सूक्ष्म, लघु एवं मध्यम उद्यम मंत्रालय
DEVELOPMENT COMMISSIONER
MINISTRY OF MICRO, SMALL & MEDIUM ENTERPRISES
MSME TECHNOLOGY CENTRE

MODEL CURRICULUM

Qualification Name: FULL-STACK DEVELOPER

Qualification Code:

Version: 1.0

NSQF Level: 4.5

Model Curriculum Version: 1.0

Submitted By:

MSME TECHNOLOGY CENTRE

O/o DC MSME, Ministry of Micro, Small and Medium Enterprises

Govt. of India

A-Wing, 7th Floor, Nirman Bhawan, Maulana Azad road

New Delhi-110108

Contact No. +91-674-2654700

Email msmetcab@gmail.com

COURSES / MODULE TEMPLATE

NOS /Module: Understanding the Concepts of C and Python programming

NOS /Module Code: MSME/FSD/01 & version 1.0

Outcomes:

After completion of course Student should be able to:

- Understand about the concept of C Programming.
- Explain basic elements of C programming language.
- Demonstrate the structure of C program
- Demonstrate different data types, variable and keywords and identifier, operator and its uses.
- Explain different control statement, types of loop and go to, continue keyword
- Explain the way to declare, define and call a function.
- Explain pointers- concepts, initialize pointer variables, Character pointers and functions, pointers to pointers
- Explain types of array, store, sort and search data within it.
- Explain string and its pre-defined function.
- Demonstrate the Python program
- Demonstrate the use of print statements
- Demonstrate the knowledge of operator, control system and looping
- Demonstrate the knowledge of functions in python
- Demonstrate the knowledge of OOP's concept

Theory Hours:10 **Practical Hours: - 50** **Theory Marks: 100** **Practical Marks: -100**

Unit No.	Unit Name	Unit Level Outcomes	Contents (Chapters/Topics)	TH Hours	PR Hours	TH Mark	PR Marks
UNIT-I	**C BASICS & KEYWORDS**	After completion of unit Student should be able to- • Understand about algorithmic thinking and apply it to programming. • Understand problem-solving techniques. • Understand about Input and output operation using printf(), scanf(). • Able to Use of if-else, switch, for, while and do-while write programs for	Printing statements Operators Data types Sequential execution Control statements	**10**	**50**	**100**	**100**

		conditional statement and looping.					
UNIT-II	**C DERIVED & USER-DEFINED DATATYPES**	After completion of unit Student should be able to- • Able to Store, retrieve, search and sort of data from an Array. • Describe function and its type and how to pass data types and array to the function. • Able to create user defined data types using Structure and union. • Knowledge to create a new file and then do different operation like read, write and append. • Understand the dynamics of memory by the use of pointers.	Array, Structure, Union, Enum, Pointer, String				
UNIT-III	**PYTHON BASICS**	After completion of unit Student should be able to- • Able to Run the python program using Python IDE • Able to run program to ask details from user and print the resume. • Knowledge Run program to enter a number and print prime if the number is prime otherwise print composite • Able to Run a program to find all Armstrong numbers among three-digit integers and print them • Define a function called is palindrome and write a program that receives a string from the user and prints whether the palindrome is correct or not. • Explain run a program to check the function to find factorial of any number using recursion.	Python program Use of print statements , knowledge of operator, control system and looping, Demonstrate the knowledge of functions in python				

UNIT-IV	**PYTHON USING OOPS CONCEPT**	After completion of unit Student should be able to- • Able to Create a class as class attribute and name, type, as Instance Attributes. • Explain Add methods to the class created above. • Able to run a program to create a text file and write some lines in it using python. • Able to Run a program to save comma separated value file.	Demonstrate the knowledge of OOP's concept, Explain the concept of exception and file handling				

COURSES / MODULE TEMPLATE

NOS /Module: Gaining knowledge in Oracle and MySQL Database Management System

NOS /Module Code: MSME/FSD/02 & version 1.0

Outcomes:

After completion of course Student should be able to:

- Understand about the concept of DBMS.
- Explain the Difference between DBMS and File Base System, DBMS Architecture.
- Explain the concept of Data Definition Language (DDL).
- Explain the concept of Data Manipulation Language (DML) to manipulate over Data.
- Explain the Concept of Different types of constraints.
- Demonstrate how two or more table can be join using different types of join operation.
- Explain the concept of stored procedure and function.

