Radical Technologies
Project Based IT Training & Certification Centre

MASTER'S PROGRAM IN FULL STACK DOT NET

Integrated with International PG Diploma From UK Government Authorised Awarding Body – OTHM -United Kingdom. This master program is Endorsed by OTHM and You will achieve the below Certification after successful completion of the course.

INTERNATIONAL POST GRADUATE DIPLOMA CERTIFICATE IN SOFTWARE DEVELOPMENT & AUTOMATION ENGINEERING

What We cover From Basic To Advanced

Complete In-depth Knowledge

1. C#
2. MS SQL Server
3. ASP.NET MVC
4. WEB API
5. Angular
6. Html | CCS5 | Java Script | Bootstrap
7. Azure Cloud Developer

ASP .Net (MVC)

Web API

HTML

JS

ASP.NET

Bootstrap

Azure

fullstack

Job Oriented Scenarios

Multiple Job Oriented Responsibilities Practically Explained – Gain Knowledge of 5+ Year Experienced Full Stack Developer. 400+ Hours of content Job Guaranteed Master Program –

Duration: 1 Year
6 Months Learning + 6 Months
Projects, Assessments & Assignments

Major Features of the course

Major achievements of this program is that, This program is integrated with International PG Diploma Certificate Program in Software Development & Automation Engineering –Awarded by OTHM – UK , Ofqual Approved UK Qualification provider (https://www.gov.uk/government/organisations/ofqual)

Program Include Major and Minor Projects with all portions practically deeply explained, 200+ Assignments and 50 hours of Mock interview session. You will acquire a knowledge of 5+ year experienced in Full Stack Development after this course
This program is good for freshers and up to 3 – 4 years of industry experienced professional & Those who have career gap , Experienced people , who would like to upskill and change their domain .

Certificates Achieved

You will receive 4 certificates after successful completion.

1. RCITP after 6 months
2. Internship Certificate explaining Projects undergone
3. International PG Diploma certificate in Software Development & Automation Engineering Awarded from OTHM UK - After 6 Months of successful completion of the course . You have to undergo multiple assignments to achieve the International PG Diploma. After successful completion of assignment, you have to appear for 12 assessments, on topics covered during the course.

After fulfilling all the above criteria, International PG Diploma certificate in Software Development & Automation Engineering is awarded directly by OTHM
4. Fill any Career gap with our consulting companies who provides documentation on the projects completed

SALARY :- Avg. salary of a Full stack, Software Developer range from **2.8 Lakh and goes up to 15 Lakhs per annum** for 5+ year experienced developer .

Program Highlight

- 18000+ Students Empowered till now
- Online/Classroom/Self-Paced
- 6 months Intensive Training followed by 6 Month Assessments and Assignments
- Program include Four Certification Start Date – Every Month One Batch (Per Batch maximum 10 Students)
- 450+ Hiring Partners
- Certified from OTHM–UK Recognized by ofqual.gov.uk
- Minimum 5 Interview Call and until you get Job
- What you get After the Training – Gain Knowledge of At least 5+ Year Experienced Professional
- Designed for Freshers to Working Professionals
- 360 Degree Career Support
- One-on-One with Industry Mentors
- Dedicated Student Mentor
- 0 % Interest EMI – No Interest on EMI Paying – Approximately 5000 per month .
- Project Driven industry mentorship in Full Stack Development
- Dedicated career support – 450 + Hiring Partners
- 150+ core project & assignment and Job oriented activities
- 100 + Assignments
- 50 hours of Mock interview session
- Dedicated Trainers with ample of Industry Experience
- Project based IT Training and Certification programs .
- You Gain Knowledge of 5+ Full Stack Dot Net Developer
- Industry standard Projects And Assignments with Mentor Support
- Lab Exercises and Practical Guidance

Job Opportunities

- Full Stack Developer
- Automation Engineer
- Backend Developer
- Software Developer
- Front End Developer
- Sftware Engineer

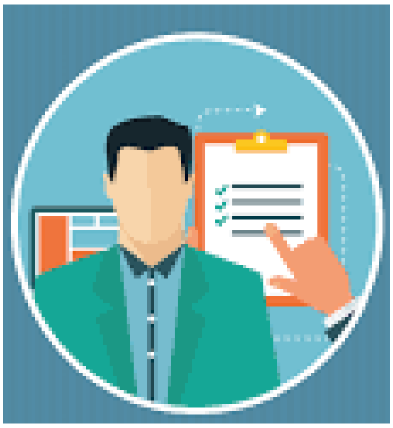
Our students placed in various Pay packages in MNC's Ranging from 5 Lakhs to 25 Lakhs . Experience Level from freshers to 18+ Year experienced candidates

Target audience?

Those who would like to go with programming technologies, Coding etc. Engineers, Freshers, Domain Experts, Software & IT Professionals, Those who having career gap & Interested in Programming Technologies and Coding etc, Already working professionals, who would like to upskill and try for another Job. Any One from Non-technical and Technical Background who wants to come into IT Engineering/Management Graduate or Post-graduate Fresher Students who want to make their career as Full Stack Developer, Software Developer, Automation Engineer,
Front End Developer, Backend Developer, Software Engineer Etc .

Minimum Eligibility

Any Bachelor's degree. Completed or not completed. No coding experience required. If you have any Educational Gap or any other career gap, you can do this program to boost up your career. The qualifications provided by UK Regulatory board is equal to Level 7 Masters Degree in UK.



How You Benefit From This Program

- International Post Graduate Diploma Certificate without quitting your job.
- Cutting-edge curriculum designed by industry experts .
- Trainers are from Real Time Industry
- Career transition with 30 to 100% salary hike

Frequently Asked Questions

1.Who Should do this course?
Freshers, Any Graduate, 2 to 4 Years Experienced up skilling enthusiasts. 3 rd Year Graduates who are going to attend campus Interviews. These courses designed in a way to be suitable for all branches of Engineering and all type of graduates – Science and
non-science Graduates. Option to customise the subject according to the interest of candidate is also available

2.How many Interview Calls You will provide?
Minimum 5 Interview Calls and Maximum , until you get the JOB

3.Do I get Refund , if I am not interested to continued?
Yes. you should inform the Administrator and send email to masters@radicaltechnologies.co.in to cancel your admission within 7 days of the enrolment. Once we allocate the trainer and further proceedings , we will not be able to refund.

4.How I will get Industry knowledge?
All our programs are developed by Industry person, who is having 10+ years real working knowledge. Under their mentorship, you will be able to gain at least 5+ year experience

5.Is this is 100% Job Guarantee program ? And how is the Salary

Structure?
Yes , it’s a 100% JOB Guaranteed program . Minimum 5 Interview calls and upto you get satisfied Job . Minimum salary Package of 5 Lakhs per annum for the right candidate .

6.Is there any limitation on number of Interview calls providing
No , You will get calls until you get placed . And we will provide unlimited interview calls after you get placed also

7.How many people got placed as part of this program?
We have placement record of over 6000+, Out of this 800+ are from Master Program.

