* **HTML and jQuery Training**Live Course

**HTML5 and jQuery Course Content**

**HTML Course Content**

**Understanding HTML**

Introduction to the Internet, and Web Technologies, learning about Server, Server types, HTTP Protocol, programming and methods, introduction to HTML programming, creation of HTML tags, marketing element with tags, adding attributes to an element, HTML5 Document Declaration and Document Structure, the strengths of HTML5 – extensive multimedia support, Canvas element, drag and drop features, geolocation, web storage, web workers and web sockets.

**Deep dive into HTML**

Detailed understanding of HTML5, the various elements like Header, Footer, Navigation, Section, Address, Article and Menu Elements, HTML specifications, creation of menu with order and unordered list, command table, distinguishing between Section & Article Element, web form creation and DIV element.

**Hands-on Exercise –**Write an htmp page with Header, Footer, Navigation elements, Create menu with order and unordered list, Create a web form, Create a div and a span element.

**Jquery Course Content**

**Introduction to JQuery**

Introducing JQuery, the objective of JQuery, fast and concise JavaScript Library, learning to use the JQuery Library, JQuery strengths of animation, event handling, HTML document traversing and Ajax interactions, accessing Google Hosted CDN, understanding DOM Loaded and DOM Not Loaded, the difference between window.onload and document.ready, understanding the wrapping element, selecting elements and performing action, finding zero elements, chaining multiple methods, JQuery Version, JQuery Prototype, no-conflict mode.

**Hands-on Exercise –**Traverse HTML document using Ajax interactions and retrived the values of elements, Access Google Hosted CDN

**Working with Selectors & CSS**

Introduction to Cascading Style Sheets, and CSS Selectors, understanding the various functions like HTML, CSS, Val, AddClass, RemoveClass, toggleClass Function, hasClass Function, understanding attr Function, removeAttr Function, prop Function, removeProp Function, Attribute Equal Selector, Attribute Not Equal Selector, Attribute Start with Selector, Attribute Ends with Selector, understanding Class, Element, ID, Lang, Not, Root Selector, working with last-of-type selector, Only Child Selector, Parent, Empty, Form and Button Selector.

**Hands-on Exercise –**Perform css operations to change the properties and alignments of page elements and tags

**Working with JQuery Core**

Introduction to JQuery Core, understanding how JQuery Core is defined in the JQuery Script, understanding the various functions like isEmptyObject, isPlainObject, isNumeric, isWindow Function, isArray Function, isEmptyObject, isPlainObject, the merge and map function, summary.

**Hands-on Exercise –**Use functions isEmptyObject, isPlainObject, isNumeric, isWindow Function, isArray Function, isEmptyObject, isPlainObject, the merge and map function

**Event Handling**

The objectives of JQuery Event Model, Binding Event Handlers, the bind and unbind event attachment, JQuery New Event API, current Target and related Target Property, JQuery Event Classification, form, mouse, keyboard events, trigger functions, custom events, the trigger Handler Function, summary.

**Hands-on Exercise –**Bind Event Handlers, Bind and Unbind event, set current Target and related Target Property, Handle events of form field entry, mouse clicks, keyboard events, trigger functions, custom events, the trigger Handler Function

**Working with DOM Elements**

Understanding the of Document Object Model, DOM Manipulation and function classification, the various Functions like wrap, wrapAll, wrapInner, unwrap, working with append, prepend, DOM Insertion, insertAfter, replaceAll Functions, summary.

**Hands-on Exercise –**Manipulate DOM using jQuery methods like wrap, wrapAll, wrapInner, unwrap, working with append, prepend, DOM Insertion, insertAfter, replace All

**Working with AJAX**

Understanding Asynchronous JavaScript, comparing Synchronous and Asynchronous, the advantages and disadvantages of AJAX, the data types, Get vs Post,  jQuery $.ajax() : The heart, The jqXHR Object, Ajax core method : Settings, A Sample AJAX request, jQuery $.get() Method, jQuery $.post() Method, .load method, AJAX EVENTS, Helper Function.