THEORY HOURS: - 10 PRACTICAL HOURS: - 80 THEORY MARKS: -100 PRACTICAL MARKS:-100

Unit No.	Unit Name	Unit level outcomes	Contents (Chapters/Topics)	TH Hours	PR hours	TH Marks	PR Marks
UNIT -I	**Data Base Management System**	**AFTER COMPLETION OF UNIT STUDENT SHOULD BE ABLE TO -** • Define the terminology, features, classifications, and characteristics embodied in database systems. • Demonstrate an understanding of the relational data model.	Install Different Data Base software Like MySQL, Oracle and SQL Server Understand how query are being processed.	**10**	**80**	**100**	**100**

UNIT-II	**SQL QUERIES**	**AFTER COMPLETION OF UNIT STUDENT SHOULD BE ABLE TO-** • Able to create table • Knowledge about Insert update records. • Able to Modify records using constraints.	Basic Data Manipulation using SQL • SQL Syntax • What is DDL, DML and DCL? • SQL Select • SQL Distinct • SQL Where • SQL And & Or • SQL Order By • SQL Insert Into • SQL Update • SQL Delete • SQL Injection • SQL Select Top • SQL Like • SQL Wildcards • SQL In • SQL Between • SQL Aliases • SQL Joins • SQL Inner Join • SQL Left Join • SQL Right Join • SQL Full Join • SQL Union • SQL Select Into • SQL Insert Into Select • SQL Create DB • SQL Create Table • SQL Constraints • SQL Not Null • SQL Unique • SQL Primary Key • SQL Foreign Key • SQL Check • SQL Default • SQL Create Index • SQL Drop • SQL Alter • SQL Auto Increment • SQL Views • SQL Dates • SQL Null Values • SQL Null Functions • SQL Data Types • SQL DB Data Types SQL Functions • SQL Avg() • SQL Count() • SQL First() • SQL Last() • SQL Max() • SQL Min() • SQL Sum() • SQL Group By • SQL Having • SQL Ucase() • SQL Lcase() • SQL Mid() • SQL Len() • SQL Round() • SQL Now() • SQL Format()				

UNIT-III	**MY SQL**	**AFTER COMPLETION OF UNIT STUDENT SHOULD BE ABLE TO-** • Able to use MySQL Workbench & Command Line • Able to create databases, table creation & they can perform multiple operation on the table in the DB. • Understand how database will work so that they can use this concept in their project easily.	**MYSQL DATA DEFINITION USING SQL** • Manage Databases • MySQL Table Types • MySQL Data Types • MySQL CREATE TABLE • MySQL Primary Key • MySQL Foreign Key • MySQL Sequence • MySQL INT Data Type • MySQL DECIMAL Data Type • MySQL DATE Data Type • MySQL TIME Data Type • MySQL DATETIME Data Type • MySQL ALTER TABLE • MySQL RENAME TABLE • MySQL ADD COLUMN • MySQL DROP TABLE • MySQL Temporary Table MYSQL STORED PROCEDURES • MySQL Stored Procedures Introduction • MySQL Stored Procedures Getting Started • MySQL Stored Procedure Variables • MySQL Stored Procedure Parameters • MySQL Stored Procedures – Return Multiple Values • MySQL IF Statement • MySQL CASE Statement • IF vs. CASE • MySQL Loop Statements • MySQL Cursor • MySQL Stored Procedures Listing • MySQL Error Handling • MySQL Stored Procedures – Raising Error Conditions • MySQL Stored Function Introduction to SQL Trigger				

			• MySQL Triggers Implementation • Create Trigger in MySQL • Managing Triggers in MySQL • Create Multiple Triggers For The Same Trigger Event And Action Time • Working with MySQL Scheduled Event • Modifying MySQL Events				

COURSES / MODULE TEMPLATE

NOS /Module: Working Back-End design with Spring Boot & Hibernate Framework

NOS /Module Code: MSME/FSD/03 & version 1.0

Outcomes:

