MCA

Semester-III

L-2 T-1 P-0 C-3

25MCA400T:ASP.Net with C#

Course Objectives

- To learn basics of ASP.Net
- To learn about form validations.
- To learn and practice ADO.Net.
- To implement about Web Services.
- To learn about C#.

Course Outcomes

1. Understand the concepts of ASP.net
2. Apply the form validations.
3. Analyze ADO.net
4. Evaluate the strengths and weaknesses of web services
5. Create Asp.net application using C#.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Overview of Asp.Net 9 Hours

Overview of ASP.NET framework, Understanding ASP.NET Controls, Applications, Web Servers, installation of IIS. Web forms, web form controls: server controls, client controls, web forms &amp; HTML, Adding controls to a web form, Buttons, Text Box, Labels, Checkbox, Radio Buttons, ListBox, etc., Running a web Applications, creating a multiform web project.

Unit-II: Form Validation 9 Hours

Form Validation: Client side validation, server Side validation, Validation Controls: Required Field, Comparison, Range, Calendar control, Ad rotator Control, Internet Explorer Control. State management- View state, Session state, Application state.

Unit-III: Introduction to ADO.Net 9 Hours

Architecture of ADO.NET, Connected and Disconnected Database, Create Connection Using ADO.NET Object Model, Connection Class, Command Class, Data Adapter Class, Dataset Class. Display data on data bound Controls and Data Grid. Database Accessing on web applications: Data Binding concept with web, creating data grid, Binding standard web server controls. Display data on web form using Data bound control.

Unit-IV: Web Services **9 Hours**

Writing datasets to XML, reading datasets with XML. Web services: Introduction, Remote methodcall using SOAP, web service description language, building &amp; consuming web service, Web Application deployment.

Unit-V: Overview of C# **9 Hours**

Overview of C#, C# and .NET, similarities &amp; differences from JAVA, Structure of C# program Language features: Type system, boxing and unboxing, flow controls, classes, interfaces, Serialization, Delegates, and Reflection.

Total Hours: 45

Reference Books:

1. Chris Ullman, John Kauffman, Beginning ASP.NET 1.1 with VB.NET 2003, Wrox Publication 2. ADO.NET Professional, Wrox Publication
2. Alex Homer, Dave Sussman, Professional ASP.NET 1.1, Wrox Publication
3. .NET Framework, OREILY Publication.
4. Crouch, ASP.NET and VB.NET Web Programming, Pearson Education
5. Richard Blair, Mathew Renolds, Beginning VB.NET 2003, 3rd edition, Wrox Publication
6. Bill Evjen, Billy, Hollis, et al, Professional VB.NET 2003, 3rd edition, Wrox Publication
7. Deitel and Deitel, Visual Basic.NET How to Program, Pearson Education,2nd edition Greg Buczek, ASP.NET Developer’s Guide, Tata McGraw-Hill, 2002.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://www.udemy.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III L-3 T-1 P-0 C-4

25MCA340T: Software Engineering

Course Objectives

- Knowledge of basic SW engineering methods and practices, and their appropriate application.
- Describe software engineering layer technology and Process framework.
- A general understanding of software process models such as the waterfall and evolutionary models.
- Understanding of software requirements and the SRS documents.
- Understanding of the role of project management including planning, scheduling, risk management, etc.

Course Outcomes

1. Understand the importance of system planning and initial investigation in project success.
2. Apply the principles of software process models to select an appropriate model for a given project scenario.
3. Analyze the implications of the make/buy decision on project cost, schedule, and quality.
4. Evaluate the effectiveness of different software quality assurance techniques in improving product quality.
5. Create a project management plan for an Object-Oriented Software Engineering project, including the selection and use of appropriate CASE tools.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation isindicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

UNIT I System concepts and Information system environment 12 Hours

The system concept, characteristics of system, elements of system, The System Development Life Cycle, The Role of System Analyst. Introduction system planning & initial investigation, various information gathering tools feasibility study concretions & structures tools of system analysis, various methods of process design, form design methodologies, introduction to information system testing, quality assurance security & diastruct computer various (deleting recovery)

UNIT II: Software Process, Product and Project: 12 Hours

The Product: Software, Software Myths, The process: Software Engineering: A Layered Technology, Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Software Process Models, Component – Based Development, Fourth Generation Techniques, Software process and Project Metrics: Software measurement

UNIT III: Software Project Planning and Design: 12Hours

Software Project Planning: Project planning objectives, Decomposition Techniques, Empirical estimation models, The Make/Buy Decision., Risk analysis. Software Design: Design Principles, Cohesion & Coupling, Design notation and specification, structure design methodology.

