Post Graduate Certification in Full Stack Web Development

Master the Art of Full Stack Web Development: Create Cutting-Edge Applications with Front-End and Back-End Expertise

Programme Head

Sathishkumar Kannan, MS (UK)

CEO, why global services

Course Name	**Post Graduate Certification in Full Stack Web Development**
Course Duration	**6 Months**
Course Objectives	• Gain a comprehensive understanding of web development concepts: The course aims to provide students with a solid foundation in web development, covering topics such as HTML, CSS, JavaScript, and other programming languages commonly used in web development. • Develop proficiency in front-end development: Students will learn how to create visually appealing and interactive user interfaces using HTML5, CSS3, and jQuery. They will gain hands-on experience in building responsive web designs and implementing best practices for front-end development. • Master back-end development skills: The course will cover server-side programming languages, frameworks, and technologies such as Node.js, Python. Students will learn how to build server-side applications, interact with databases, and handle data processing and storage. • Understand database management and integration: Students will learn about database concepts, including data modeling, querying, and database management systems such as MySQL or MongoDB. They will gain the skills to design and implement efficient database solutions for web applications. • Learn about web application frameworks: Students will be introduced to popular web application frameworks such as ReactJS & Django. They will understand the architecture, features, and best practices of these frameworks and learn how to build scalable and maintainable web applications. • Acquire knowledge of API development and integration: Students will learn about RESTful API design principles and techniques for building and integrating APIs into web applications. They will understand how to consume and utilize APIs to enhance the functionality and connectivity of their applications. • Develop skills in mobile app development: The course may include modules or units on mobile app development, where students will learn how to build mobile applications using frameworks like React Ionic. They will gain insights into mobile app design, user experience, and cross-platform development. • Explore deployment practices: Students will learn about tools for automating software

	development, testing, and deployment processes. They will understand how to set up and manage development environments, version control systems, and continuous integration and deployment pipelines. • Enhance problem-solving and critical thinking abilities: The course aims to develop students' analytical and problem-solving skills by challenging them with real-world web development scenarios and projects. They will learn how to approach complex problems, debug issues, and make informed decisions in the development process. • Build a portfolio of projects: Throughout the course, students will have the opportunity to work on hands-on projects that demonstrate their skills and knowledge in full stack web development. They will create a portfolio of projects that showcase their abilities to potential employers or clients.

Contents

Paper 1: Introduction to Full Stack Web Development

Unit 1: Introduction

- Understanding Web Development
- Importance of Full Stack Development
- The Role of a Full Stack Developer
- Trends in Web Development

Unit 2: How Websites Work

- Client-Server Model
- HTTP Protocol
- Web Browsers
- Request-Response Cycle
- DNS and Domain Hosting

Unit 3: Software Development Life Cycle (SDLC)

- Overview of SDLC
- Phases of SDLC
- Agile vs. Waterfall Methodologies
- Importance of SDLC in Web Development

Unit 4: Frontend Development

- Introduction to Frontend Development
- HTML (Hypertext Markup Language)
- CSS (Cascading Style Sheets)
- JavaScript and the Document Object Model (DOM)
- Responsive Web Design
- Frontend Frameworks

Unit 5: Backend Development

- Introduction to Backend Development
- Server-Side Programming
- Databases and Data Storage
- Backend Languages
- API Introduction

Unit 6: Git

- Version Control and Its Importance
- Introduction to Git and GitHub
- Git Basics: Commits, Branches, and Merges
- Collaborative Development with Git
- Best Practices in Git Workflow

Paper 2: User Interface & User Experience (UI/UX)

Unit 1: Introduction to UI/UX Design

- Understanding User Interface (UI) and User Experience (UX)
- Importance of Good UI/UX Design
- Role of UI/UX Designers
- Key Principles of UI/UX Design

Unit 2: Information Design and Wireframing

- Information Hierarchy and Organization
- Wireframing and Prototyping
- Low-Fidelity vs. High-Fidelity Wireframes

Unit 3: Visual Design and Graphic Elements

- Colour Theory and Psychology in Design
- Typography and Fonts
- Iconography and Symbolism
- Imagery and Visual Content
- Branding and Consistency

Unit 4: Figma

- Introduction to Figma
- Designing Artboards and Frames
- Working with Shapes and Vector Tools
- Adding and Styling Text
- Importing and Editing Images
- Using Components and Libraries
- Collaborative Design and Prototyping
- Designing Responsive Layouts
- Plugins and Integrations
- Design Handoff and Exporting Assets

Paper 3: Front-End Languages

Unit 1: Introduction to HTML

- Introduction to HTML
- HTML Document Structure
- Lists and Tables
- Links and Anchors
- Images and Multimedia
- Forms and Input Elements
- CSS and Styling in HTML
- HTML 5 Semantic Elements