After completion of course Student should be able to:-

- Explain the structure of a java program.
- Demonstrate the Different types of data types, variables, keywords and identifier.
- Explain the concept of different Data types like Arithmetic, logical, assignment, bitwise, instance of, new, conditional.
- Explain the concept of control statement and looping
- Explain the concept of class and object and using object access the member of a class.
- Explain Different ways of representing a method passing primitive data types to method, method overloading, Types of method: instance, static, recursive, factory method. Passing object to method, passing array to method.
- Explain the use of inheritance in programming Language.
- Create a Thread by using Thread class and Runnable interface, Lifecycle of Thread, Thread class constructor and methods, Thread priorities, thread group, multithreading and synchronization
- Explain the Concept of file
- Explain the concept of TCP/IP protocol, sockets, Knowing IP Address, URL and URL Connection class

- Develop GUI application using swing through Net Beans IDE.
- Explain the Architecture of JDBC and Types of JDBC Driver.
- Explain the Concept of Statement, Prepared Statement
- Execute stored procedure and function using Callable Statement Interface.
- Explain the concept of Spring & Spring Boot.
- Develop CRUD application using Hibernate with Data JPA.
- Explain the concept of Restful web services.

THEORY HOURS: 20 PRACTICAL HOURS: -190 THEORY MARKS: 100 PRACTICAL MARKS: -100

Unit No.	Unit Name	Unit level outcomes	Contents (chapters/topics)	TH Hours	PR hours	TH Marks	PR Marks
Unit I	**BASICS OF JAVA**	**AFTER COMPLETION OF UNIT STUDENT SHOULD BE ABLE TO -** • Understand the Sequential execution using predefined classes and packages. • Knowledge about Control statements execution. • Understand Predefined and undefined data types.	Understand JVM and its Architecture, Understand the concept of control statement and looping Define class and object and Constructor.	**20**	**190**	**100**	**100**
Unit II	**OOPS CONCEPT IN JAVA**	**AFTER COMPLETION OF UNIT STUDENT SHOULD BE ABLE TO -** • Able to do coding using oops concept • Able to do programs using Interfaces and • Knowledge about Abstract classes and methods.	Describe Different Types of method and types • Write program by using abstract class and interface to understand the concept of inheritance • Create a package and convert into a jar file and then use that jar file in a class.				

Unit III	**MULTI THREADING, PACKAGES & EXCEPTION HANDLING**	After completion of unit Student should be able to • Able to handle different types of errors using exceptional handling mechanism. • Explain How multithreading occurs in a program.	• Handle the Exception using try, catch, finally, throws and throw • Create file and do different operation like read, write and append • Design Graphical User Interface (GUI) program using AWT and Swing • Connect Your Program to Data Base using JDBC.				
Unit IV	**Collections Framework**	After completion of unit Student should be able to • Understand Collections framework reduces programming effort: By providing useful data structures and algorithms, the Collections Framework frees you to concentrate on the important parts of your program rather than on the low-level "plumbing" required to make it work.	• Interfaces *(Set, List, Queue, Deque)* • Classes *(ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).*				
Unit V	**INTERACTING WITH DATABASE**	After completion of unit Student should be able to- • Understand about JDBC & ODBC depending on the given application requirement. • Able to use relevant type of JDBC driver for the specified environment. • Elaborate steps with examples to establish connectivity with the specified databases.	Understand the concept of Statement Prepared Statement and Callable Statement interface to connect Java program with Data base Execute Stored procedure and function using JDBC Callable Statement Interface Driver Interfaces and Driver Managers class : Connection Interface Statement ,prepared Statement Interface, Result set Interface				

Unit VI	**SPRING & SPRING BOOT**	After completion of unit Student should be able to- • Understand about the concept of spring & spring boot. • Able to create spring boot projects. • Able to use different annotation. • Understand the concept of Lombok.	Introduction to spring and spring boot. Creating a spring boot project IOC and DI, Constructor Injection and Setter Injection @Autowired, @Qualifier @Resource and Other Annotation Spring container, Spring bean, Scope of spring bean, Spring bean life cycle. Properties file, Spring profile Working with Logger in spring boot Project Lombok JUnit inspiring boot				
Unit VII	**HIBERNATE WITH DATA JPA**	After completion of unit Student should be able to- • Understand about the concept of Hibernate with Data JPA. • Able to design crud Application. • Understand the concept of JPA Mapping.	Introduction To JPA, Hibernate and MYSQL MYSQL - Creating table, and Working with SQL Queries Entity in spring boot and Creating JPA Entity Repository and CRUD Repository and JPA Repository Paging and sorting Repository JPA named Quires and @Query annotation Working with Procedure in Hibernate Hibernate and JPA Mapping - One to One and one to many Hibernate and JPA Mapping - Many to one and Many to Many Hibernate Caching				