Radical Technologies
Project Based IT Training & Certification Centre

C##

MS.NET Framework Introduction

- The .NET Framework - an Overview
- Framework Components
- Framework Versions
- Types of Applications which can be developed using MS.NET
- MS.NET Base Class Library
- MS.NET Namespaces
- MSIL / Metadata and PE files.
- The Common Language Runtime (CLR)
- Managed Code
- MS.NET Memory Management / Garbage Collection
- Common Type System (CTS)
- Common Language Specification (CLS)
- Types of JIT Compilers
- Security Manager

VS.NET and Entry Point Method –Main

- Introduction to Project and Solution in Studio
- Entry point method - Main.
- Compiling and Building Projects
- Using Command Line Arguments
- Importance of Exit code of an application
- Different valid forms of Main
- Compiling a C# program using commandline utility CSC.EXE

C # Language Syntax

- Why Datatypes
- Global, Stack and Heap Memory
- Common Type System
- Reference Type and Value Type
- Datatypes &amp; Variables Declaration
- Implicit and Explicit Casting
- Checked and Unchecked Blocks – Overflow Checks
- Casting between other datatypes
- Boxing and Unboxing
- Enum and Constant
- Operators
- Control Statements
- Working with Arrays
- Working with Methods
- Pass by value and by reference and out parameters
- Misc...

OOPs-Concept

- Learning about Class, Object, Component, Encapsulation, Inheritance,

- Polymorphism & Object Creation and Instantiation.

OOPs-Programming Encapsulation

- Understanding Encapsulation Concept through an example. OOPs-
- Inheritance
- Introduction to Inheritance
- Constructor &Inheritance
- Type Casting of Reference Types
- Static and Dynamic Binding
- Abstract Class

OOPs-Interface & Polymorphism

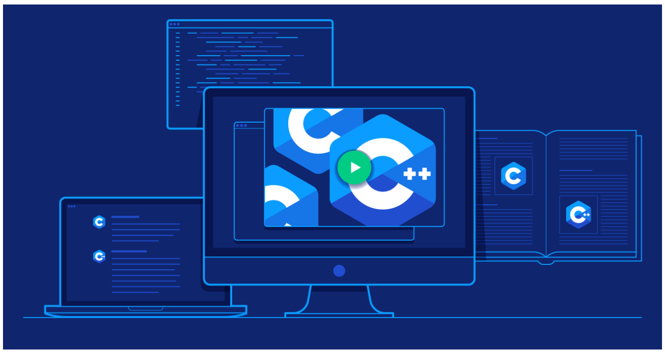
- What is Polymorphism
- Overview of Interface
- Interface with examples
- Types of Inheritance

Collections and Generics

- Introducing Collections .
- Benefits of Collection Classes .
- Understanding and using commonly used collections.
- Generics
- Advantages of Generics .
- How Generics Work at Runtime .
- Constraint on Type Parameters .
- Generic Methods
- Generic Collections
- Selecting a Collection Class

Assemblies and GAC

- What is a DLL and how is it different from EXE
- Types of DLL
- What is an Assembly Assemblies
- How to build a ClassLibrary?
- How to use a ClassLibrary in another Application?
- What is Namespace?
- Internal Access Specifier
- Types of Assemblies
- Global Assembly Cache



Exception Handling

- Defining Exception
- Understandings try and catch keywords
- Using “finally” block
- “using” statement
- Throwing exceptions
- Creating User defined/ Custom Exception class.

IO Streams

- What is a streams?
- Types of Stream

- Standard I/O Streams
- Console
- Handling text in files
- Dealing with Binary files
- Serialization / Deserialization

Unsafe Code
Reflection and Attributes

- What is Reflection?
- Using Reflection to read type information
- Attributes.
- Pre-defined Attributes
- Custom Attributes.
- Using Reflection to read custom attributes

More on Classes

- Operator Overloading
- Partial Classes and Methods
- Anonymous Types
- Extension Methods
- Tuples
- Caller Method Information
- Configuration File

Developing GUI Application Using WINFORMS

- Basic Controls
- Panel & Layouts
- Drawing and GDI Devices
- MenuStrip, ToolbarStrip and ContextMenuStrip
- Model and Modeless Dialog boxes
- Mutiple Document Interface(MDI)
- Form Inheritance
- Building Login Form
- Working with Resource Files and Setting
- Notify Icon Controls
- Using Components like Timer, FileSystemWatcher, Process, BackgroundWorker
- Drag and Drop
- Working with Advanced Controls like TreeView and ListView

Database Programming Using ADO.NET

- Prerequisite - Knowledge of SQL Queries
- Introduction and Evolution of ADO.NET
- Understanding the Role of Managed Provider and ADO.NET Objects
- installing Required Software - Sql Server and Management studio
- Connecting to Database and Connection Pooling
- Performing Insert, Update and Delete Operations
- Fetching Data from database - Executing Select Statements
- How to implement Login

facility with database

- Use of Multiple Active Result Sets
- Parameterized Prepared Statements
- Inserting Image into Database table
- Executing Stored Procedure
- Using Transaction
- Asynchronous Execution of Queries
- Writing Provider Independent Code
- Writing Common Code for Execution of Stored Procedures
- Quick Overview of all ADO. NET objects

Managing Data using DataSet

- DataSet and its Object Model
- Filling DataSet using DataAdapter
- Binding DataSet to DataGridView
- Updating changes to database using DataAdapter
- Using SqlCommandBuilder
- Managing DataTable Programmatically
- DataAdapter events
- Handling concurrency issue
- Working with DataViews
- Constraints in DataTable
- Using DataRelations object
- Creating DataSet/DataTable dynamically
- Working with Typed DataSet
- Summary and Important Classes and their properties and methods

- N-Tier Layered Architecture Application
- Understanding Tier and Layer
- Dividing Application into multiple layers
- Developing an application using Layered Architecture

XML

- a.Creating Table and Stored Procedure
- b.Creating Data Class
c.Creating DAL Class
d.Creating BO Class
- e.Creating Form and handlingevents
- f.Creating Dialog Box for Add and Edit OPerations.
- Introduction
- Well Formed and Valid XML Document
- Structure of XML Document
- XML DOM Parser

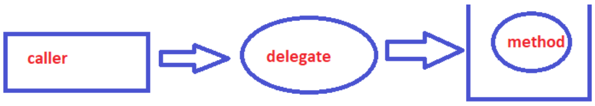
- XPath Specification
- XML and DataSet
- XMLDataDocument for reading from DataSet
- XMLTextWriter & XML Text Reader
- XPathDocument & XPath Navigator

Windows Services

- Introduction to Windows Service
- Windows Service Project Template
- Developing Windows Services
- Installing, Deploying and Launching Windows Service
- Developing a Service Controller Application Handling Custom Commands in Windows
- Services