**Hands-on Exercise –**Use heart plugin to show ratings of a webpage, Use  $.ajax() to define GET and POST request handlers, Use jqXHR Object, Send a AJAX request, Use jQuery $.get() Method, Use $.post() Method

**What projects I will be working on this HTML and Jquery training?**

**Project –** Login Page and Load the data

**Industry :**Internet

**Problem Statement –** How to create a typical login page

**Topics :**It creates a login page which contains two fields that is username and password. If user enters the correct value then it will show a successful message and redirect to another page which includes three options: Load Data, Reload Data, Load Content. This is a hands-on project that helps you understand how HTML and JQuery works.

**Highlights:**

* + Detailed HTML like Header, Footer, Navigation
  + Working with JQuery library
  + DOM programming API
* **Node.js**Live Course

**Introduction to Node.js**

* + What is Node.js?
  + Why Node.js?
  + Node Modules
  + Node Built-in packages.
  + Creating your own package.
  + Import your own Package.
  + NPM (Node Package Manager)
  + Local and Global Packages

**Hands-on:**

* + Install Node.js
  + Run a simple Node.js Script
  + Create a Node Module and use it in your code
  + Search the NPM Repository to find a package and install it in your project

**Asynchronous Programming**

* + What is Asynchronous Programming in JavaScript?
  + JavaScript EventLoop
  + Callbacks
  + Callback Hell
  + Promises
  + Promise Chaining
  + Async Await

**Hands-on:**

* + Demonstrate the use of callbacks, promises, and async-await

**FileSystem**

* + Synchronous vs Asynchronous IO
  + File Constants (\_\_dirname and \_\_filename)
  + Synchronous File Operations
  + Asynchronous File Operations
  + FileSystem Watcher

**Hands-on:**

* + Write Node.js code to asynchronously perform file operations
  + Write Node.js code to log to the console if a file changes in a directory

**Streams and Events**

* + What are Streams in Node.js
  + Reading and Writing Streams
  + Understanding and using EventEmitter class
  + Emitting Events
  + Responding to events

**Hands-on:**

* + Read data from a file using Input Stream and display on screen.

**HTTP**

* + What is HTTP
  + How Does HTTP Work?
  + Standard HTTP response codes
  + Node HTTP Module
  + Processing Form Data
  + Processing Query Strings from URL
  + Sending Back response

**Hands-on:**

* + Create a simple HTTP Server using Node

**ExpressJS**

* + What is ExpressJS?
  + Installing ExpressJS
  + Creating a simple HTTPServer using ExpressJS
  + Routing in Express
  + Templating in Express
  + Templating using JADE
  + Middlewares

**Hands-on:**

* + Create a simple ExpressJS Application with routes and templates

**ExpressJS and MongoDB**

* + What is MongoDB
  + MongoDB Advantages
  + Documents
  + Collections
  + Mongoose ODM
  + Creating Schemas and Models using mongoose
  + CRUD Operations in MongoDB using Mongoose and Express

**Hands-on:**

* + Create a simple TodoList Application using ExpressJS and Mongoose

**Testing**

* + What is Unit Testing?
  + Why Unit Testing?
  + Unit Testing Frameworks
  + What are Mocks and Stubs?
  + Writing and Running tests in Express

**Hands-on:**

* + Write and run some unit tests to test a module

**REST API using Express and Mongoose**

* + What is a REST API
  + Why REST API?
  + Data Exchange Formats
  + JSON vs. XML
  + Monolith Architecture
  + Microservice Architecture

**Hands-on:**

* + Build a small application using the REST API
* **Angular**Live Course

**Angular Course Content**

**Introduction to Angular**

What is Angular, what are its features and advantages, understanding the Angular Technology Stack and Angular Library Components.

**TypeScript**

What is TypeScript, what is its need, how to install TypeScript, Nodemon for monitoring changes, interfaces in Class, String Templates, Maps, Sets and Object Destructuring, Var, Let, and Const keywords.

**Features of Angular**

The top features of Angular, CLI prompts, Angular Compiler, dependency and document updates, Angular elements and improved accessibility.