UNIT IV: Software Quality Assurance and Testing: 12 Hours

Software Quality Assurance: Quality Concepts, The Quality Movement, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, Mistake Proofing for Software, Introduction to ISO standard.

Testing Strategies: A strategic approach of software testing strategic issues, unit testing, integration testing, validation testing, system testing, the art of debugging. OOA, OOD.

UNIT V: Advanced Topics: 12 Hours

MIS & DSS: Introduction to MIS, long range planning, development and implementation of an MIS, applications of MIS in manufacturing sector and in service sector.

Decision Support System concepts, types of DSS. Object Oriented Software Engineering: Object Oriented Concepts, Identifying the Elements of an Object Model, Management of Object-Oriented Software Projects. CASE tools, Re-engineering.

Total: 60 Hours

Reference Books:

1. R. S. Pressman, “Software Engineering – A practitioner’s approach”, 6th ed., McGraw Hill Int. Ed., 2002.
2. Pankaj Jalote “Software Engg” Narosa Publications.
3. Ian Sommerville: Software Engineering 6/e (Addison-Wesley)
4. Richard Fairley: Software Engineering Concepts (TMH)
5. Elis Awad, "System Analysis & Design", Galgotia publications
6. W.S. Jawadekar: Management Information Systems, TMH Publication, India
7. Hoffer “Modern System Analysis & Design” 3e, Pearson Edition

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA
Semester-III

L-2 T-1 P-0 C-3

25MCA350T: Data Science

Course Objectives

- To learn basics of python programming
- To learn object-oriented concepts of python
- To learn numpy basics
- To learn pandas module
- To learn Data Cleaning and Preparation

Course Outcomes

1. Understand the significance of Python in the field of data science .
2. Apply file manipulations and utilize file and directory-related methods in Python.
3. Analyze mathematical and statistical methods provided by NumPy for data analysis and computation.
4. Evaluate and summarize descriptive statistics using pandas for data summarization and analysis.
5. Create strategies for handling missing data effectively within pandas data structures.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I Foundations of Data Science with Python 9Hours

Introduction to Data Science - Why Python? - Essential Python libraries - Python Introduction-Features, Identifiers, Reserved words, Indentation, Comments, Built-in Data types and their Methods: Strings, List, Tuples, Dictionary, Set - Type Conversion- Operators. Decision Making- Looping- Loop Control statement- Math and Random number functions. User defined functions - function arguments & its types.

Unit-II User-Defined Modules, Packages, and OOPs Concepts 9Hours

User defined Modules and Packages in Python- Files: File manipulations, File and Directory related methods, Python Exception Handling.

OOPs Concepts -Class and Objects, Constructors – Data hiding- Data Abstraction- Inheritance.

Unit-III Mastering NumPy: Fundamentals and Advanced Techniques 9 Hours

NumPy Basics: Arrays and Vectorized Computation- The NumPy ndarray- Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicing - Boolean Indexing-Transposing Arrays and Swapping Axes.

Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical Methods-Sorting- Unique and Other Set Logic.

Unit-IV Unveiling Data Structures and Essential Operations 9 Hours

Introduction to pandas Data Structures: Series, DataFrame, Essential Functionality: Dropping Entries- Indexing, Selection, and Filtering- Function Application and Mapping- Sorting and Ranking. Summarizing and Computing Descriptive Statistics- Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.