Delegates & Events

- Introduction to Delegates
- Creating a Chat Application Using Delegates
- Events Declaration, Raising and Handling
- Anonymous Methods



User Control and Custom Control Multithreading

- Threading Overview
- Scheduling
- Thread States
- Programming Threads
- Methods of Thread Class
- Thread Pool
- Thread Synchronization Monitor
- Mutex
- Semaphore
- Events
- Parallel Programming using Task Parallel Library
- Asynchronous Programming using async and wait
- keywords

Packaging and Deployment

- File System Editor
- Registry Editor
- File Types Editor
- User Interface Editor
- Custom Actions
- Launch Condition Editor
- Creating Uninstall Shortcut

Debugging and Diagnostics

- What is Debugging?
- Build Configuration(Debug and Release)
- List of Debugging Windows
- Break Point Hit Count and Condition
- Debugging Exception □
- What is Diagnostics?
- Debug and Trace Classes
- Types of Listeners
- Boolean and Trace Switch

MS SQL Server

Introduction To DBMS

- File Management System And Its Drawbacks
- Database Management System (DBMS) and Data Models
- Physical Data Models
- Logical Data Models
- Hierarchical Data Model (HDBMS)
- Network Data Model (NDBMS)
- Relational Data Model (RDBMS)
- Object Data Model (ODBMS)
- Object Relational Data Model (ORDBMS)
- Conceptual Data Models
- Entity – Relationship (E-R) Model

Introduction To SQL Server

- Advantages and Drawbacks Of SQL Server Compared To Oracle And DB2
- Connecting To Server
- Server Type
- Server Name
- Authentication Modes
- Sql Server Authentication Mode
- Windows Authentication Mode
- Login and Password
- Sql Server Management Studio and Tools In Management Studio
- Object Explorer
- Object Explorer Details
- Query Editor

Introduction To TSQL

- History and Features of TSQL
- Types Of TSQL Commands
- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- Database
- Creating Database
- Altering Database
- Deleting Database
- Constrains
- Procedural Integrity Constraints
- Declarative Integrity Constraints
- Not Null, Unique, Default and Check constraints
- Primary Key and Referential Integrity or foreign key constraints
- Data Types In TSQL
- Table

- Creating Table
- Altering Table
- Deleting Table

Data Manipulation Language

- Insert
- Identity
- Creating A Table From Another Table
- Inserting Rows From One Table To Another
- Update
- Computed Columns
- Delete
- Truncate
- Differences Between Delete and Truncate

Data Query Language (DQL)

- Select
- Where clause
- Order By Clause
- Distinct Keyword Isnull() function
- Column aliases
- Predicates
- Between … And
- In
- Like
- Is Null

Built In Functions

- Scalar Functions
- Numeric Functions
- Character Functions
- Conversion Functions
- Date Functions
- Aggregate Functions
- Convenient Aggregate Functions
- Statistical Aggregate Functions
- Group By and Having Clauses
- Super Aggregates Over(partition by …) Clause
- Ranking Functions
- Common Table Expressions (CTE)

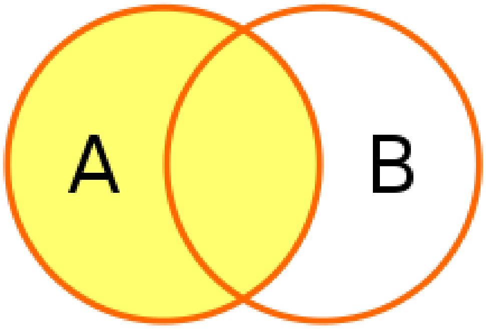
Top n Clause

Set Operators

- Union
- Intersect
- Except

Joins

- Inner Join
- Equi Join
- Natural Join
- Non-Equi Join
- Self Join
- Outer Join
- Left Outer Join
- Right Outer Join
- Full Outer Join
- Cross Join



Sub Queries

- Single Row Sub Queries
- Multi Row Sub Queries
- Any or Some
- ALL
- Nested Sub Queries
- Co-Related Sub Queries
- Exists and Not Exists

Indexes

- Clustered Index
- NonClustered Index
- Create , Alter and Drop Indexes
- Using Indexes

Security

- Login Creation
- SQL Server Authenticated Login
- Windows Authenticated Login
- User Creation
- Granting Permissions
- Revoking Permissions
- Roles

Views

- Purpose Of Views
- Creating , Altering and Dropping Indexes
- Simple and Complex Views
- Encryption and Schema Binding Options in creating views

Transaction Management

- Introduction
- Begin Transaction
- Commit Transaction
- Rollback Transaction
- Save Transaction□ Role Of Log File In Transaction Management
- Implicit Transactions

TSQL Programming

- Drawbacks Of TSQL that leads to TSQL Programming
- Introduction To TSQL Programming

- Control statements In TSQL Programming
- Conditional Control Statements
- If
- Case
- Looping Control Statements
- While

Cursors

- Working With Cursors
- Types Of Cursors
- Forward_Only and Scroll Cursors

- Static, Dynamic and Keyset Cursors
- Local and Global Cursors

Stored Sub Programs

- **Advantages Of Stored Sub Programs compared to Independent SQL Statements**
- **Stored Procedures**
-
- Creating , Altering and Dropping
- Optional Parameters
- Input and Output Parameters
- Permissions on Stored Procedures
-
- **User Defined Functions**
-
- Creating, Altering and Dropping
- Types Of User Defined Functions
-
- **Scalar Functions**
- **Table Valued Functions**
-
- Inline Table Valued Functions
- Multi Statement Table Valued Functions
- Permissions On User Defined Functions
-
- **Triggers**
-
- Purpose of Triggers
- Differences Between Stored Procedures and User Defined Functions and Triggers
- Creating, Altering and Dropping Triggers
- Magic Tables
- Instead Of Triggers
-
- **Exception Handling**
- Implementing Exception Handling
- Adding and removing User Defined Error Messages To And From SQL Server Error Messages
- List
- Raising Exceptions Manual

CLR Integration

- What is CLR Integration and The Steps For Implementing It
- A Simple Example With CLR Integration
- Working With XML Data Type
- Backup and Restore Of Database
- Attach and Detach of Database

ASP .NET MVC

Intro to ASP.NET MVC 6

- Introduction to MVC 6
- NET Web Forms (vs) ASP. NET MVC
- Advantages and disadvantages of each
- Determining when-to-use which one
- List of Versions of ASP.NET MVC
- Differences between versions of ASP.NET MVC
- Overview of installing ASP. NET MVC in various versions of Visual Studio
- MVC Architecture
- What is MVC?
- Architecture Explanation
- What is Controller and action method
- What is View
- What is Model
- Request Flow in ASP.NET MVC

Getting started with MVC Core

- New Project in Visual Studio
- Overview of Folders and files of MVC project

Controllers

- Introduction to Controllers
- Creating Controllers and Actions
- Calling action methods thru the browser
- Returning from action methods
- Parameters in Action methods
- ActionLink
- URL Routing
- Overview
- The need of URL Routing
- Parameters in URL
- Default Parameter Values
- Parameters with Constraints
- Literals in URL