**How to build with Angular Components**

Building with Angular Components, building web components, custom elements, Angular CLI, ng-packagr, the lifecycle of Angular Components, creating a component and Deeper Nesting.

**Responsive web designing**

What is a responsive web design, building responsive web design with Angular, introduction to Bootstrap, creating responsive layouts with Bootstrap and code design for responsive websites.

**Event binding**

What is event binding, how to create event binding in Angular, building directives, template model, what is SystemJS and Webpack, responding to DOM events mostly created by users and getting inputs by binding to these inputs.

**Dependency injection**

Understanding Dependency Injection, the Dependency Injection API, Angular framework for Dependency Injection, coding pattern for dependencies, overview of service and how to create a service.

**Directives in Angular**

The function of a Directive in Angular, extending the power of HTML, giving it a new syntax, various types of Directives, how to create a custom Directive, different types of built-in Directives and custom structural Directives.

**Pipes in Angular**

What are Pipes in Angular and their features, transforming value in Angular template, various built-in Pipes in Angular, how to use it in any template and creating a custom Pipe in Angular.

**Forms in Angular**

What are Forms in Angular and what are their functions, advantages of Forms, template-driven Forms, reactive Forms, what is Angular Validation and Model driven approach.

**Angular Routing**

What is Angular Routing, its fundamentals, benefits and features, building a single page application and updating it dynamically with Angular Routing, what is Parameter Routing, Router Lifecycle Hooks and Child Routes.

**Promises and Observables**

What are Promises and Observables in Angular, comparing the two, handling HTTP requests with Promise, resolving multiple values with Observables, RxJS library for reactive programming with Observables, interacting with http get() in Angular and http Post.

**Testing Angular applications**

Introduction to testing Angular applications, setup and tools for testing, deploying Angular Test Bed for testing on Angular framework, testing services in Angular, E2E and DOM testing.

* **MongoDB**Live Course

**MongoDB Course Content**

**Introduction to NoSQL and MongoDB**

RDBMS, types of relational databases, challenges of RDBMS, NoSQL database, its significance, how NoSQL suits Big Data needs, introduction to MongoDB and its advantages, MongoDB installation, JSON features, data types and examples

**MongoDB Installation**

Installing MongoDB, basic MongoDB commands and operations, MongoChef (MongoGUI) installation and MongoDB data types

**Hands-on Exercise:** Install MongoDB and install MongoChef (MongoGUI)

**Importance of NoSQL**

The need for NoSQL, types of NoSQL databases, OLTP, OLAP, limitations of RDBMS, ACID properties, CAP Theorem, Base property, learning about JSON/BSON, database collection and documentation, MongoDB uses, MongoDB write concern—acknowledged, replica acknowledged, unacknowledged, journaled—and Fsync

**Hands-on Exercise:**Write a JSON document

**CRUD Operations**

Understanding CRUD and its functionality, CRUD concepts, MongoDB query and syntax and read and write queries and query optimization

**Hands-on Exercise:**Use insert query to create a data entry, use find query to read data, use update and replace queries to update and use delete query operations on a DB file

**Data Modeling and Schema Design**

Concepts of data modelling, difference between MongoDB and RDBMS modelling, model tree structure, operational strategies, monitoring and backup

**Hands-on Exercise:**Write a data model tree structure for a family hierarchy

**Data Management and Administration**

In this module, you will learn MongoDB® Administration activities such as health check, backup, recovery, database sharding and profiling, data import/export, performance tuning, etc.

**Hands-on Exercise:**Use shard key and hashed shard keys, perform backup and recovery of a dummy dataset, import data from a CSV file and export data to a CSV file

**Data Indexing and Aggregation**

Concepts of data aggregation and types and data indexing concepts, properties and variations

**Hands-on Exercise:**Do aggregation using pipeline, sort, skip and limit and create index on data using single key and using multi-key

**MongoDB Security**

Understanding database security risks, MongoDB security concept and security approach and MongoDB integration with Java and Robomongo

**Hands-on Exercise:**MongoDB integration with Java and Robomongo

**Working with Unstructured Data**

Implementing techniques to work with variety of unstructured data like images, videos, log data and others and understanding GridFS MongoDB file system for storing data

**Hands-on Exercise:**Work with variety of unstructured data like images, videos, log data and others

**What projects I will be working on this MongoDB training?**

**Project:**Working with the MongoDB Java Driver

**Industry: General**

**Problem Statement:**How to create table for video insertion using Java

**Topics:**In this project, you will work with MongoDB Java Driver and become proficient in creating a table for inserting video using Java programming. You will work with collections and documents and understand the read and write basics of MongoDB database and the Java virtual machine libraries.