Unit 2: Introduction to CSS

- Introduction to CSS
- CSS Syntax
- CSS Selectors
- CSS Box Model
- CSS Layout
- CSS Colors and Backgrounds
- CSS Typography
- CSS Flex and Grid
- CSS Transitions and Animations
- CSS Transformations
- CSS Positions
- CSS Pseudo-elements and Pseudo-classes

Unit 3: Bootstrap

- Introduction to Bootstrap
- Layout and Components
- Forms and Buttons
- Bootstrap CSS Classes & Utilities
- Responsive Web Design with Bootstrap

Unit 4: Introduction to JavaScript

- Introduction to JavaScript
- Adding JavaScript to Your Web Pages

- Difference Between Client-side and Server-side Scripting
- ES6 Concepts
- Variables and Data Types
- Operators
- Control Structures
- Functions
- Classes
- JavaScript Objects
- JavaScript Async
- DOM (Document Object Model)
- OOPS Concepts (Object-Oriented Programming)
- Error Handling
- Regular Expressions (Regex)

Paper 4: Front-End Framework (ReactJS)

Unit 1: Introduction to React

- What is React?
- Why Use React ?
- React Vs Other Frameworks

Unit 2: Setting Up Your Development Environment

- Node.js and npm installation
- Create React App
- Understanding the Project Structure

Unit 3: Hello World in React

- Create a Simple React Component
- JSX syntax

Unit 4: Understanding components

- Functional components
- Class components
- State and lifecycle methods

Unit 5: Props and State

- Passing data with props
- Managing component state
- setState Function

Unit 6: Handling Events

- Event handling in React
- Updating state with events
- Common event paterns

Unit 7: JSX Syntax in Depth

- Expressions in JSX
- Conditional rendering
- List and Keys

Unit 8: Rendering Lists and Keys

- Mapping through arrays
- Adding Keys to List items

Unit 9: Working with Forms

- Controlled components
- Handling Form Submission
- Form Validation

Unit 10: React Forms Best Practices

- Controlled components Vs Uncontrolled components
- Form libraries (e.g.Formik)

Unit 11: Introduction React Router

- Setting Up to React Router
- Navigation between Routes

Unit 12: Nested Routes and Route Parameters

- Creating nested routes
- Passing and accesing parameters

Unit 13: Introduction to Redux

- State Management in Redux
- Actions, Reducers and the store

Unit 14: Connecting React with Redux

- Using react redux library
- Mapping state and dispatch props

Unit 15: Introduction to React Hooks

- UseState, useEffect, useContext
- Custom hooks

Unit 16: Building a Custom Hook

- Understanding the use of Custom Hooks
- Creating the custom Hook

Unit 17: Working with API’s

- Fetching Data in React
- Using ‘fetch’ for API Calls
- Axios for data fetching
- Handling asynchronous operations

Unit 18: State Management with API Data

- Updating state with API data
- Error handling and loading status

Unit 19: Testing in React

- Introduction to Testing
- Best Practices for Testing in React

Unit 20: Deployment

- Building and Deploying React Applications

Paper 5: Mobile Application Development (React Ionic)

Unit 1: Introduction to React and Ionic

- What is React?
- What is Ionic?
- Setting up the development environment (Node.js, npm, create-react-app).

Unit 2: Setting Up Ionic in a React App

- Installing Ionic and React
- Understanding the Ionic folder structure
- Understanding Ionic CLI and Commands

Unit 3: Building Basic UI with Ionic Components

- Commonly used Ionic components (Button, Card, Header, etc.)
- Using Ionic’s grid system for responsive design.

Unit 4: Navigation with React Router

- Setting up React Router with Ionic
- Stateful and stateless widgets

Unit 5: Managing State with React Hooks

- Using `useState` and `useEffect` hooks
- Understanding component lifecycle

Unit 6: Working with APIs

- Using `fetch` or `axios` for API calls
- Handling asynchronous operations in React

Unit 7: Adding Theming and Customization

- Using CSS variables for theming
- Customizing component styles

Unit 8 : Adding Capacitor or Cordova to Ionic App

- Understanding Capacitor and Cordova
- Adding Android and iOS platforms to Ionic app
- Adding Native features

Unit 9: Building and Deploying Your App

- Building the app for production
- Deployment options (Netlify, Vercel, etc.)