Views (Razor)

- Introduction to Views (Razor)
- ActionResult and ViewResult
- Returning a view
- Creating a Simple Razor View
- Intermingling Code and Markup in Razor Views
- View Bag / View Data / Temp Data
- Shared Views
- ASPX View Engine (vs) Razor

Microsoft .NET MVC

LayoutViews

- Introduction to LayoutViews
- The need of layout views
- cshtml
- Creating custom layout views
- Layout Views with Sections
- RenderSection()
- cshtml
- Overriding the _ViewStart. cshtml

Partial Views

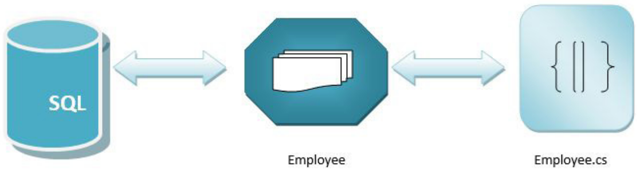
- Partial Views
- RenderPartial()

Models

- Introduction to Models
- Need of models
- Creating models using ‘CodeFirst approach’

Entity Framework in MVC

- Introduction to Entity Framework
- Need of Entity Framework Creating DbContext and DbSet
- Configuring connection string



Scaffold Templates in MVC

- Introduction to scaffold Templates in MVC
- Need of Scaffolding
- □ Creating controllers and views using scaffold
- Strongly typed views
- Understanding Index, Details, Create, Edit, Delete action methods and views

HTML Helpers

- Introduction to HTML helpers
- DisplayNameFor( )
- DisplayFor( )
- BeginForm( )
- LabelFor()
- EditorFor( )
- ValidationMessageFor( )
- RadioButtonFor( )
- DropDownListFor( )
- ListBoxFor( )
- CheckBoxFor( )
- AntiForgeryToken()
- Code First Migrations

Action Filters

- Introduction to action filters
- [HttpPost]
- [HttpGet]
- [ValidateAntiForgeryToken]
- [OutputCache]
- [HandleError]
- [NonAction]
- [ActionName]

Bundles and Minification

- Introduction to Bundles and minification
- Understanding BundleConfig.cs
- ScriptBundle (vs) StyleBundle
- Importing a ScriptBundle and StyleBundle
- Creating and consuming custom bundles

Validations

- Introduction to Validations
- Model level validations (vs) View level validations
- Importing jQuery Validation Plug in
- [Required]
- [RegularExpression]
- [Range]
- [StringLength]
- [Compare]
- [Remote]
- IsValid

Data Annotations

- [Display]
- [Datatype]
- [DisplayFormat]
- [ScaffoldColumn]

jQuery in MVC

- Introduction to jQuery in MVC
- Need of jQuery in MVC
- Importing/updating jquery plug in
- Demos

jQuery AJAX in MVC

- Need of jQuery-AJAX in MVC
- Demos on jQuery-AJAX (with json) in MVC with database handling
- Intro to jQGrid
-

• Web API

-
- Introduction to Web API
- WCF REST (vs) Web API
- Understanding HTTP methods(GET, POST, PUT, DELETE)
- Creating API controller
- API configuration
- Calling Web API from jQuery
- Demo: CRUD operations using jQuery AJAX with Web API

Security in MVC

- Forms based security with [Authorise]
- Windows based security
- What’s new in MVC Core
- Web API 2
- Overview of Facebook Application
- SIGNAL R 3
- Web API 3
- OWIN Abstraction
- Inbuilt Support for Dependency Injection

- SIGNAL R 3
- Web API 3

Repository Layer and Service Layer

- Introduction to Repository Layer and Service Layer
- MVC Architecture with both
- layers
- Demos

Dependency Injection

- Problems in accessing repository layer / service layer directly in controller
- What is Dependency Injection
- Installing ‘Unity’ plug in
- Configuring Unity
- Construction injection and property injection using interface

Database – First approach in MVC

- Introduction to Database-First approach in MVC
- Demos on ‘Database-First’ approach

Areas in MVC

- Introduction to areas in MVC
- Demos on areas

Mobile Web Apps using MVC

- Introduction to jQuery Mobile
- Mobile Application’ project template
- Overview of Customizing jQuery Mobile
- Deployment
- Configuring IIS
- Publishing MVC web applications to local IIS

Angular 9 Integration

- What is Angular9
- Advantages
- Getting started
- MVVM and MVC Architecture
- What is Model, View, ViewModel
- Creating view model
- data-bind
- Observable properties
- Subscribe
- Working with Directives, Services, Filters, Functions

Single Page Application (SPA)

- What is SPA
- Need of SPA
- Getting started
- Angular JS with API
- Reading and posting data

HTML

Web Programming Introduction

- Architecture of a website
- Different technologies in making the website
- Web Development Introductio

HTML-Introduction

- History of HTML
- What you need to do to get going and make your first HTML page
- What are HTML Tags and Attributes?
- HTML Tag vs. Element
- HTML Attributes

HTML-Basic Formatting Tags

- HTML Basic Tags
- HTML Formatting Tags
- HTML Color Coding

HTML-Grouping Using Div Span

- Div and Span Tags for Grouping

Module 5: HTML-Lists

- Unordered Lists
- Ordered Lists
- Definition list

HTML-Images

- Image and Image Mapping

HTML-Hyperlink

- URL – Uniform Resource Locator
- URL Encoding

HTML-Table

- < table >
- < th >
- < tr >
- < td >
- < caption >
- < thead >
- < tbody >
- < tfoot >
- < colgroup >
- < col >

HTML-Iframe

- Using Iframe as the Target

HTML-Form

- < input >
- < textarea >
- < button >
- < select >

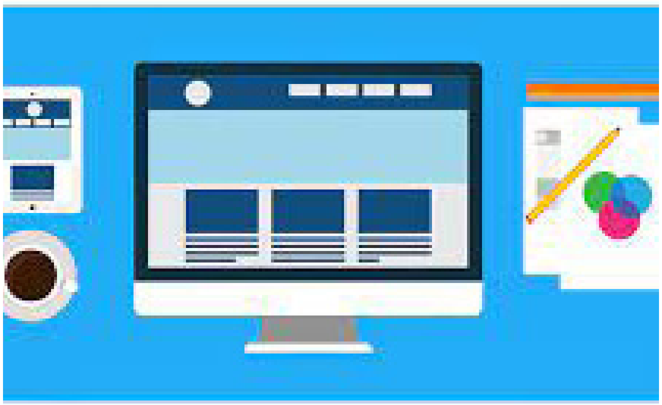
- < label >

: HTML-Headers

- Title
- Base
- Link
- Styles
- Script
- Meta

HTML-Miscellaneous

- HTML Meta Tag
- XHTML
- HTML Deprecated Tags & Attributes



CSS3

CSS3-Introduction

- Benefits of CSS
- CSS Versions History
- CSS Syntax
- External Style Sheet using < link >
- Multiple Style Sheets
- Value Lengths and Percentages