Unit-V Refining Data: Cleaning, Transforming, and Visualizing with pandas 9 Hours

Data Cleaning and Preparation: Handling Missing Data - Data Transformation: Removing Duplicates, Transforming Data Using a Function or Mapping, Replacing Values, Detecting and Filtering Outliers- String Manipulation: Vectorized String Functions in pandas.

Plotting with pandas: Line Plots, Bar Plots, Histograms and Density Plots, Scatter or Point Plots.

Total: 45 Hours

Reference Books

1. Y. Daniel Liang, "Introduction to Programming using Python", Pearson,2012.
2. Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", O'Reilly, 2nd Edition,2018.
3. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data", O'Reilly, 2017.
4. Wesley J. Chun, "Core Python Programming", Prentice Hall,2006.
5. Mark Lutz, "Learning Python", O'Reilly, 4th Edition, 2009.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III

L-3 T-1 P-0 C-4

25MCA215T: Data WareHousing and Mining

Course Objectives:

- To learn the basic concepts, principles, and techniques in data mining.
- To know most of the classical data mining algorithms.
- To learn to perform a systematic analysis of real-world data mining problems end to end.
- To know the data mining problems and evaluate, visualize and communicate statistical models.
- To know the use of Data Mining Tools for data analysis to maintain a data warehouse.

Course Outcomes:

1. Understand the functionality of the various data mining and data warehousing components.
2. Apply the concept of data mining components and techniques in designing data mining systems.
3. Analyze data warehousing and data mining tools for different applications.
4. Implement different approaches to data warehousing and data mining with various technologies.
5. Create various data mining and data warehousing models.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PSO-1	PSO-2
CO-1	1	3	2	-	-	-	-	-	-	-	-	-
CO-2	-	1	3	2	-	-	1	-	-	-	-	-
CO-3	-	1	2	3	2	1	-	1	-	-	-	-
CO-4	-	1	-	2	3			-	-	1	2	1
CO-5	2	-	1	-	2	3	-	-	1	-	1	2

High-3 Medium-2 Low-1

UNIT – I: Data Warehouse 12 Hours

Introduction to Data warehouse, Differences between operational database systems and data warehouse, Data Ware House characteristics, Data Ware House Architecture and its components, Extraction-Transformation-Loading, Logical (Multi-Dimensional), Data Modeling, Schema Design, star and snow-flake Schema, Fact Constellation, Fact Table, Fully Addictive, Semi-Addictive, Non-Addictive Measures; Fact- Less-Facts, Dimension Table characteristics; Fact-Less-Facts, Dimension Table characteristics; OLAP cube, OLAP Operations, OLAP Server Architecture-ROLAP, MOLAP and HOLAP.

UNIT -II: Introduction to Data Mining 12 Hours

Introduction to Data Mining: Introduction, what is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing- Data Cleaning, Missing Data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binarization, Data Transformation; Measures of similarity and dissimilarity-Basics.

UNIT – III: Association Rules **12 Hours**

Problem Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation, APRIORI Algorithm, The Partition Algorithms, FP-Growth Algorithms, Compact Representation of Frequent Item Set-Maximal Frequent Item Set, Closed Frequent Item Set.

UNIT -IV: Classification **12 Hours**

Problem definition, General Approaches to solving a classification problem, Evaluation of Classifiers, Classification techniques, Decision Trees-Decision Tree Construction, Methods for expressing attribute test conditions, Measures for Selecting the Best split, Algorithm for Decision tree Induction, Naïve-Bayes Classifier, Bayesian Belief Networks; K-nearest neighbor classification-Algorithm and characteristics.

UNIT – V: Clustering **12 Hours**

Problem Definition, Clustering overview, Evaluation of clustering algorithms, Partitioning clustering K-Means Algorithm, K-Means Additional Issues, PAM Algorithm, Hierarchical Clustering-Algorithm- Agglomerative Methods and Divisive Methods, Basic Agglomerative Hierarchical Clustering Algorithm, Specific techniques, Key Issues in Hierarchical Clustering, Strengths and weakness, Outlier Detection.

Total Hours: 60

Text Book:

1. Data Mining-Concepts and Techniques- Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, Elsevier, 2 Edition, 2006.
2. Introduction to Data Mining, Pang-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.

