Program	Bachelor of Technology (BTech)	**Semester - 4**
Type of Course	Professional Electives	
Prerequisite	Web Design, DBMS, Object Oriented Programming Concepts	
Course Objective	This course provides the skills to develop robust, scalable web applications using ASP.NET Core. It covers essential concepts like MVC architecture, security, and deployment.	
Effective From A.Y.	2023-24	

Teaching Scheme (Contact Hours)				Examination Scheme				
				Theory Marks		Practical Marks		
Lecture	**Tutorial**	**Practical**	**Credit**	**SEE (T)**	**CIA (T)**	**SEE (P)**	**CIA (P)**	**Total Marks**
3	0	4	5	40	30	20	10	100

***SEE** - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)*

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
1	**C# Fundamentals** Overview, Namespace, Variables, Data Types, Operators, Strings, Conditions & Looping, Collection Classes, Exception Handling, Classes/Objects, Constructors, Access Modifier, Properties, Inheritance, Interface	10	20
2	**Introduction to ASP.NET Core** Overview, ASP.NET Core Features, Advantages, MVC Architecture Pattern, Environment Setup, Project Layout, ASP.Net Core Life Cycle, Request Response Pipeline & Middleware **Areas, Controllers & Action Methods:** Understanding Areas, Adding Areas, Defining Area Routes, Linking Between Areas, Controllers Overview, Action Methods and IActionResult object, Understanding Action Selectors, Action Filters, Non Action Methods	8	20
3	**Views & Helpers** Overview, Advantages of Razor View, Razor Syntax, Types of Views, Partial Views, Layout Pages, Helpers Introduction, Html Helpers, URL Helpers, Tag Helpers, and Custom Tag Helpers	10	20
4	**Data Annotations, Validations, Model Binding & Data Passing** Validation Overview, Validations with Data Annotation, Client Side and Server-Side Validation, Html Form behavior, Default Model Binder, IFormCollection Model Binding, Bind Attribute, Data Passing Techniques	9	20
5	**Routing, State Management & ADO.NET** Routing Overview, Custom Routes, Attribute Routing, Routing Constraints, ADO.NET Architecture, .NET Framework Data Providers, Connection & Command Object, DataReader Object, Query Strings, Session	8	20
	Total	45	100

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	30	40	30	0	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	**discuss** the basic concepts of .NET Framework and fundamentals of C# programming language
CO2	**classify** the MVC Pattern, Controllers & Action Methods
CO3	**perform** views with different layouts with tag helpers
CO4	**implement** data annotations for model validations
CO5	**apply** ADO.Net objects/classes to communicate with database

Reference Books

1.	**Pro ASP.NET Core6** By Adam Freeman | Apress | 9th
2.	**ASP.NET Core in Action** By Andrew Lock | Manning | 3rd
3.	**ASP.NET Core Razor Pages in Action** By Mike Brind | Manning
4.	**ASP.NET Core5.0 – A Complete Guide for Beginners** By Pradeep Gohil
5.	**Programming Asp.Net Core** By Dino Esposito | Microsoft Press

List of Practical

1.	**To implement concept of Variables, Data Types, Operators** 1. (A) Write a program to print your name, address, contact number & city. 2. (A) Write a program to get two numbers from user and print those two numbers. 3. (A) Write program to prompt a user to input his/her name and country name and then output will be shown as given: Hello from country 4. (A) Write a program to calculate the size of the area in square-feet based on Specified length and width. 5. (A) Write a program to calculate area of Square, Rectangle and Circle. 6. (B) Write a program to calculate Celsius to Fahrenheit and vice-versa using function. 7. (B) Write a program to find out Simple Interest using function. (I = PRN/100) 8. (B) Write a program to create a Simple Calculator for two numbers (Addition, Multiplication, Subtraction, Division) [Also using if…else & Switch Case] 9. (C) Write a program to Swapping without using third variable. 10. (C)Write a program to find maximum numbers from given 3 numbers using ternary operator.
2.	**To implement concept of Class and Object, Constructors, Inheritance** 1.(A) Write a program to create a class named Candidate with ID, Name, Age, Weight and Height as data members & also create a member functions like GetCandidateDetails() and DisplayCandidateDetails(). 2.(A) Write a program to create a class Staff having data members as Name, Department, Designation, Experience & Salary. Accept this data for 5 different staffs and display only names & salary of those staff who are HOD. 3.(A) Write a pogram to Create a class Bank_Account with Account_No, Email, User_Name, Account_Type and Account_Balance as data members. Also create a Member function GetAccountDetails() & DisplayAccountDetails(). 4.(A) Write a program with following specifications: Class Name: Student Data Members: Enrollment_No, Student_Name, Semester, CPI and SPI Get Students Details using constructor and DisplayStudentDetails() using member function. 5.(A) Write a program to calculate area of a Rectangle using constructor. 6.(A) Write a program for implementing single inheritance which creates one class Account_Details for getting account information and another class