UNIT VIII	**RESTFUL WEB SERVICES**	After completion of unit Student should be able to- • Understand about the concept of Restful Web Services. ▪ Able to Build Restful Web services using spring boot and data JPA. • Able to design backend application.	Introduction to Rest and SOAP, Build Restful Web services using spring boot and data JPA Restful web services CRUD application and Test through Postman. @Rest Controller Restful web services CRUD application using Hibernate – 1 Restful web services CRUD application using Hibernate - 2 Spring boot Swagger Spring Boot - Actuator & Micrometer				

COURSES / MODULE TEMPLATE

NOS /Module Mastering UI design with Angular.

NOS /Module Code: MSME/FSD/04 & version 1.0

Outcomes:

After completion of course Student should be able to:-

- Demonstrate the Structure of an html Page.
- Demonstrate different types of basic html tag.
- Demonstrate the use of list and table tag.
- Demonstrate the different types of form tag and their uses for User interfaces.
- Demonstrate External Style Sheets, Internal Style Sheets, and Inline Style, The class selector, div & span tag.
- Able to run a HTML Program using manual process and Adobe Dream weaver IDE.
- Able to design and develop a web page using heading, font, image, marquee tag.
- Develop a web page using order list, unordered list and definition list.
- Design a Student registration form that will take all student Data including their photo and CV.

- Design a web page using html and CSS with batter look and fill and then understand the concept of class and id in CSS.
- Design web page using Angular JS.

THEORY HOURS: -10 PRACTICAL HOURS: 50 THEORY MARKS: -100 PRACTICAL MARKS: 100

Unit No.	Unit Name	Unit Level Outcomes	Contents (Chapters/Topics)	TH Hours	PR hours	TH Marks	PR Marks
Unit I	**HTML**	After completion of unit Student should be able to - • Knowledge to create a webpage using html basic tags. • Able to change webpages font size and style, colors. • Understand how to properly markup content using semantic HTML tags. • Able to build unique websites • Understand how to create links between different pages to form an overall website • Able to modify the structure of emails and other web documents.	<u>INTRODUCTION AND ADVANTAGES OF HTML5</u> • Introduction HTML • HTML Basics • HTML Elements • HTML5 Semantic • HTML Attributes • HTML Headings • HTML Paragraph • HTML Styles • HTML Formatting • HTML Quotations • HTML Computer Code • HTML Comments & Colours • HTML CSS, Links and Images • HTML Lists • HTML Blocks • HTML Classes • HTML Layout • HTML Responsive • HTML iframes • HTML JavaScript • HTML Head • HTML Entities and URI Code • HTML Symbols and XHTML • HTML Charset and Forms	**10**	**50**	**100**	**100**

Unit II	**CSS**	After completion of unit Student should be able to - • Knowledge to add styling to the website. • Explain to modify text and colors using selectors. • Using Box model student can Organized HTML elements on Screen. • Able to add styling to the website. • Able to modify text and colors using selectors. • Able to creating visual design of web content. • Adapting web pages to different devices and browsers. • Able to Use Flex box and Grid Student can able to Create All Shots of Complex layouts. • Able to Use Transitions Student can Control animation speed when changing CSS properties.	<u>INTRODUCTION ABOUT THE CONCEPT OF CSS</u> • Introduction CSS3 • CSS3 Syntax • CSS3 How To • CSS3 Colours • CSS3 Backgrounds • CSS3 Boarders • CSS Padding • CSS Height/Width • CSS3 Gradients • CSS3 Shadows • CSS3 Text • CSS3 Fonts • CSS3 2D Transforms • CSS3 3D Transforms • CSS Links • CSS Lists • CSS Tables • CSS Box Model • CSS Outline • CSS Display • CSS Max-width • CSS Position • CSS Float • CSS Inline-block • CSS Align • CSS Combinators • CSS Pseudo-class • CSS Pseudo-element • CSS Navigation Bar • CSS Dropdowns • CSS Tooltips • CSS3 Images • CSS Attr Selectors • CSS Forms • CSS Counters • CSS3 Animations • CSS3 Buttons • CSS3 Pagination • CSS3 Multiple Columns • CSS3 User Interface • CSS3 Box Sizing • CSS3 Filters • CSS3 Media Queries • CSS3 Responsive				

