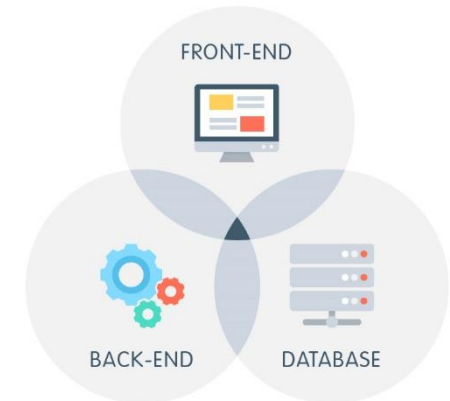




Full Stack Development Course Guide

FSD training curriculum is designed to develop the skills necessary to build the full fledged professional web application. In this program, you'll prepare for a job as a Full Stack Web Developer, and learn to create front-end as well as complex server-side web applications that uses powerful databases to persistently store data.

FULL-STACK DEVELOPMENT



FSD contains globally accepted technologies

FSD is a collection of technologies that are trusted and accepted by leading employers, educational institutions and governments worldwide.



Get all the support you need

ICT Trainings offers lots of support to help you prepare for your future.

Find a wide range of support, including official ICT preparation materials – www.ictrainings.com/

Find quizzes and learning tips on our Facebook page – www.facebook.com/ICTtrainingsLahore

Register now

You can take the course in a class or online on computer.

Step 1: Visit ICT Trainings Centre

Located at 349 H/3, Johar Town, Lahore, Pakistan.

Step 2: Choose from over 50 Courses

Find out about registering, costs and preparation courses.

Step 3: Choose your starting date and time

We have dates every month – and you can register just one week before taking your course.

Find out more at

www.ict-trainings.com

Show you are serious about your career

Join millions of people that have selected IT and using in official/daily chores.

ICT Certificates to show they have the work-focused

IT and Telecom skills for career success. These high-quality international courses are developed by Industry leading professionals

Boost your employability – get the IT skills you need to access the best jobs.

Excel in business-related studies with proven IT skills.

Recognized by top local & multinationals as a quality delivering educational institution.

Contact us

ICT Trainings Lahore

349 H/3, Johar Town
Lahore
Punjab
Pakistan
Cell: +92 321 3499131
LandLine: +92 42 35951406-7



Website: ict-trainings.com

<http://facebook.com/ICTtrainingsLahore>

www.ict-trainings.com/curriculum/full-stack-development-leaflet.pdf

ICT develop and produce the most valuable range of qualifications for learners of IT and Telecom.

Over 20000 students trained in Two Decades.

Universities, employers, government, ministries and other organizations recognize us as valuable partner.
ICT Trainings Institute – Engineering Your Career

Full Stack Development Course Contents

Learn full Stack JavaScript programming, developing real-time applications and the most refined, up-to-date curriculum.

Front-End Development

HTML

- Fundamentals of HTML
- Basic HTML tags
- Format text on Web Pages
- Incorporate images
- Create hyperlinks
- Create complex image maps
- Create tables and nested tables
- Insert a form on a web page
- Set, modify form field properties: text field, drop-down, check box, radio button
- Validating HTML

Cascading Style Sheet - CSS

- Introduction
- Designing with Style Sheets
- Style Sheet Syntax
- ID, Class Contextual Selectors
- Linked (External) Style Sheets
- Cascading Order
- Properties: Text, Font, Colors, Backgrounds, Borders
- Floating Elements
- Absolute and Relative Positioning i.e. Div's etc.
- Layering Elements with the Z-Index
- Animation of objects

JavaScript & Document Object Model - DOM

- Introduction to JavaScript
- Variables and Objects
- Decision Making Statement: If, Switch
- Loops: For, While & Do While
- Arrays
- Functions and Prototypes
- Core JavaScript Objects
- DOM Introduction
- DOM Event Model
- DOM Functions i.e. Document.Write, Document.GetElementById, Document.bgColor etc.

JQuery

- Introduction
- Install & Configuration
- jQuery Syntax
- Selectors

- Effects: Hide/Show, Fade, Slide, Animate, Stop, CSS
- Events: Click/DbClick, Focus, Blur, Mousemove/leave etc
- jQuery Callback
- jQuery Chaining

JSON

- Core Concepts of JSON
- Arrays
- JSON Parse
- JSON Stringify
- Accessing JSON via AJAX in Javascript

Bootstrap

- Core Concepts of Bootstrap
- Bootstrap Classes
- Alerts, Buttons, Badges, Cards
- Carousel
- Forms
- Modal
- Bootstrap Grid System

Single Page Application

Typescript

- What is TypeScript?
- Language Overview
- Type Annotations [Datatypes]
- Type Assertions [Type Casting]
- Inferred Typings
- Expressions & Operators
- Control Structures
- Arrays
- OOPs in TypeScript
- Creating Classes & Objects
- Transpiling Typescript to Javascript