	Interest for calculating and displaying total interest from the data inserted from account details. 7.(B) Write a program to Define a class Salary which will contain member variable Basic, TA, DA, HRA. Write a program using Constructor with default values for DA and HRA and calculate the salary of employee. 8.(B) Write a program to Define a class Distance have data members dist1, dist2, dist3. Initialize the two data members using constructor and store their addition in third data member using function and display addition. 9.(C) Create a class Furniture with material ,price as data members. Create another class Table with Height , surface_area as data members. Write a program to implement single inheritance. 10.(C) Program to implement the following multiple inheritance using interface. Interface: Gross Method- Gross_sal() Class : Salary Data Members - HRA, TA,DA Methods - Disp_sal() Class : Employee Data Members - Name Methods - basic_sal()
3.	**Implement concept of Exception Handling, Interface, Abstraction, String Functions** 1.(A) Write a program to Create a divide by zero exception and handle it. 2.(A) Write a program that reads 5 numbers from user. Demonstrate concept of IndexOutOfRange Exception. 3.(A) Write a program to create an abstract class Sum having abstract methods SumOfTwo(int a, int b) and SumOfThree(int a, int b,int c). Create another class Calculate which extends the abstract class and implements the abstract methods. 4.(A) Write a program to create interface Calculate. In this interface we have two member functions Addition() and Subtraction(). Implements this interface in another class named Result. 5.(A) Write program showing use of common methods of String class. 6.(B) Write a program to Replace lower case characters to upper case and Vice-versa. 7.(B) Write a program to create interface named Shape. In this interface, we have three methods Circle(), Triangle() and Square() which calculates area of Circle, Triangle and Square respectively. Implement Shape interface. 8.(B) Write a program to accept a number from the user and throw an exception if the number is not an even number. 9.(C) Write a program to find the longest word in a string. 10.(C) Write a program to change the case of entered character.
4.	**Write program using Method Overloading, Method Overriding** 1.(A) Write a program using method overloading by changing datatype of arguments to perform addition of two integer numbers and two float numbers. 2.(A) Write a program using method overloading by changing number of arguments to calculate area of square and rectangle. 3.(A) Create a class named RBI with calculateInterest() method. Create another classes HDFC, SBI, ICICI which overrides calculateInterest() method. 4.(A) Create a class Hospital with HospitalDetails() method. Create another classes Apollo, Wockhardt, Gokul_Superspeciality which overrides HospitalDetails() method. 5.(B) Write a programs to Find Area of Square, Rectangle and Circle using Method Overloading. 6.(C) Create a BankAccount class having constructor accepting initialBalance and accountHolderName. Also create Deposite() and withdraw() overloaded methods by which user can deposit/withdraw amount using different types of parameters (ex. cash, check).
5.	**To demonstrate concept of Collection Classes** 1.(A) Create an ArrayList for StudentName and perform following operations: a. Add() - To Add new student in list b. Remove() - To Remove Student with specified index c. RemoveRange() - To Remove student with specified range. d. Clear() - To clear all the student from the list 2.(A) Create a List for StudentName and perform following operations: a. Add() - To Add new student in list b. Remove() - To Remove Student with specified index