**Highlights:**

* + Setting up MongoDB JDBC Driver
  + Connecting to the database
  + Java virtual machine libraries
* **DevOps**Live Course

**DevOps Certification Course Content**

**Infrastructure Setup**

* + EC2 Walkthrough
  + Installation of DevOps Tools on cloud
    - Git
    - Docker
    - Selenium
    - Maven
    - Jenkins
    - Puppet
    - Ansible
    - Kubernetes
    - Nagios

**Introduction to DevOps**

* + What is Software Development
  + Software Development Life Cycle
  + Traditional Models for SDLC
  + Why DevOps?
  + What is DevOps?
  + DevOps Lifecycle
  + DevOps Tools

**Software Version Control**

* + What is Version Control
  + Types of Version Control System
  + Introduction to SVN
  + Introduction to Git
  + Git Lifecycle
  + Common Git Commands
  + Working with Branches in Git
  + Merging Branches
  + Resolving Merge Conflicts
  + Git Workflow

**Hands-on Exercise –**

* + Git Life cycle Commands
  + Pushing Code to Github
  + Stashing Code in git
  + Creating, Deleting Git Branches
  + Reverting a Push to GitHub
  + Merging branches using git merge
  + Merging branches using git rebase
  + Resolving merge conflicts using git merge tool

**Containerization using Docker - Part I**

* + Introduction to Docker
  + Understanding Docker Lifecycle
  + Components of Docker Ecosystem
  + Common Docker Operations
  + Creating a DockerHub Account
  + Committing changes in a Container
  + Pushing a Container Image to DockerHub
  + Creating Custom Docker Images using Dockerfile

**Hands-on Exercise –**

* + Common Docker Operations
  + Creating a DockerHub Account
  + Committing Changes to a Container
  + Pushing container to DockerHub
  + Creating Local Image Repository
  + Building an Image using Dockerfile

**Containerization using Docker - Part II**

* + What are Docker Volumes
  + Deploying a Multi-Tier Application using Docker Network
  + Using Docker Compose to deploy containers
  + What is Container Orchestration
  + Container Orchestration Tools
  + Introduction to Docker Swarm
  + Deploying a 2-Node Cluster using Docker Swarm

**Hands-on Exercise –**

* + Creating Docker Volumes
  + Using Docker Compose to deploy multiple containers
  + Deploying a Multi Node Cluster using Docker Swarm
  + Deploying a multi-service app on Docker Swarm

**Configuration Management using Puppet**

* + Need of Configuration Management
  + Configuration Management Tools
  + What is Puppet
  + Puppet Architecture
  + Setting up Master Slave using Puppet
  + Puppet Manifests
  + Puppet Modules
  + Applying configuration using Puppet
  + Puppet File Server

**Hands-on Exercise –**

* + Setting up Master Slave on AWS
  + Testing Connection of nodes with Puppet
  + Creating a Manifest
  + Deploying Manifest on Node
  + Creating a Module
  + Deploying sample software on nodes using Puppet Modules and Manifests
  + Implementing a File Server Module on Puppet

**Configuration Management using Ansible**

* + What is Ansible?
  + Ansible vs Puppet
  + Ansible Architecture
  + Setting up Master Slave using Ansible
  + Ansible Playbook
  + Ansible Roles
  + Applying configuration using Ansible

**Hands-on Exercise –**

* + Installing Ansible on AWS
  + Creating a Playbook using YAML
  + Creating an Ansible Role
  + Using Roles in Playbook

**Continuous Testing using Selenium**

* + What is Continuous Testing?
  + Introduction to Selenium
  + What is Maven?
  + Using Maven with Selenium
  + Creating Test Cases with Selenium
  + Running Test Cases on Chromium Web Driver
  + What is Headless Mode?