CSS3-Syntax

- CSS Syntax
- single Style Sheets
- Multiple Style Sheets
- Value Lengths and Percentages

CSS3-Selectors

- ID Selectors
- Class Selectors
- Grouping Selectors
- Universal Selector
- Descendant /Child Selectors
- Attribute Selectors
- CSS – Pseudo Classes

CSS3-Color Background Cursor

- background-image
- background-repeat
- background-position
- CSS Cursor

CSS3-Text Fonts

- color
- background-color
- text-decoration
- text-align
- vertical-align
- text-indent
- text-transform
- white space
- letter-spacing
- word-spacing
- line-height
- font-family

- font-size
- font-style
- font-variant
- font-weight

CSS3-Lists Tables

- list-style-type
- list-style-position
- list-style-image
- list-style
- CSS Tables
- border
- width & height
- text-align
- vertical-align
- padding
- color

CSS3-Box Model

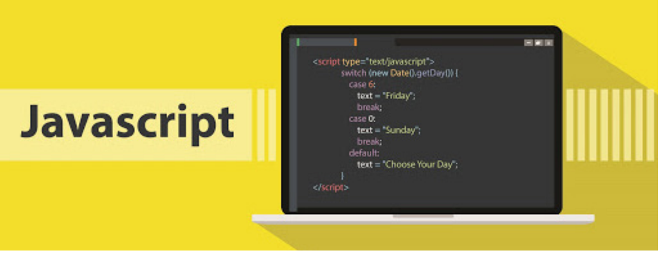
- Borders & Outline
- Margin & Padding
- Height and width
- CSS Dimensions

CSS3-Display Positioning

- CSS Visibility
- CSS Display
- CSS Scrollbars
- CSS Positioning
- Static Positioning
- Fixed Positioning
- Relative Positioning
- Absolute Positioning
- CSS Layers with Z-Index

CSS Floats

- The float Property
- The clear Property
- The clear fix Hack



```
<script type="text/javascript">
      switch (new Date().getDay()) {
        case 6:
          text = "Friday";
          break;
        case 0:
          text = "Sunday";
          break;
        default:
          text = "Choose Your Day";
      }
</script>
```

JAVASCRIPT

- Introduction
- Setup
- What is Javascript
- Course Exercise Files
- Practice App & Display Functions

Assignments 1:

- Using a Code Editor
- Variables
- Arrays

Assignments 2:

- Practice App

Section 2: Control Structures, Functions, Math & Events

- If Statements
- Loops
- Math
- Functions
- Even Handlers Part:1 (Example:
- Event Handlers Part:2

Assignments:03

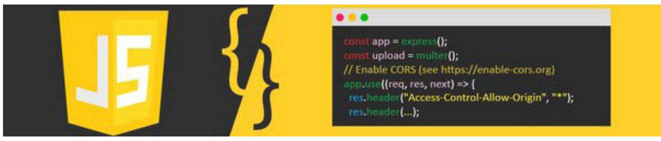
- Practice App – If Statements and Loops
- Practice App – Math & Functions
- Practice App – Events

Section 3: Advance Scenario #01

- Printing Arrays
- Text Output Function with a Loop
- Built-in Functions (Displaying the Time In Javascript)
- Math Built-in Functions

Section 4: Advance Scenario #02

- Assigning, Adding and Removing Elements from Arrays
- Inserting and Extracting From Arrays more functions
- ForloopArray Checking with Challenge Part #1
- ForloopArray Checking with Solution Part #2
- Setting Up Flags and Using Booleans
- Where to go next?
- Quick Syntax Reference/ Resource File
- Conclusion



```
const app = express();
const upload = multer();
// Enable CORS (see https://enable-cors.org)
app.use((req, res, next) => {
  res.header("Access-Control-Allow-Origin", "*");
  res.header(...);
```

BOOTSTRAP

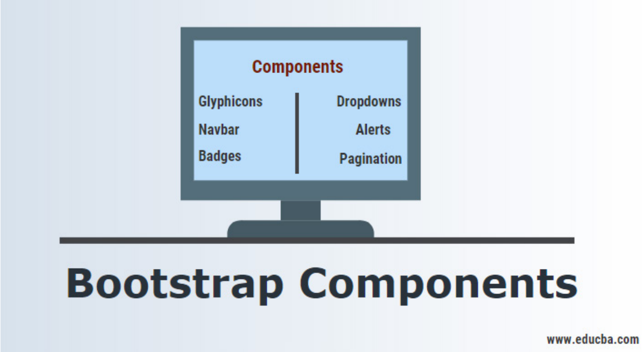
Introduction to Bootstrap

- What is Bootstrap Framework
- Why Bootstrap
- History of Bootstrap
- Advantages of Bootstrap Framework
- What is Responsive web page
- How to remove Responsiveness
- Major Features of Bootstrap
- What is Mobile-First Strategy
- Setting up Environment
- How to apply Bootstrap to Applications

Bootstrap Grid

- What is Bootstrap Grid
- How to apply Bootstrap Grid
- What is Container
- What is Offset Column
- How to Reordering Columns
- Advantages of Bootstrap Grid
- How to Display responsive Images
- How to change class properties

- How to use readymade themes
- How to customize Bootstrap&#39;s components, Less variables, and jQuery plug-in.
- What is Bootstrap Typography
- How to use Typography
- What is Bootstrap Tables
- What is Bootstrap Form Layout
- What is Bootstrap Button
- How display images in different styles like Circle shape etc
- How to display text like muted and warning etc
- What is Carets Classes
- How to hide or show the text in Bootstrap



Bootstrap Components

- What is Bootstrap Components
- Why Bootstrap Components
- Advantages of Bootstrap Components
- What are the different types of Bootstrap Components
- What is Glyphicons Component
- How to use Glyphicons Component
- What is Bootstrap Dropdown Menu Component
- What is Button Groups and Button Toolbar
- How to use Button Groups and Button Toolbar
- What are different Input Groups Components
- What is Navigation Pills & Tabs Components
- How to use Navigation Pills and Tabs Components
- What is Navbar Component
- How to build a Responsive Navbar
- How to Add Forms and other controls to Navbar
- How to Fix the position of navbar
- What is Breadcrumb Component
- What is Pagination Component
- How to apply Pagination in Application
- What is Labels / Badge Components
- What is Jumbotron / Page Header Components
- What is Thumbnail Component
- What is Alerts & Dismissible Alerts
- How to Create Progress Bar
- What is Media Objects Component

- Why Media Objects Component
- How to use Media Objects Component
- What is Bootstrap List Group Component
- What is Bootstrap Panel Component