Unit III	**JAVA SCRIPT**	**AFTER COMPLETION OF UNIT STUDENT SHOULD BE ABLE TO-** • Understand about the concept of JAVA SCRIPT. • Able to make websites work • Able to define functions and methods • demonstrate handling web page events • Understand how to build web applications using JS and be familiar with its libraries and framework • Using Events Handling Student can detect when certain events happen, and cause things to occur in response to those events. • Able to be document object model allows you to access and manipulate all the html elements on a page.	**INTRODUCTION ABOUT JAVA SCRIPT** • JS Statements • JS Syntax • JS Comments • JS Variables • JS Operators • JS Arithmetic • JS Assignment • JS Data Types • JS Functions • JS Objects • JS Numbers • JS Number Methods • JS Math EVENT HANDLING • JavaScript events • Event handler • Event flow • Event bubbling and capturing • Event listeners • Event types DOCUMENT OBJECT MODEL (DOM) • Introduction to DOM • Types of DOM • DOM standards and methods • Manipulating documents using DOM • Handling images • Table manipulation • Animation • Node and Node-list handling BROWSER OBJECT MODEL (BOM) • Introduction to BOM • DOM vs BOM differences • Window object and methods • BOM navigator • BOM history • BOM location • BOM timer • Introduction to Cookies • Session and persistent cookies FORM HANDLING				

			• Introduction to forms • Form processing • Forms object • Accessing data from forms • Form validation • Additional features in forms • Validation APIs <u>DEBUGGING TECHNIQUES</u> • JavaScript Errors • Error handling mechanisms • Introduction to Google Chrome debugger • Deep dive into debugger window • Introduction to Breakpoints • Changing variable values in runtime Avoiding mistakes				
Unit IV	Angular	After completion of unit Student should be able to- ▪ Understand about the concept of Angular JS ▪ Understand about the concept of Angular Components. ▪ Understand about the concept of Angular Directives. ▪ Understand about the concept of Backend Services ▪ Able to implement the concepts & doing some projects.	▪ Introduction to Angular ,install VS code & plugins ▪ Setting up workspace, running the first angular application ▪ Understanding angular components, Templates, directive and dependency injections. ▪ Adding bootstrap with angular. ▪ Introduction to type Script, property and data binding. ▪ Doing exercises related to property binding. ▪ Introduction to components and components lifecycle ,components interaction 1) @input & @ output decorator ▪ Templates – class and style binding, Event binding. ▪ Pipes in angular , creating user defined pipes ▪ NgClass , Ngstyle , NgModel ▪ Attribute directive ▪ Structural directive – ngif , ngFor , ngSwitch ▪ Angular Routing &				

			Navigation –Router Link ▪ Angular Form ▪ Adding validation implementing custom validation ▪ Understanding communicating with backend service using HTTP , Introduction to JSON ,HTTP Methods. ▪ Backend service using get and post. ▪ Updating and deleting data from backend service. ▪ Explain a dummy Angular Project Deploying Angular App and course roundup				

COURSES / MODULE TEMPLATE

NOS /Module: Deploying & Hosting Application with Docker & Kubernetes

NOS /Module Code: MSME/FSD/05 & version 1.0

Outcomes:

After completion of course Student should be able to:-

- Explain the concepts of cloud computing.
- Understand the importance of virtualization in distributed computing and how this has enabled the development of Cloud Computing.
- Analyze the performance of Cloud Computing.
- Understand the concept of Cloud Security.
- Learn the Concept of Cloud Infrastructure Model.
- Understand about the concept of AWS.
- Able to create AWS Account.
- Demonstrate the concept of IAM & AWS CLI.
- Able to create an EC2 Instance.
- Understand the concept of Amazon RDS.
- Able to deploy Spring Boot application on AWS using Elastic Beanstalk.
- Able to explain the concepts of Docker & Kubernetes.

