



**Ed-Tech**  
A F R I C A

# FULL STACK WEB DEVELOPMENT

6 Months Course





# Ed-Tech

A F R I C A

---

Transforming education  
through technology

---





# FULL-STACK WEB DEVELOPMENT

Full-stack web development refers to the practice of building both the front end (client-side) and back end (server-side) components of a web application. A full-stack web developer is someone who is proficient in working with both the user interface and the server-side logic, making them capable of handling all aspects of web development.

Here's a breakdown of the key components involved in full-stack web development:

- **Front-end Development (Client-Side)**
  - HTML/CSS/JavaScript
  - Front-end Frameworks/Libraries
- **Back-end Development (Server-Side)**
  - Server-Side Programming Languages
  - Server
- **Database Management**
  - Database Systems
  - ORM (Object-Relational Mapping)
- **Version Control/Git**

- **APIs (Application Programming Interfaces)**
  - RESTful APIs
  - Integration with External Services
- **Deployment and Hosting**
  - DevOps
  - Cloud Platforms

Full-stack web development is characterized by a holistic understanding of the entire web development process, allowing developers to work on different layers of the technology stack and deliver end-to-end solutions.

## JOB OPPORTUNITIES

- **Web Developer**
- **Full-Stack Developer**
- **Software Engineer**
- **Technical Lead or Architect**
- **Freelancer or Consultant**
- **Startups and Entrepreneurship**
- **E-commerce Development**
- **Content Management System (CMS) Development**
- **Mobile App Development**
- **Specialized Areas**



## Curriculum Index

### Python

#### Introduction To Python

- Why Python
- Application areas of python
- Python implementations
  - Cpython
  - Jython
  - Ironpython
  - Pypy
- Python versions
- Installing python
- Python interpreter architecture
  - Python byte code compiler
  - Python virtual machine(pvm)

#### Writing and Executing First Python Program

- Using interactive mode
- Using script mode
  - General text editor and command window
  - Idle editor and idle shell
- Understanding print() function
- How to compile python program explicitly

#### Python Language Fundamentals

- Character set
- Keywords
- Comments
- Variables
- Literals
- Operators
- Reading input from console
- Parsing string to int, float

#### Python Conditional Statements

- If statement
- If else statement
- If elif statement
- If elif else statement
- Nested if statement

#### Looping Statements

- While loop
- For loop
- Nested loops
- Pass, break and continue keywords

#### Standard Data Types

- Int, float, complex, bool, nonetype
- Str, list, tuple, range
- Dict, set, frozenset

#### String Handling

- What is string
- String representations
- Unicode string
- String functions, methods
- String indexing and slicing
- String formatting

#### Python List

- Creating and accessing lists
- Indexing and slicing lists
- List methods
- Nested lists
- List comprehension

## Python Tuple

- Creating tuple
- Accessing tuple
- Immutability of tuple

## Python Set

- How to create a set
- Iteration over sets
- Python set methods
- Python frozenset

## Python Dictionary

- Creating a dictionary
- Dictionary methods
- Accessing values from dictionary
- Updating dictionary
- Iterating dictionary
- Dictionary comprehension

## Python Dictionary

- Creating a dictionary
- Dictionary methods
- Accessing values from dictionary
- Updating dictionary
- Iterating dictionary
- Dictionary comprehension

## Python Functions

- Defining a function
- Calling a function
- Types of functions
- Function arguments
  - Positional arguments, keyword arguments
  - Default arguments, non-default arguments
  - Arbitrary arguments, keyword arbitrary arguments
- Function return statement
- Nested function
- Function as argument
- Function as return statement
- Decorator function
- Closure
- Map(), filter(), reduce(), any() functions
- Anonymous or lambda function

## Modules & Packages

- Why modules
- Script v/s module
- Importing module
- Standard v/s third party modules
- Why packages
- Understanding pip utility

## File I/O

- Introduction to file handling
- File modes
- Functions and methods related to file handling
- Understanding with block

## Regular Expressions(Regex)

- Need of regular expressions
- Re module
- Functions /methods related to regex
- Meta characters & special sequences

## Object Oriented Programming

- Procedural v/s Object Oriented Programming
- OOP Principles
- Defining a Class & Object Creation
- Inheritance
- Encapsulation
- Polymorphism
- Abstraction
- Garbage Collection
- Iterator & Generator

## Exception Handling

- Difference Between Syntax Errors and Exceptions
- Keywords used in Exception Handling
- try , except , finally , raise , assert
- Types of Except Blocks
- User-defined Exceptions