Unit 10: React Ionic App Deployment

- A home page showing tasks
- Ability to add and delete tasks
- Navigation between a home page and an about page

Paper 6: Backend Development

6.1 Python

Unit 1: Introduction to Python

- Overview of Python
- Installing Python
- Python development environments

Unit 2: Variables, Data Types, and Operators

- Understanding Python variables
- Data types in Python
- Operators in Python

Unit 3: Control Flow and Looping

- Conditional statements (if, else, elif)
- Loops in Python (for, while)
- Flow control in Python

Unit 4: Functions and Modules

- Creating and using functions in Python
- Modular programming in Python
- Python modules and libraries

Unit 5: Data Structures

- Working with data structures (lists, dictionaries, sets, etc.)
- Manipulating data using data structures
- Data structure selection

Unit 6: File Handling and Input/Output

- Reading and writing files in Python
- File handling operations
- Working with text and binary files

Unit 7: Object-Oriented Programming (OOP) in Python

- Introduction to OOP concepts in Python
- Classes and objects
- Inheritance, encapsulation, and polymorphism in Python

Unit 8: Exception Handling

- Handling exceptions in Python
- Using try, except, and finally blocks
- Custom exceptions

Unit 9: Python Libraries and Packages

- Overview of popular Python libraries
- Using third-party packages with pip
- Common use cases for Python libraries

Unit 10: Advanced Topics in Python

- Advanced Python concepts (decorators, generators, context managers, etc.)
- Multithreading and multiprocessing
- Working with databases, web APIs, and more

6.2 Django

Unit 1: Introduction to Django

- **Overview of Django**
    - What is Django?
    - Why use Django for web development?
- **Setting Up the Development Environment**
    - Installing Python and Django
    - Setting up a virtual environment

Unit 2: Django Basics

- **Creating a Django Project**
    - Using the django-admin command
    - Project structure and files
- **Building the First Django App**
    - Creating a Django app

  - Understanding models, views, and templates
- **Django Admin Interface**
  - Exploring the admin panel
  - Creating and managing models through the admin

Unit 3: Working with Models and Databases

- **Defining Models**
  - Creating models and fields
  - Migrations in Django
- **Querying the Database**
  - Performing CRUD operations
  - Django ORM basics

Unit 4: Views and Templates

- **Writing Views**
  - Handling HTTP requests
  - Connecting views to URLs
- **Creating Templates**
  - Building dynamic templates
  - Template inheritance and rendering

Unit 5: Django Forms

- **Building Forms**
  - Creating HTML forms in Django
  - Form handling and validation

Unit 6: Django and Frontend

- **Integrating Bootstrap**
  - Adding Bootstrap for styling
  - Creating a responsive design

Unit 7: Building a Simple Web Application

Paper 7: Backend Development (Node.js and Express.js)

Unit 1: Introduction to Node.js and Installation

- What is Node.js?
- Installing Node.js and npm
- Node.js runtime environment

Unit 2: How to Use the Native Node Modules

- Utilizing native modules in Node.js
- Common built-in modules in Node.js
- Extending Node.js with C/C++ addons

Unit 3: NPM Package Manager and Installing External Node Modules

- Introduction to npm
- Managing packages with npm
- Installing and using external Node.js modules

Unit 4: Introduction to Express.js with Node.js

- Overview of Express.js
- Building web applications with Express.js
- Express.js middleware and routing

Unit 5: Creating Our First Server with Express

- Creating a basic server with Express.js
- Handling HTTP requests and responses
- Routing and serving static files

Unit 6: HTTP Requests in Node.js

- Making HTTP requests in Node.js
- Consuming APIs with Node.js
- Promises and async/await for asynchronous operations

Unit 7: Postman Introduction and Usage

- Introduction to Postman

- API testing and documentation
- Using Postman for API development and testing

Unit 8: Introduction to Middleware in Node.js

- Understanding middleware in Express.js
- Creating and using middleware
- Middleware in request processing pipeline

Unit 9: Custom Middleware Creation in Node.js

- Building custom middleware functions
- Handling authentication, logging, and other tasks
- Middleware order and execution flow

Unit 10: Authentication and Authorization with JSON Web Token (JWT) in Node.js

- Implementing authentication and authorization
- Using JWT for secure token-based authentication
- Role-based access control

Unit 11: Error Handling and Debugging in Node.js

- Error handling strategies in Node.js
- Debugging Node.js applications
- Utilizing debugging tools and techniques

Paper 8: Database Management System (DBMS)

8.1 Introduction of Databases

Unit 1: Introduction to Databases

- Understanding databases and their importance
- Types of databases
- Relational vs. non-relational databases

Unit 2: Relational Database Management Systems (RDBMS)

- Introduction to RDBMS
- Key concepts: tables, rows, columns, and keys

- Examples of popular RDBMS (e.g., MySQL, PostgreSQL)