**Hands-on Exercise –**

* + Using Maven to import dependencies in Eclipse
  + Create Sample Test Case for a website using Selenium
  + Implementing a headless test in selenium using Chrome WebDriver

**Continuous Integration using Jenkins**

* + Introduction to Continuous Integration
  + Jenkins Master Slave Architecture
  + Understanding CI/CD Pipelines
  + Creating an end to end automated CI/CD Pipeline

**Hands-on Exercise –**

* + Creating a Jenkins Master Slave on AWS
  + Installing Plug-ins in Jenkins
  + Creating Jenkins Builds
  + Creating Scheduled Builds
  + Triggering Jobs using Git Web Hooks
  + Using the Pipeline Plugin In Jenkins

**Continuous Orchestration using Kubernetes**

* + Introduction to Kubernetes
  + Docker Swarm vs Kubernetes
  + Kubernetes Architecture
  + Deploying Kubernetes using Kubeadms
  + Alternate ways of deploying Kubernetes
  + YAML Files
  + Creating a Deployment in Kubernetes using YAML
  + Services in Kubernetes
  + Ingress in Kubernetes
  + Case Study – Kubernetes Architecture

**Hands-on Exercise –**

* + Setting up Kubernetes using kubeadm
  + Installing Kubernetes using kops and GCK
  + Creating a Deployment
  + Creating Services
  + Creating an Ingress
  + Demonstrating the use of Ingress, services and deployments together

**Continuous Monitoring using Nagios**

* + What is Continuous Monitoring
  + Introduction to Nagios
  + Nagios Architecture
  + Monitoring Services in Nagios
  + What are NRPE Plugins
  + Monitoring System Info using NRPE plugins

**Hands-on Exercise –**

* + Installing Nagios
  + Monitoring of different servers using Nagios

**What projects I will be working on this DevOps certification training?**

In this DevOps project you will be introduced to the DevOps pipeline demo in various industry domains like media, finance, medical projects and more. You will get hands-on experience in Docker containerization by deploying Jenkins, working with integration tests in DevOps, Project Reports and finance app configuration.

**Project 1 : Containerizing various frameworks and NGINX Application using Docker.**

**Industry :**General

**Problem Statement :**How to integrate the software projects deployed in diverse environments using Docker.

**Topics :**In this Docker project you will containerize the software running in different environment like Node.Js, MongoDB, NGINX, and ASP.NET. You will first setup these frameworks and the NGINX application on Docker container and then run them on the individual containers.

**Highlights**

* + Setup Node.Js on Docker container
  + Pull NGINX image from Docker hub
  + Deploy PowerShell & Hyper-V.

**Project 2: Using Git for version control and tracking of software.**

**Industry :** Internet related

**Problem Statement :** How to ensure the changes on the website made by various developers is implemented.

**Topics :** In this project you will be working on coordinating the work of multiple developers making changes to the website of an online pet’s store. You will focus on source code management, creating a repository on Git for all the codes created by various developers, Next, we check the status of the code, bring it to staging, then commit it, and view its status in log all within the Git repository.

**Highlights**

* + Create a repository on GitHub
  + The various Git commands
  + Push local repository into Git repository.

**Project 3 : Automating the IT infrastructure of a company**

**Industry :**Information Technology

**Problem statement :** How to speed up the setting up and configuration of a software tool of an organization using Puppet

**Topics :** In this project you will be working on installing WordPress along with its dependencies like Apache, PHP, MySQL and others. You will deploy the Puppet configuration management tool, handle the infrastructure like it was code, and automate the entire workflow using Puppet to get the job done faster and better.

**Highlights**

* + Configuration management
  + Infrastructure as Code
  + Deployment automation.