## GUI Programming

- Introduction to Tkinter Programming
- Tkinter Widgets
- Layout Managers
- Event handling
- Displaying image

## Multi-Threading Programming

- Multi-processing v/s Multi-threading
- Need of threads
- Creating child threads
- Functions /methods related to threads
- Thread synchronization and locking

## SQL Using MySQL

### Introduction to RDBMS

- What is Relational Database Package
- Difference between SQL & Database
- Installing MySQL Server database

## SQL Basic

- DDL: Create, Alter, Drop, etc.
- DML: Insert, Update, Delete ,etc.
- DQL : Select
- Autoincrement field
- SQL Comments
- SQL Aliases
- Savepoint & rollback

## SQL Constraints

- Not NULL, Unique key
- Primary key, Check
- Default, Foreign key
- Conditional operators
- Like, between, in operators

## SQL Clauses

- Order by
- Where
- Limit/top
- Group by
- having

## SQL Joins

- Inner Join
- Left Join
- Right Join
- Full Join

## SQL View

- creating view
- updating view
- fetching data from view

## SQL Functions

- String functions
- Aggregate functions
- Date & time functions

## Stored Procedures & Functions

- Understanding stored procedures and their key benefits
- Working with stored procedures
- Studying user-defined functions

## Working with CSV Files:

- How to write result to csv file
- How to read csv file

## Python Database Connectivity

- Database Drivers and connectors
- Creating connection object
- Understanding cursor object
- Executing SQL statements using cursor
- Fetching records from cursor
- Storing and retrieving Date and Time

## MONGODB

### Introduction To MongoDB

- Understanding NoSQL DB
- NoSQL vs. SQL DB
- Understanding Mongo DB
- Downloading & Installation
- Introduction of MongoDB shell and Compass
- Understanding database, collection & document

### Crud Operations

- Insert Document
- Delete Document
- Update Document
- Query Document

### Operators In MongoDB

- Query and Projection operators
- Update operator
- Aggregation Pipeline operators

### Methods In MongoDB

- limit and sort
- bulk methods
- other methods

### Indexing And Relationships

- Types of Indexes
- Creating an Indexes
- Dropping an Indexes
- Defining Relationships between Documents

### Python Connectivity With MongoDB

- Introduction to pymongo
- Installing pymongo module
- MongoClient
- Getting database and collection
- CRUD operations
- Range Query

## Front End

### Introduction to WEB

- What are web applications?
- Static v/s dynamic web applications
- Web application Architecture
  - Front End
  - Back End



## HTML

- Introduction to HTML 5
- Parts in HTML Document
- Head Section
- Meta Information
- Body Section
- Heading & Paragraph
- HTML FORMS
- Anchors, Images
- HTML Comments
- HTML Table
- DIV Section

## CSS

- Introduction CSS3
- Inline CSS
- Internal CSS
- External CSS
- Styling Text
- Styling Fonts
- CSS Borders
- Selectors
- Backgrounds and Borders
- Text Effects
- Margin & Padding

## JavaScript

- What is Script?
- Introduction to JavaScript
- DOM and BOM
- Comments and Types of Comments
- Popup Boxes
- Variables & Operators JavaScript
- Functions and Events Conditional
- Statements Looping Control
- Types of Errors
- Exception Handling
- Java Script Objects
- Browser Objects
- Validations in JS

## Typescript

- Why Typescript
- Basic Types
- Class and Interfaces
- Modules

## Bootstrap

- Introduction to Bootstrap4
- Bootstrap CDN & Local
- Bootstrap Classes
- Bootstrap Forms
- Bootstrap Buttons
- Bootstrap Colors
- Bootstrap Grid System

## Introduction To React Js

- What is React JS?
- What is SPA?
- DOM vs Virtual DOM
- Advantages and Disadvantages
- Key Features

## Environmental Setup

- Node | NPM
- Installation of CLI
- Setup Project
- Directory Structure
- Code Editors
- How React JS Application Boot

## Basic Features Of React Js

- React Concepts
- JSX and TSX
- Render Elements
- Function and Class Components
- Props and State
- Handling Events
- Dynamic Data Rendering
- Property Binding

## Key Features Of React Js

- Conditional Rendering
- List and Keys
- Forms Handling
- Forms Validations

## Component Lifecycle Hook

- Understanding component lifecycle
- All Lifecycle Hooks

## Event Handling React

- Understanding React Event System
- Passing arguments to event Handlers

## Network Call

- Fetch
- Axios

## Custom Services

- Introduction to Services
- Building a Service

## Local data storage

- Local Storage
  - Session Storage
  - Cookies
-

## Routing With React Router

- Setting up React Router
- Configuring route with Route Component
- Making routes dynamic with Route Params
- Working with nested routes
- Link and NavLink
- Redirect Routes