Module 4:- Bootstrap Plug-Ins

- What is Bootstrap Plug-Ins
- Why Bootstrap Plug-Ins
- How to use Plug-Ins
- What is Transition Plug-in
- What Modal Dialog Box
- What are the different Properties, Methods and Events of Model Dialog Box
- What is Scrollspy Plug-In
- What is Tab Plug-in
- How to use Tab Plug-in
- What is Drop Down Plug-in
- What is Tooltip Plug-in
- How to use Button Plug-in
- What are the different methods and events of Tooltip Plug-in
- What is Popover Plug-in
- What is alert and Button Plug-ins
- What is Collapse Plug-in
- What are different types of Properties, Methods and Events of Collapse Plug-in
- What is Carousel Plug-in
- What is Affix Plug-in

ASP .NET WEB API

Introduction to Web API

- Introduction to SOA
- Introduction to REST
- Introduction to ASP.NET Web API
- WCF vs. Web API
- Building Web API
- Testing API Using Postman

ASP.NET Web API Fundamentals

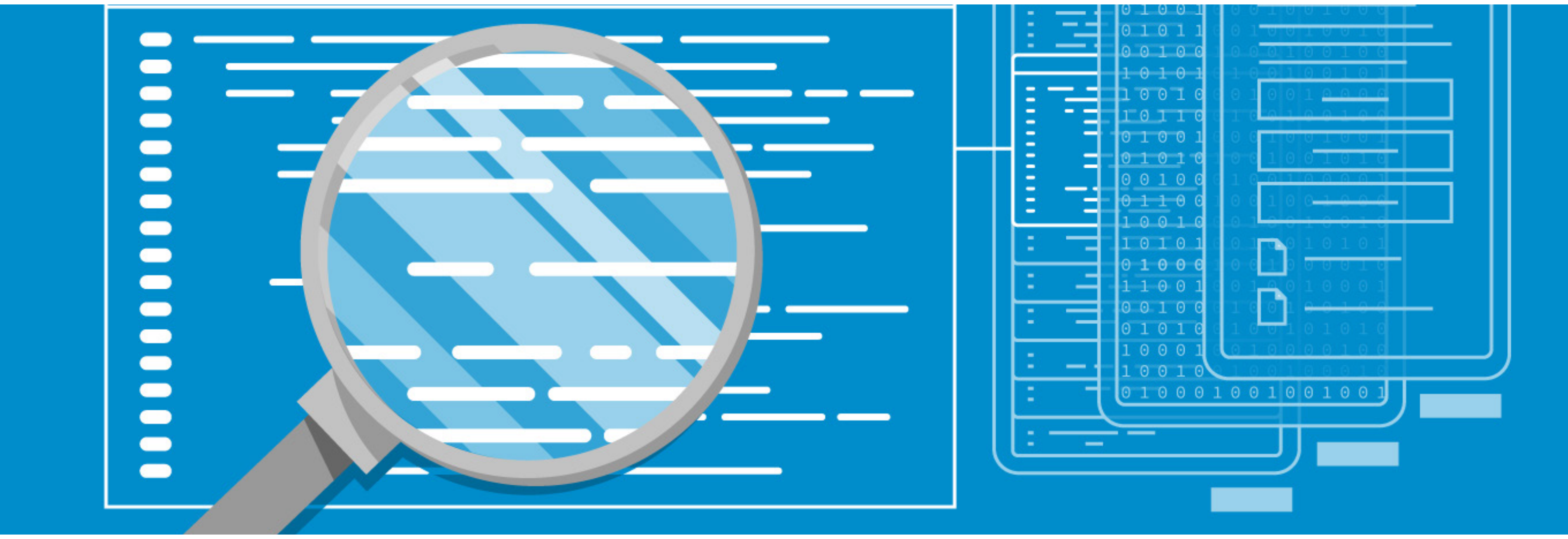
- Routing - Convention and Attribute Routing
- Route Constraints
- Action Result
- Content Negotiation
- Media Type Formatters
- Model Binding
- Model Validation
- Handling Validation Errors

Building RESTful Services with ASP.NET Web API

- Introduction to ASP.NET Web API
- Building ASP.NET Web API
- ASP.NET Web API CRUD Operations
- Testing Web API Using Postman
- Consuming Web API in ASP. NET MVC
- Performing CRUD Operations using ASP.NET MVC

ASP.NET Web API Advanced Concepts

- ASP.NET Web API Pipeline
- Web API Filters
- Creating Custom Filters
- Exception Handling
- Dependency Injection
- Web API Versioning
- Web API Hosting



ANGULAR

Web development history

[ASP.net / PHP (all synchronous operations)] => [Introduction of Ajax] => [MVC.net websites (async operations)] => [Introduction of JSON web services] => [introduction of UI frameworks]

Service side VS client side programming

Available UI frameworks

Why Angular?

Environment setup

- Choice of IDE & installing it (visual studio code, visual studio, eclipse or any text editor)
- Installing node and npm
- Install angular CLI
- What is angular CLI? Installing angular CLI. Comprehensive list of CLI command.

Creating first Angular project

- Creating first Angular application with angular CLI
- Building and running angular Application
- Angular standards and deep dive into project structure (Significance of each folder & file) (Revise this after few lectures)
- How to run End to end test cases and unit testing

Angular project flow

- Flow from boot strapping app to running into browser
- Making small and easy changes to first app
- How to debug angular application

Prerequisite of learning Angular

- Introduction to HTML
- Introduction to bootstrap CSS (optional)
- Introduction to Typescript (Necessity of Typescripts in enterprise applications) – Create classes use it in first app.

Angular Architecture

- Architecture overview
- Intro to Modules
- Intro to Component
- Intro to Services and DI (will cover in next lecture)

Components in details

- Component (selector, inline template, template url, styles, provider)
- Template syntax (various ways of data binding)

A. Interpolation

- Template expression
- Expression context

B.Property binding

- Difference between element attribute and property
- Element property binding
- attribute, class, style binding

C.Event binding

D.Two way binding

- Using various html elements in angular e.g. textbox, button, dropdown, checkbox, radio button etc
- Child components – Event emitter and custom property

Directives

- Attribute directives (ngClass, ngStyle,ngModel)
- Structural directives (ngIf, ngFor, ng Switch)
- Creating custom directive

Pipes

- using inbuilt pipes e.g. DatePipe, UpperCasePipe, LowerCasePipe, CurrencyPipe, and PercentPipe etc
- Parameterized pipe
- Creating custom pipe

Animations

- Setup (npm install @ angular/animations@latest –save)
- Example

Forms

- User input
- Form validations

15.NgModules in details

- NgModules introduction (declaration, imports, exports, providers, bootstrap)
- js modules vs NgModules
- Frequently used modules
- Roots module vs Feature module