**Project 4 : Automating WordPress application using Kubernetes**

**Industry :** General

**Problem Statement :** How to automate the deployment of WordPress with MySQL with increasing workloads

**Topics :** In this project you will handle the deployment of WordPress with MySQL which is quite challenging with increasing workloads. So,in order to automate the whole process, you will setup the Kubeadmn and Kubectl in your system and deploy the WordPress application along with MySQL using Kubernetes.

**Highlights**

* + Installation of Docker containers
  + Setting up Kubernetes environment
  + Container orchestration with Kubernetes.

**Project 5 :**Implementing a HAProxy server with Docker

**Industry :**Information Technology

**Problem Statement :**How to ensure the HAProxy server is run using a Docker container

**Topics :**In this DevOps project you will create containers using Docker. The Docker compose script will have a code for each container separately, various containers are connected to one another, scaling up and scaling down of the web containers is done. You will write codes for Redis, Load Balancer, and the web and join them later.

**Highlights:**

* + Working with Docker-compose script
  + Connecting multiple Docker containers
  + Creating high performance TCP load balancer.

**Project 6 :**DevOps Continuous Integration

**Industry:**General

**Problem Statement:**Maintaining a state of application using Playbook/Cookbook and using CI tool to run periodic jobs

**Topics**: In this DevOps project you will first write a Playbook/Cookbook/Module for maintaining an application and then using a Continuous Integration or orchestration tool to run jobs on them periodically. You will deploy continuous integration with Software Configuration Management. You can then have the advantage of any deployment tool/orchestration agent on server.

**Highlights**

* + Pushing configuration code to CM
  + Continuous Integration with SCM
  + Playbook/Cookbook for DevOps

**Project 7:** Deployment of LAMP and WordPress using Ansible

**Industry:** Open Source Website

**Problem Statement:** How to get a WordPress website running by installing LAMP stack using Ansible

**Description:** You will automate the deployment of a LAMP stack and install WordPress. LAMP stands for Linux, Apache (a Web server), MySQL (a database) and PHP (server-side scripting). It is a technology stack on which you can deploy different Web applications. Also, configure the installation of WordPress, which is free and open source software for creating websites and blogs. Perform all these operations using Ansible.

**Highlights:**

* + Linux & Ansible Installation
  + Issuing commands from Ansible
  + Deploying LAMP Server.

**Project 8 :** Provision EC2 Virtual Machine with Ansible

**Industry:** Cloud Computing

**Problem Statement:** How to launch a Virtual Machine with AWS EC2

**Topics:** In this Project you will provision your EC2 Virtual Machine using Ansible, install Ansible and run playbooks manually on your local machine to provision a Virtual Machine for more consistency and to put some logic into EC2 parameters.

**Highlights:**

* + Preparing the AWS EC2 VM
  + Preparing Ansible Playbook
  + Running the Playbook.
* **SQL**Live Course

**SQL Course Content**

**Introduction to SQL**

Various types of databases, introduction to Structured Query Language, distinction between client server and file server databases, understanding SQL Server Management Studio, SQL Table basics, data types and functions, Transaction-SQL, authentication for Windows, data control language, and the identification of the keywords in T-SQL, such as Drop Table.

**SQL Operators**

Introduction to relational databases, fundamental concepts of relational rows, tables, and columns; several operators (such as logical and relational), constraints, domains, indexes, stored procedures, primary and foreign keys, understanding group functions, the unique key, etc.

**Working with SQL: Join, Tables, and Variables**

Advanced concepts of SQL tables, SQL functions, operators & queries, table creation, data retrieval from tables, combining rows from tables using inner, outer, cross, and self joins, deploying operators such as ‘intersect,’ ‘except,’ ‘union,’ temporary table creation, set operator rules, table variables, etc.

**Deep Dive into SQL Functions**

Understanding SQL functions – what do they do?, scalar functions, aggregate functions, functions that can be used on different datasets, such as numbers, characters, strings, and dates, inline SQL functions, general functions, and duplicate functions.

**Working with Subqueries**

Understanding SQL subqueries, their rules; statements and operators with which subqueries can be used, using the set clause to modify subqueries, understanding different types of subqueries, such as where, select, insert, update, delete, etc., and methods to create and view subqueries.