THEORY HOURS: 10 PRACTICAL HOURS: -80 THEORY MARKS: 100 PRACTICAL MARKS: -100

Unit No.	Unit Name	Unit level outcomes	Contents (chapters/topics)	TH Hours	PR hours	TH Marks	PR Marks
Unit I	**Basic of Cloud Computing**	After completion of unit Student should be able to – • Explain the core concepts of the cloud computing paradigm • Able to Understand how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.	• Recent trends in computing, evolution of cloud computing, Cloud computing (NIST model), properties, characteristics and disadvantages, role of open standards. • Cloud computing stack, Service models (XAAS), Deployment models.	**10**	**80**	**100**	**100**
Unit ii	**AWS**	After completion of unit Student should be able to – • Understand about the concept of AWS. • Able to create AWS Account. • Demonstrate the concept of IAM & AWS CLI. • Able to create an EC2 Instance. • Understand the concept of Amazon RDS. • Able to deploy Spring Boot application on AWS using Elastic Beanstalk.	• Introduction to AWS(Amazon Web Services). • Create an AWS Account. • Demonstrate the concept of IAM & AWS CLI. • create an EC2 Instance. • Introduction to Amazon RDS. • Deployment Spring Boot application on AWS using Elastic Beanstalk.				

Unit iii	**Working with Dockers**	***Module 1: Docker World – An introduction (Duration-5hrs)***	➢ Introducing Docker ➢ Comparing VM and Docker ➢ Docker –An Architectural overview ➢ The Docker Hub A brief Introduction ➢ Preparing docker-machine – Installation and configuration ➢ Start containerizing ➢ Play with docker images ➢ Customizing container on your own ➢ Running Container with Docker – commands ➢ Port forwarding with docker container ➢ Exercise: Installation of docker and Image Setup ➢ Exercise: Creating own Images ➢ Exercise: Creating own Images ➢ Exercise: Exposing Container Ports to the Host and test it				
		Module 2:The Dockerfile, Builds and Network Configuration (Duration-5hrs)	➢ Dockerfile Directives ➢ USER and RUN ➢ RUN Order of Execution ➢ ENV ➢ CMD vs. RUN ➢ ENTRYPOINT ➢ EXPOSE ➢ Docker Container Volume Management – An introduction ➢ Docker Networking concepts ➢ List and Inspect				

			➢ Create and Remove ➢ Assign to Containers ➢ Exercise: Creating a Custom Image from a Docker file ➢ Exercise: Managing Containers ➢ Exercise: Adding External Content to Containers				
		Module 3: Docker Commands and Structures (Duration-5hrs)	➢ Inspect Container Processes ➢ Previous Container Management ➢ Controlling Port Exposure on Containers ➢ Naming Our Containers ➢ Docker Events ➢ Managing and Removing Base Images ➢ Saving and Loading Docker Images ➢ Image History ➢ Taking Control of Our Tags ➢ Pushing to Docker Hub ➢ Exercise: Base Image Maintenance and Cleanup ➢ Exercise: Advanced Container Creation at the Command Line ➢ Exercise: Create a Dockerized Basic Web Server ➢ Continuous Integration for Docker				
		Module 4: Docker-Compose (Duration-5hrs)	➢ Networking Overview ➢ The Default Network ➢ Isolating Containers				

			➢ Aliases & Container Names ➢ Links ➢ How Updates Affect Networking ➢ Using External Networks ➢ Configuring Compose ➢ Bringing an Environment Up ➢ Changing a Running Environment ➢ Introspecting On An Environment ➢ Taking an Environment Down				
		Module 5: Docker swarm – A deep dive (Duration-5hrs)	➢ Swarm Intro and Creating a 3-Node Swarm Cluster ➢ Swarm Mode A Built-In Orchestration ➢ Creating Your First Service and Scale It Locally ➢ Creating a 3-Node Swarm Cluster ➢ Swarm Basic Features and How to Use Them In Your Workflow ➢ Scaling Out with Overlay Networking ➢ Create A Multi-Service Multi-Node Web App ➢ Service Placement Preference ➢ Node Availability				
Unit iv	**Working with Kubernets**	Introduction to kubernetes, setup and running docker containers. Running your first app in kubernetes and load balancing using load balancer.	➢ Containers introduction ➢ Kubernetes setup ➢ Local setup with minikube ➢ Installing kubenetes using the docker client				