Angular

Introduction and Overview

- Course Introduction
- What is Angular?
- Features and benefits of Angular
- So many Angular versions! What's up with them and which version does this course cover?
- Angular vs Angular 2 vs Angular ++

CLI Deep Dive and Troubleshooting

- Installing Angular
- Creating New Project

A Basic Project Setup using Bootstrap for Styling

- How an Angular App gets Loaded and Started
- Project Setup and First App
- Editing the First App

Architecture

- Architecture Overview
- Intro to Modules
- Intro to Components
- Intro to Services and DI

Components

- Components are Important!
- Creating a New Component
- Understanding the Role of AppModule and Component Declaration
- Using Custom Components
- Creating Components with the CLI & Nesting Components

Components & Templates

- Displaying Data
- Template Syntax
- Lifecycle Hooks
- Component Interaction
- Component Styles
- Angular Elements
- Dynamic Components

Data Binding

- What is Databinding?
- String Interpolation
- Property Binding
- Property Binding vs String Interpolation
- Event Binding
- Bindable Properties and Events
- Binding to Custom Properties
- Assigning an Alias to Custom Properties
- Binding to Custom Events
- Assigning an Alias to Custom Events
- Passing and Using Data with Event Binding
- Important: FormsModule is Required for Two-Way-Binding!
- Two-Way-Databinding
- Combining all Forms of Databinding

Filteration - Pipes

- Pipes

Directives

Structural Directives

- Using ngIf to Output Data Conditionally
- Enhancing ngIf with an Else Condition
- Styling Elements Dynamically with ngStyle
- Applying CSS Classes Dynamically with ngClass
- Outputting Lists with ngFor
- Getting the Index when using ngFor
- Understanding ngSwitch

Attribute Directives

- Building an attribute directive

Forms

- User Input
- Template-driven Forms
- Form Validation
- Reactive Forms
- Dynamic forms

Observables & RxJS

- Observables
- The RxJS Library
- Observables in Angular
- Practical Usage
- Compare to Other Techniques

View Encapsulation

- Understanding View Encapsulation
- Using Local References in Templates
- Getting Access to the Template & DOM with @ViewChild
- Projecting Content into Components with ng-content
- Getting Access to ng-content with @ContentChild

NgModules

- NgModules Introduction
- JS Modules vs NgModules
- Frequently Used NgModules
- Types of Feature Modules
- Entry Components
- Feature Modules
- Providers

Using Services & Dependency Injection

- Why would you Need Services?
- Creating a Logging Service
- Injecting the Logging Service into Components
- Creating a Data Service
- Understanding the Hierarchical Injector
- How many Instances of Service Should It Be?
- Injecting Services into Services
- Using Services for Cross-Component Communication
- What new in Services?
- Singleton Services

HttpClient

Routing & Navigation

Material Design

Testing

Server-Side Development

Database Management System

MongoDB

- What is MongoDB and its Versions?

- MongoDB CLI
- Administration & Query Browser
- Creating Databases and Collections
- Data Types:Varchar, Int, Char, Date and Time
- Deleting databases and collections
- Inserting, Retrieving, Updating and Deleting documents
- MongoDB User Accounts, Privileges and Access Control
- MongoDB documentation

Node.js

- Node.js - Get Started
- What is Node.js
- Node.js Process Model
- Setup Dev Environment
- Node.js Console
- Node.js Basics
- Node.js Modules
- Local Modules
- Export Module
- Node Package Manager
- Node.js Web Server
- Node.js File System
- Debugging Node.js
- Node Inspector
- Node.js EventEmitter
- Frameworks for Node.js: Express.js
- Express.js Web App
- Serving Static Resources
- Node.js Data Access: Access MongoDB
- Template Engine: Jade
- Project: Web Application and WebAPI

IONIC

Get Started

- Create Account
- Ionic CLI
- Build with ionic Creator

Development Environment for Ionic

- Introduction to Ionic
- Ionic Commands
- Project Structure
- Configure Android Development Environment for ionic
- Configure IOS Development Environment for ionic
- Creating the ionic application
- Serving the first application
- Ionic Development

CSS Components

- Colors
- Icons

- Buttons, check boxes, Inputs
- Other utilities

Mobile UIs

- Introduction to Mobile UIs
- Where mobile UI Framework Fit In
- First Page
- Second Page
- Toolbars
- Toast
- Dialogs
- List Views

Ionic APIs

- Camera API
- Connection API
- File API
- Notifications API
- Storage API
- Media API
- Storage: an easy way to store key/value pairs and JSON objects. Storage uses a variety of storage engines underneath, picking the best one available depending on the platform.

www.ict-trainings.com/curriculum/full-stack-development-leaflet.pdf