## UI Components

- Angular Material
- PrimeNG

## Introduction To Redux

- Why Redux
- Install and setup
- Store ,Reducer , actions
- Dispatcher
- High order Components
- mapStateToProps and mapDispatchTo-Props usage

## Advance Redux

- Async Actions
- Middleware
- Redux Thunk and Redux Saga

## React Hooks

- Why We Need Hooks.
- Different Types Of Hooks
- Using State And Effect Hooks
- Usereducer , Userref Etc.
- Custom Hooks
- Rules Of Hooks

## Third Party Modules

- Social Login
- Pagination
- Search
- Filter
- JWT Token
- File Upload
- Many More

## Django

### Getting Started With Django

- Language v/s Framework
- What is Django Framework?
- Django Versions
- Installing Django

## Understanding Django Project

- Creating Django Project
- Creating Django App
- Understanding Directory Structure of Project
- UrlMappings, Views,manage.py
- HttpRequest and HttpResponse Objects
- Registering App in settings.py
- Django Development Server
- Control flow of request processing
- MVT Design Pattern

## Django Templates (Presentation Logic)

- Built-in Templates Tags & Filters
- Template Variables
- Template Inheritance
- Building custom template tag
- Integration of Static Contents
  - How to display static images
  - How to use CSS and bootstrap
  - Loading static data to template

## Understanding Django Views (Business Logic)

- Why this Component?
- Types of Views
  - Function Based
  - Class Based
- How Views interact with other components
- How to get request data in views
- How to generate response in other formats

## Understanding Django Forms

- How to render django form to template
- Form Validations
- GET and POST methods
- Form class &Field Types
- Why this Component?
  - as table
  - as paragraph
  - as li
- How to use Form in views

## Django Models (ORM)

- Why this component?
- SQL v/s ORM approach
- create (insert)
- Performing CRUD operations
  - read (select)
  - update
  - delete



- Fetching Records from database
  - Understanding QuerySet
  - Fetching all records
  - Fetching records based on conditions
  - Fetching Data in an order
  - Data Slicing
- Understanding Django Forms
  - Integration with MySQL database
  - many to many
  - many to one
  - one to many
  - one to one
- Integration with MySQL database

### **Django Admin Interface**

- How to create super user
- How to access admin site
- Registering models with admin
- Managing users in the admin

### **Django State and Session Handling**

- Understanding stateless behavior of http protocol
- Why maintain state?
- State management techniques
  - Cookies
  - URL-Rewriting (query strings)
  - Hidden form fields
  - Session
- Understanding Sessions in Details
  - Enable / Disable Session
  - Get and set data with session
  - Using session in template
  - Understanding Sessions in Details
  - Session Expiry

### **Django Middleware**

- Django User Authentication
- Working with User objects
- Permissions and authorization
  - Authentication in web requests
  - Managing users in the admin
- How to enable custom middleware

### **Django User Authentication**

- Working with User objects
- Permissions and authorization
- Authentication in web requests
- Managing users in the admin
- Extending the existing User Model

### **Developing Web Services(APIs)**

- What is a Web API?
- Difference between SOAP and RESTful APIs
- Django REST Framework for developing RESTful APIs

### **Django Middleware**

- Why middleware?
- Understanding built-in middleware
- Creating custom middleware
  - Function based
  - Class based
- How to enable custom middleware

### **Django User Authentication**

- Extending the existing User Model
- Managing users in the admin
- Authentication in web requests
- Permissions and authorization
- Working with User objects

### **Developing Web Services(APIs)**

- What is a Web API?
- Difference between SOAP and RESTful APIs
- Django REST Framework for developing RESTful APIs
- Understanding JSON response
- How to consume RESTful API
  - requests package
  - get and post
  - how to parse JSON

### **Miscellaneous**

- Sending email using django library
- Pagination
  - Function based
  - Class based

### **Project Deployment (Making Project LIVE)**

- Understanding domain and hosting
- Hosting Providers
- Deploy the project



**Ed-Tech**  
A F R I C A

+267 3914472

ED Tech Africa

Ed-Tech Africa

+267 75 546 649

@edtech.bw

edtechafricabw

[www.ed-techafrica.com](http://www.ed-techafrica.com)



[www.ed-techafrica.com](http://www.ed-techafrica.com)