			➢ Minikube vs docker client vs kops vs kubeadm ➢ Introduction to kops ➢ Building docker images ➢ Docker image registry ➢ Running 1st app on kubernetes ➢ Service with load balancer ➢ Service with AWS ELB load balancer ➢ Practise test. **<u>Hands-On</u>** ➢ Kubernetes setup using Minikube ➢ Building docker images ➢ Run app in kubernetes ➢ Load balance the app ➢ Load balance the app using AWS load balancer				
		Understanding the architecture and pod lifecycle, creating different services, replicas and setting the healthchecks for your pods.	➢ Node architecture ➢ Replication controller ➢ Deployments ➢ Services ➢ Labels ➢Healthchecks ➢ Readiness probe ➢ Pod state ➢ Pod lifecycle ➢ Secrets ➢ WebUI ➢ Practise test				

			Hands-On ➢ Deploying different kind of services. ➢ Replica set and replication controller ➢ Deployments and health checks				
		Creating persistent volumes, setting up auto scaling and tolerations, monitoring the resource usage.	➢ Service directory ➢ Configmap ➢ Ingress controller ➢ External DNS ➢ Volumes ➢ Volumes auto-provisioning ➢ Pod presets ➢ Statefulsets ➢ Daemon sets ➢ Resource usage monitoring ➢ Auto scaling ➢ Affinity / auto-affinity ➢ Interpod affinity and anti-affinity ➢ Taints and tolerations ➢ Custom Resource Definitions (CRDs) ➢ Operators ➢ Practise test **Hands-On** ➢ Using volumes ➢ Using daemon sets ➢ Auto scaling ➢ Working with statefulsets ➢ Monitoring the resources				

			➢ Using tolerations ➢ Creating affinity and interpod affinity				
		Managing the users, Maintaining the nodes, making the nodes highly available and using TLS	➢ The kubernetes master services ➢ Resource quotas ➢ Namespaces ➢ User management ➢ RBAC ➢ Networking ➢ Node maintenance ➢ High availability ➢ TLS on ELB using annotations **Hands-On** ➢ Managing user ➢ Working with networking ➢ Viewing the resource quotas ➢ Maintainin the nodes ➢ Making the nodes highly available				
		Understanding and creating helm charts.	➢ Introduction to Helm ➢ Creating your own helm charts Hands-On ➢ Working with Helm ➢ Own helm charts				
		Understanding skaffold on a high level.	➢ Introduction to skaffold **Hands-On** ➢ Getting to know skaffold				

		Understang serverless and kuberless and what you can do with this	➢ Introduction to serverless ➢ Introduction to kuberless **Hands-On** ➢ Serverless and kuberless				
		Microservices Understanding istio, authenticating the users using istio	➢ Introduction to istio ➢ Mutual TLS ➢ RBAC with istio ➢ End-user authentication with istio (JWT) **Hands-On** ➢ Working with Istio ➢ Authenticating using Istio				
		Installing kubernetes using kubeadm	➢ Installing kubernetes using kubeadm. ➢ Introduction to kubeadm Hands-On ➢ Running basic commands using kubeadm On-prem or cloud agnostic ➢ Using TLS certificates, understanding kubernetes dashboard and working with prometheus. ➢ Managing TLS certs with cert-manager ➢ Kubernetes dashboard ➢ Kubernetes with Prometheus Hands-On ➢ Using TLS certificates ➢ Dashboard in kubernetes ➢ Using prometheus with kubernetes				

COURSES / MODULE TEMPLATE

NOS /Module: Employability Skill

NOS /Module Code: MSME/ES/01

THEORY HOURS: 30 PRACTICAL HOURS: - THEORY MARKS: 100 PRACTICAL MARKS: -

Refer Standard Curriculum developed by NCVET. (https://nqr.gov.in/downloads/pdfs/60-hours_MC_Employability_Skills.pdf)