**SQL Views, Functions, and Stored Procedures**

Learning SQL views, methods of creating, using, altering, renaming, dropping, and modifying views; understanding stored procedures and their key benefits, working with stored procedures, studying user-defined functions, and error handling.

**Deep Dive into User-defined Functions**

User-defined functions; types of UDFs, such as scalar, inline table value, multi-statement table, stored procedures and when to deploy them, what is rank function?, triggers, and when to execute triggers?

**SQL Optimization and Performance**

SQL Server Management Studio, using pivot in MS Excel and MS SQL Server, differentiating between Char, Varchar, and NVarchar, XL path, indexes and their creation, records grouping, advantages, searching, sorting, modifying data; clustered indexes creation, use of indexes to cover queries, common table expressions, and index guidelines.

**Managing Data with Transact-SQL**

Creating Transact-SQL queries, querying multiple tables using joins, implementing functions and aggregating data, modifying data, determining the results of DDL statements on supplied tables and data, and constructing DML statements using the output statement.

**Querying Data with Advanced Transact-SQL Components**

Querying data using subqueries and APPLY, querying data using table expressions, grouping and pivoting data using queries, querying temporal data and non-relational data, constructing recursive table expressions to meet business requirements, and using windowing functions to group and rank the results of a query.

**Programming Databases Using Transact-SQL**

Creating database programmability objects by using T-SQL, implementing error handling and transactions, implementing transaction control in conjunction with error handling in stored procedures, and implementing data types and NULL.

**Designing and Implementing Database Objects**

Designing and implementing relational database schema; designing and implementing indexes, learning to compare between indexed and included columns, implementing clustered index, and designing and deploying views and column store views.

**Implementing Programmability Objects**

Explaining foreign key constraints, using T-SQL statements, usage of Data Manipulation Language (DML), designing the components of stored procedures, implementing input and output parameters, applying error handling, executing control logic in stored procedures, and designing trigger logic, DDL triggers, etc.

**Managing Database Concurrency**

Applying transactions, using the transaction behavior to identify DML statements, learning about implicit and explicit transactions, isolation levels management, understanding concurrency and locking behavior, and using memory-optimized tables.

**Optimizing Database Objects**

Accuracy of statistics, formulating statistics maintenance tasks, dynamic management objects management, identifying missing indexes, examining and troubleshooting query plans, consolidating the overlapping indexes, the performance management of database instances, and SQL server performance monitoring.

**Microsoft Courses: Study Material**

* + Performance Tuning and Optimizing SQL Databases
  + Querying Data with Transact-SQL

**What are the projects I will be working on during this Microsoft SQL certification training?**

**Project 1**: Writing Complex Subqueries

**Industry**: General

**Problem Statement**: How to create subqueries using SQL?

**Topics**: This project will give you hands-on experience in working with SQL subqueries and utilizing them in various scenarios. Some of the subqueries that you will be working with and gaining hands-on experience in are: IN or NOT IN, ANY or ALL, EXISTS or NOT EXISTS, and other major queries.

**Highlights**:

* + Accessing and manipulating databases
  + Operators and control statements in SQL
  + Executing queries in SQL against databases

**Project 2**: Querying a Large Relational Database

**Industry**: General

**Problem Statement**: How to get details about customers by querying the database?

**Topics**: In this project, you will work on downloading a database and restoring it on the server. You will then query the database to get customer details like name, phone number, email ID, sales made in a particular month, increase in month-on-month sales, and even the total sales made to a particular customer.

**Highlights**:

* + Table basics and data types
  + Various SQL operators
  + Various SQL functions

**Project 3**: Relational Database Design

**Industry**: General

**Problem Statement**: How to convert a relational design into a table in SQL Server?

**Topics**: In this project, you will work on converting a relational design that has enlisted within it various users, user roles, user accounts, and their statuses. You will create a table in SQL Server and insert data into it. With at least two rows in each of the tables, you will ensure that you have created respective foreign keys.

**Highlights**:

* + Defining relations/attributes
  + Defining the primary keys
  + Creating foreign keys