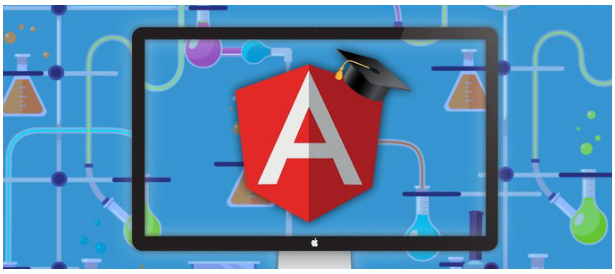
- Creating feature module and use it in app (which requires routing)

Routing and navigation

- Configuring router for app
- Significance of <base href=”/”>
- Router outlet
- Router link
- Passing param using router
- Router events
- #Advance topics
- Routing components from other modules
- Lazy loading

Http client

- Setup
- Using Observable and RxJS
- Calling rest services (GET, PUT, DELETE, PUT).
- Error handling
- Calling aws rest service endpoint (or mocked rest service endpoint using www.mockable.io) #Advance topics
- Debouncing requests e.g. search widget
- Request/response interceptors



Angular 8 Course Content

Introduction of Angular 8

- Angular 8 Introduction
- History and versions of Angular 8
- Architecture of Angular 8
- How to install Angular 8 and set-up it.
- Creating our first Angular 8 app
- Angular 8 app loading

Difference between

- Angular And react
-
- Angular vs react

Angular 8 Advantages Disadvantage

- Advantage and Disadvantage of Angular 8

Angular 8 file structure

- Angular 8 file structure

Angular 8 components

- Components of Angular 8

Angular 8 CLI Commands

- All CLI commands of Angular

Angular 8 with Bootstrap

- How to install bootstrap for Angular 8
- Libraries of Angular 8

Angular 8 Routing

- Routing in Angular 8

Angular 8 directives

- Angular 8 Directives
- Angular 8 ngIf directive
- Angular 8 ngFor directive
- Angular 8 ngSwitch directive
- Angular 8 ngClass directive
- Angular 8 ngStyle directive

Angular 8 pipes

- Angular 8 Pipes

Angular 8 databinding

- Angular 8 Data binding
- Angular 8 Event binding
- Angular 8 Property binding
- Two-way data binding in Angular 8

String Interpolation In Angular 8

- Angular 8 String interpolation

Angular 8 forms

- Angular 8 Forms
- Data flow of forms in Angular 8
- Creating forms in Angular 8
- Testing and validation of forms in Angular 8

Error fixing in Angular 8

- Error fixing in Angular 8

Dependency injection and services in Angular 8

- Dependency injection services in Angular 8

Angular 8 Animations

- Angular 8 Animations

Dynamic components in Angular 8

- Dynamic components in Angular 8

Angular 8 Module

- Angular 8 Module

- Deploying an angular 8 app

Introduction of unit testing in angular 8

- unit testing in angular 8

Observables in angular 8

- Observables in angular 8

Angular 8 universal

- Angular 8 universal

Angular 8 Changes and new features

- New features and changes in Angular 8

Conclusion

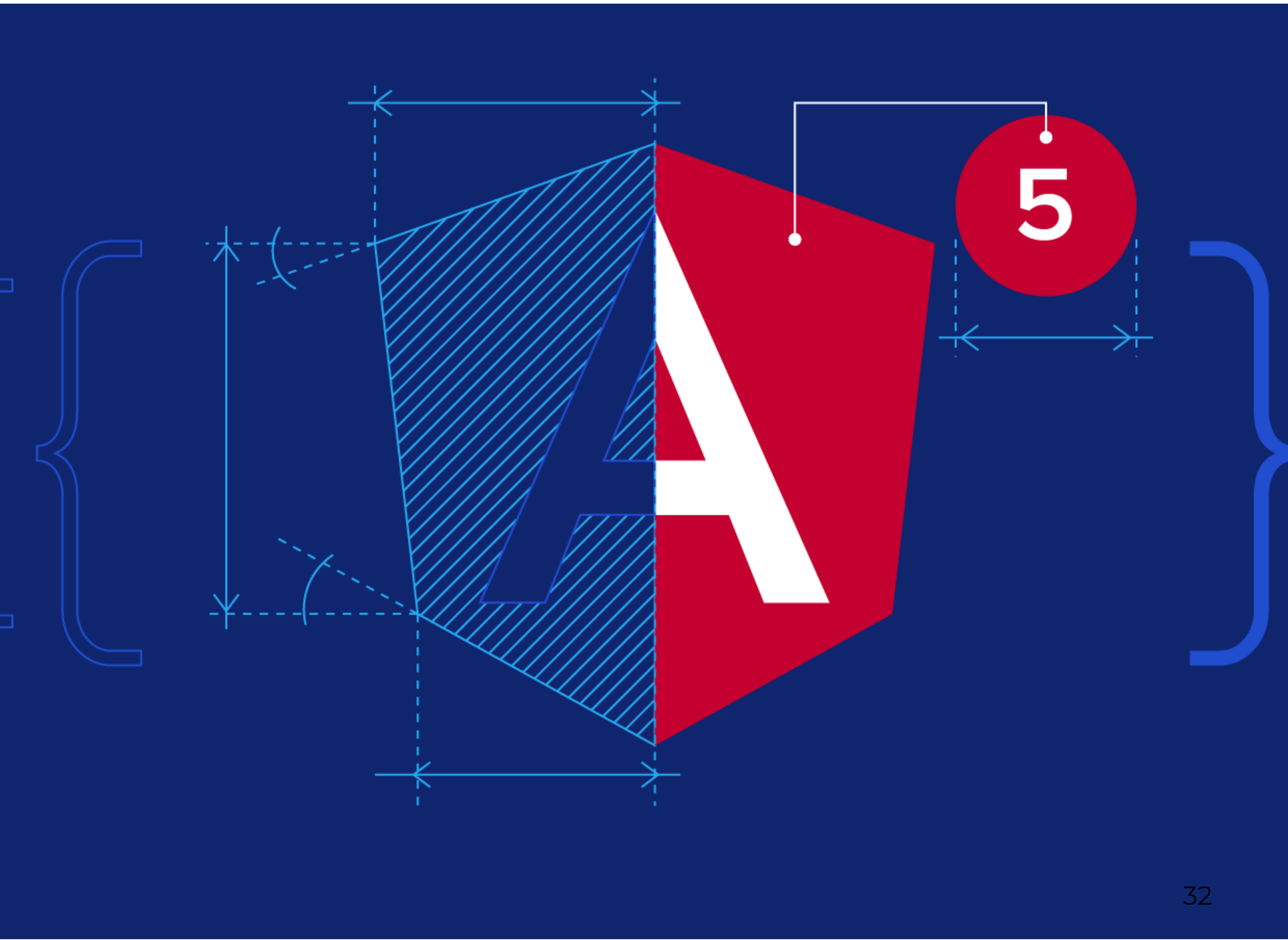
- Angular 8 Conclusion

Creating application – User management system

Deploying app

Interview questions will be discussed with every topic

Git repository setup(optional)