	c. RemoveRange() - To Remove student with specified range. d. Clear() - To clear all the student from the list 3.(B) Create a Stack which takes integer values and perform following operations: a. Push() - To Add new item in stack b. Pop() - To Remove item from the stack c. Peek() – To Return the top item from the stack. d. Contains() - To Checks whether an item exists in the stack or not. e. Clear() - To clear items from stack 4.(B) Create a Queue which takes integer values and perform following operations: a. Enqueue() - Adds an item into the queue. b. Dequeue() - Returns an item from the beginning of the queue and removes it from the queue. c. Peek() - Returns an first item from the queue without removing it. d. Contains() - Checks whether an item is in the queue or not e. Clear() - Removes all the items from the queue 5.(C) Create a Dictionary collection class object and preform following operations: a. Add: Adds a key-value pair. b. Remove: Removes a key-value pair by key. c. ContainsKey: Checks if a key exists in the hashtable. d. ContainsValue: Checks if a value exists in the hashtable. e. Clear: Removes all key-value pairs. 6.(C) Create a Hashtable collection class object and preform following operations: a. Add: Adds a key-value pair. b. Remove: Removes a key-value pair by key. c. ContainsKey: Checks if a key exists in the hashtable. d. ContainsValue: Checks if a value exists in the hashtable. e. Clear: Removes all key-value pairs.
6.	**To setup project environment and MVC Overview with Visual Studio** 1.(A) Introduction to IDE, how to create project of .net core, how to add controllers, action methods and views. How to add NuGet package references. 2.(A) Create a project and add Home Controller with Home, About and Contact Us Action methods with Views. Add appropriate navigation between these pages. 3.(B) Add Admin, User Areas and Define Appropriate Area Routes and Linking Between Areas. Add Home, About and Contact Us views inside User Area.
7.	**Demonstrate concept of Razor Syntax by displaying student details** 1.(A) Print table of 5 using Razor. 2.(A) Prepare a page which displays student details and his/her semester wise SPI in table format. 3.(B) Prepare semester wise SPI table data in controller file and store it in ViewBag. Display the data to view page using foreach loop.
8.	**Demonstrate concept of bootstrap theme using attribute routing** 1.(A) Create a project and add Home, About, Contact Us views. And add appropriate routing between these pages. Use bootstrap for better design. (use conventional routing) 2.(A) Add attribute routing in above practical along with optional parameters and apply ignore route templates functionality.
9.	**Implement concept of Static CRUD operation on Employee model** 1.(A) Prepare a static CRUD for employee table/model which displays employee details in tabular format. Create employee model class for it and use List collection class object to pass data from controller to view. 2.(B) Do create contact us form & display entered data below in same view page.
10.	**Conversion of Custom bootstrap Templates into MVC project to demonstrate concept of layout page** 1.(A) Single page bootstrap theme conversion [Personal CV]. 3.(B) Multiple page admin theme conversion for the project with required pages.

11.	**Create Database and prepare stored procedures for Select command** 1.(A) Create Database : AddressBook also create SelectAll and SelectByPK stored procedure for Country, State and City tables. 2.(B) Pass the City Name in procedure and display all the records based on city name. 3.(B) Pass the State Name in procedure and display all the cities belongs to the that state. 4.(C) Pass the Country name in procedure and display all the states with number of cities.
12.	**Prepare stored procedure for Insert, Update and Delete command** 1.(A) Create all tables Insert, Update and Delete stored procedures for Country, State and City tables 2.(B) Create an Insert, Update & Delete Stored Procedure for your own one table with minimum 6-7 columns
13.	**Hands on practice to standard HTML helpers and strongly typed HTML helpers by creating student registration form** 1.(A) Student registration form using Standard html helpers.(StudentName, Branch, Semester, Birthdate, Mobile, Email, Address, City, Hobbies, Gender) 2.(A) Student registration form using Strongly typed html helpers.(StudentName, Branch, Semester, Birthdate, Mobile, Email, Address, City, Hobbies, Gender) 3.(B) Employee Add Form using Both Helpers. 4.(C) Job Inquiry form using Strongly typed html helpers.
14.	**Perform of file upload operation** 1.(A) Design a view by which user can upload his/her resume to the server and display the uploaded resume. 2.(B) Design a view from user can upload their profile picture.
15.	**Create a project and implement area** 1.(A) Create a new asp.net core project with MVC Template and Create appropriate MVC Areas for AddressBook Project.
16.	**Create design for view pages** 1.(A) Design List Page & Add/Edit Pages.
17.	**Implement attribute routing between view pages** 1.(A) Apply Attribute Routing between all the views. Use appropriate routing attributes as required.
18.	**Create Model classes and apply appropriate Data Annotations** (A) Prepare model classes as per requirement and Implement data annotation on all the model classes
19.	**Database connectivity and Implementation of read operation** 1.(A) Create database connectivity and Display data (All Records) for Country List view page. 2.(B) Display the data for State List and City List view pages.
20.	**Implementation of delete functionality** 1. (A) Implement Delete functionality for City, State, Country. 2. (B) Add prompt with 'Are you sure you want to delete record? ' using Sweet alert when user want to delete any record.
21.	**Perform server side validation** (A) Apply server-side validations with proper message for Country, State & City.
22.	**Implementation of insert functionality** 1.(A) Implement insert functionality for Country view page. 2.(B) Add dropdown for Country, State as required and implement insert functionality for State, City view pages with required validations
23.	**Implementation of update functionality** 1.(A) Implement update functionality for Country, State & City view pages with appropriate message as required.
24.	**Demonstrate concept of session using Login functionality** 1.(A) Implement Login functionality. 2.(B) Implement User Registration functionality.
25.	**Implementation of Search and Excel functionality**

	1.(A) Implement Search functionality for all the list pages. 2.(B) Add a button by which user can export table data to excel.
26.	**Demonstrate concept of URL encryption and decryption** 1.(A) Perform URL Encryption and Decryption using standard encryption decryption algorithm in whole project