
- Installing Node
- Your First Node Program
- Event Loop

#### Node Module System

- Introduction
- Global Object
- Modules
- Creating a Module
- Loading a Module
- Module Wrapper Function

- Path Module
- OS Module
- File System Module
- Events Module

### **Node Package Manager (NPM)**

- Introduction
- Package .json
- Installing a Node Package
- Using a Package
- Package Dependencies
- NPM Packages and Source Control
- Semantic Versioning
- Listing the Installed Packages

### **Course Outline**

- Event Arguments
- Extending Event Emitter
- HTTP Module
  
- Viewing Registry Info for a Package
- Installing a Specific Version of a Package
- Updating Local Packages
- Dev Dependencies
- Uninstalling a Package
- Working with Global Packages
- Publishing a Package
- Updating a Published Package