MICROSOFT AZURE DEVELOPER

Cloud Computing

- Overview of Cloud Computing
- Architecture of Cloud Computing
- Types of Cloud
- Benefits of Cloud

Windows Azure

- Azure as PaaS (Platform as a Service)
- Azure as IaaS (Infrastructure as a Service)

Azure Management Portal

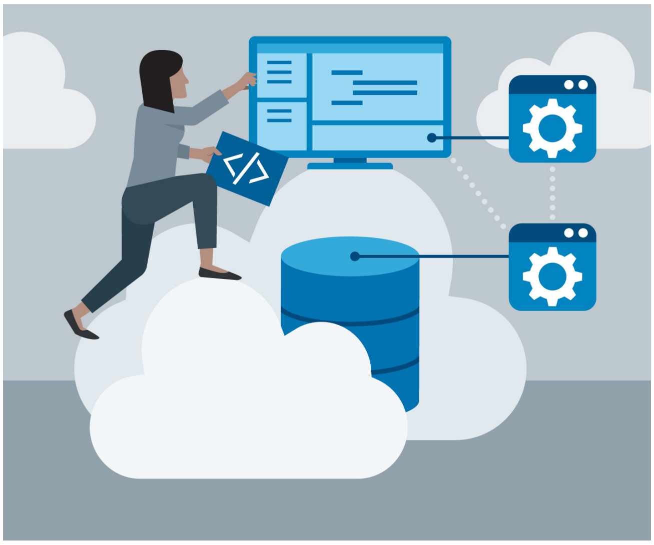
- Create a New Application.
- Check Credit and Subscriptions
- Add a New Subscription

Application Deployment

- Create a Website in Azure Management Portal
- Create a Deployment Package
- Deploy Website using Deployment Package
- Deploying Azure Website from Visual Studio
- Monitoring the Website
- Staged Publishing

Azure –Components

- Compute / Execution Models
- Data Management
- Networking
- Big Data and Big Compute
- Messaging
- Caching
- Identity and Access
- Mobile Service
- Backup
- Media
- Commerce
- Software Development Kit (SDK)



Compute Module

- Create a Web App
- Create a Virtual Machine
- Creating a Mobile Service
- Creating Batch Service

Scalability

Enable Diagnostics

Fabric Controller

Storage

- Creating Azure Storage Account
- Storage Account Endpoints
- Generating an Access Key
- Managing Data to Azure Storage

Blobs

- Create a Container
- Upload a Blob using PowerShell
- Download a Blob
- Manage Blobs using Azure Storage Explorer

Queues

- Managing Queues using PowerShell
- Managing Queues using Azure Storage Explorer

Tables

- How to Manage Tables Using PowerShell
- How to Manage Table using Azure Storage Explorer

CDN

- Create a CDN
- Create CDN for Custom Origin Links
- Manage CDN
- Map a Custom Domain Name

Security

- Creating an Active Directory
- Mapping a Custom Domain
- Creating Users
- Integrating with Azure Active Directory
- Integrating On-Premise Active Directory
- Reports

Traffic Manager

- Create Traffic Manager
- Create Endpoints to be monitored via Traffic Manager
- Configure the Policy
- Setting up Alert Rules

Datacenters

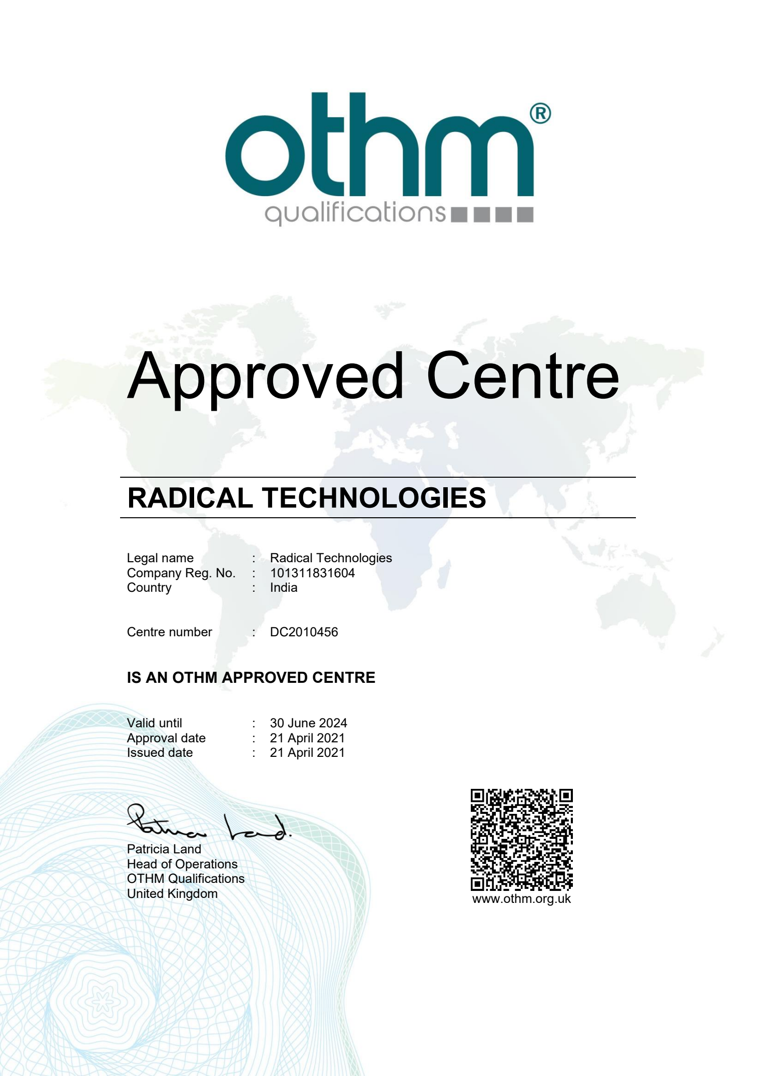
- How to Choose the Right Data Center for Your Application

Backup & Recovery

- Create Backup Vault
- Schedule a Backup

Data Import and Export Job

- Data Export Job
- Create an Export Job
- Create Import job



othm qualifications

Approved Centre

RADICAL TECHNOLOGIES

Legal name	:	Radical Technologies
Company Reg. No.	:	101311831604
Country	:	India
Centre number	:	DC2010456

IS AN OTHM APPROVED CENTRE

Valid until	:	30 June 2024
Approval date	:	21 April 2021
Issued date	:	21 April 2021

Patricia Land
Head of Operations
OTHM Qualifications
United Kingdom

www.othm.org.uk

PUNE | BANGALORE | KOCHI

☎ 8055223360

☎ 8882400500

☎ 8103400400

Training Queries:

training@radicaltechnologies.co.in

www.radicaltechnologies.co.in

Aundh | Hinjewadi | Pimple Saudagar | Kharadi | Karve Road | Magarpatta | Banglore | Kochi