Unit 3: Database Design and Modeling

- Database design principles
- Data modeling techniques
- Normalization and denormalization

8.2 SQL

Unit 1: Introduction to SQL

- Overview of Databases and SQL
- Setting Up a Practice Environment

Unit 2: SQL Basics

- SQL Syntax and Structure
    - Understanding SQL statements
    - Components of a SQL query
- Retrieving Data with SELECT
    - Basic SELECT statements
    - Filtering and sorting data
- Filtering and Sorting Data
    - WHERE clause for conditions
    - ORDER BY clause for sorting

Unit 3: Manipulating Data

- Inserting Data
    - Adding records to a table
    - Handling auto-incrementing columns
- Updating and Deleting Data
    - Modifying existing records
    - Removing records from a table

Unit 4: Advanced SQL Queries

- Aggregation Functions
    - Using functions like COUNT, SUM, AVG
    - GROUP BY clause for grouping data
- Joins and Relationships

    - Understanding JOIN operations
    - Working with multiple tables

Unit 5: Data Manipulation Language (DML)

- Transactions
    - Introduction to transactions
    - COMMIT and ROLLBACK statements
- Subqueries
    - Using subqueries in SELECT statements
    - Correlated subqueries

Unit 6: Data Definition Language (DDL)

- Creating and Modifying Tables
    - Creating tables with CREATE TABLE
    - Modifying table structure with ALTER TABLE
- Indexes and Constraints
    - Adding indexes to improve performance
    - Implementing constraints for data integrity

8.3MongoDB

Unit 1: Introduction to MongoDB

- What is MongoDB?
- Document-based data modeling
- CRUD operations in MongoDB

Unit 2: MongoDB Atlas Explanation and Compass Installation

- MongoDB Atlas as a cloud database service
- Installing MongoDB Compass for GUI access
- Connecting to MongoDB databases

Unit 3: MongoDB Concepts Overview

- Collections, documents, and fields in MongoDB
- BSON data format
- Indexing and querying in MongoDB

Unit 4: Schema Connection with Express.js

- Integrating MongoDB with Node.js and Express.js
- Using the Mongoose library for schema modeling
- Creating models and connecting to MongoDB

Unit 5: CRUD Operations in MongoDB

- Performing CRUD operations in MongoDB
- Inserting, updating, retrieving, and deleting documents
- Querying with conditions

Unit 6: Data Modeling in MongoDB

- Designing data models in MongoDB
- Embedding and referencing documents
- Schema design for efficient queries

Paper 9: Software Testing

Unit 1: Introduction to Software Testing

- Basics of software testing
- Importance of testing in the software development lifecycle
- Common misconceptions about testing

Unit 2: Software Development Life Cycle (SDLC) and Testing

- Phases of SDLC and their relevance to testing
- Different software development methodologies (Waterfall, Agile, etc.)
- How testing fits into each methodology

Unit 3: Testing Levels

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing
- Alpha and Beta Testing

Unit 4: Testing Types

- Functional Testing
- Non-Functional Testing (Performance, Security, Usability, etc.)
- Regression Testing
- Smoke and Sanity Testing
- Exploratory Testing

Unit 5: Test Planning and Documentation

- Test strategy and test plan
- Test cases and test scripts
- Traceability matrix
- Test data and test environment setup

Paper 10: Deployment

Unit 1: Introduction to Deployment

- Understanding the importance of deployment in web development
- Deployment strategies and considerations

Unit 2: Web Servers and Hosting

- Types of web servers
- Choosing a hosting provider

Unit 3: Domain and DNS Setup

- Registering a domain name
- Configuring DNS settings
- Handling domain redirects and subdomains

Unit 4: Version Control and CI/CD

- Integrating version control (e.g., Git) into the deployment workflow
- Setting up Continuous Integration and Continuous Deployment (CI/CD) pipelines

Unit 5: Server Configuration and Security

- Configuring server settings and environment variables
- Implementing security measures (e.g., HTTPS, firewalls)

Unit 6: Deploying Static Websites

- Deploying simple HTML/CSS/JavaScript websites

Unit 7: Deploying Dynamic Web Applications

- Deploying web applications built with frameworks
- Setting up databases and application servers

Paper 11: Capstone Project

Any of the below projects or customised projects:

Ecommerce Website

- Developing an ecommerce website
- Features, user experience, and payment integration

Social Media Platform

- Building a social media platform
- User profiles, posts, and interactions

Online Learning Management System

- Creating an online learning platform
- Courses, user registration, and progress tracking

Task Management Application

- Developing a task management app
- Task creation, assignment, and tracking

Event Booking Platform

- Building an event booking and management platform
- Event listings, booking, and ticketing