References Books:

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.
2. Data Ware Housing Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.
3. The Data Ware House Life Cycle Toolkit- Ralph Kimball, Wiley Student Edition.
4. Data Mining, Vikaram Pudi, P Radha Krishna, Oxford University.

List of E-Learning Resources:

1. https://www.nptel.com
2. https://www.coursera.com
3. https://www.javatpoint.com
4. https://www.simplilearn.com

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III

L-3 T-1 P-0 C-4

25MCA216T: IT Infrastructure and Data Canter

Course Objectives:

- Knowledge in setting-up and Services provided by data centers.
- Datacenter fundamentals
- students to understand the basic concepts of Datacenter architecture
- Network infrastructure in a Datacenter, server frames fault tolerance, Datacenter availability, network implementation and disaster recovery.
- To learn about Business Continuity (BC) Infrastructure Services.

Course Outcomes:

1. Understand the concept of datacenters and their role in enterprise and service provider environments.
2. Apply safety measures against natural and man-made disasters.
3. Analyze guidelines for planning and structuring data center facilities.
4. Evaluate the components and segments of a data center architecture.
5. Create the BC technology solutions and backup and recovery considerations.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	-	-	-	-	-	1	-	2	-
CO2	-	-	-	-	1	-	-	1	-	-
CO3	-	2	-	2	-		1	-	-	-
CO4	-	-	1	-	2	2	-	-	-	1
CO5	-	-	-	2	1	-	-	-	-	-

High-3 Medium-2 Low-1

Unit-I Foundations of Datacenters and Application Architecture 12 Hours

Datacenters Defined, Datacenter Goals, Datacenter Facilities, Roles Datacenters in the Enterprise, Roles of Datacenters in the Service Provider Environment, Application Architecture Models. The Client/Server Model and Its Evolution, n-Tier Model, Multitier Architecture Application Environment, Data Center Architecture.

Unit-II Essentials of Data Center Infrastructure Planning 12 Hours

Data Center Prerequisites, Required Physical Area for Equipment and Unoccupied Space, Required Power to Run All the Devices, Required Cooling and HVAC, Required Weight, Required Network Bandwidth, Budget Constraints, Selecting a Geographic Location, Safe from

Natural Hazards, Safe from Man-Made Disasters, Availability of Local Technical Talent, Abundant and Inexpensive Utilities Such as Power and Water, Selecting an Existing Building (Retrofitting), tier standard.

Unit-III Designing a Robust Data Center Infrastructure 12 Hours

Characteristics of an Outstanding Design, Guidelines for Planning a Data Center, Data Center Structures, No-Raised or Raised Floor, Aisles, Ramp, Compulsory Local Building Codes, Raised Floor Design and Deployment, Plenum, Floor Tiles, Equipment Weight and Tile Strength, Electrical Wireways, Cable Trays, Design and Plan against Vandalism

Unit-IV Exploring Diverse Data Center Architectures and Service Models 12 Hours

Types of server farms and data center, internet server farm, intranet server farm, extranet server farm, internet datacenter, corporate datacenter, software defined datacenter, datacenter topologies, Aggregation Layer, Access Layer, Front-End Segment, Application Segment, Back-End Segment, Storage Layer, Data Center Transport Layer, Data Center Services, IP infrastructure Services, Application Services, Security Services, Storage Services

Unit-V Strategies for Business Continuity 12 Hours

Business continuance infrastructure services, the need for redundancy, Information availability, BC terminology, BC planning life cycle, BC technology solutions, backup and recovery considerations, backup technologies, Uses of local replicas, Local replication technologies, Restore and restart considerations, Modes of remote replications, remote replication technologies

Total Hours 60

Reference Books:

1. IP Storage Networking by: Gary Oreinstein, Addison Wesley Professional, 2006
2. Information Storage and Management, G. Somasundaram – Alok Srivastava, Wiley; 1 edition
(April 6, 2009)
3. Administering Data-Centers, Kailash Jayswal, Wiley; 1 edition (November 28, 2005)

List of e-Learning Resources:

1.https://nptel.ac.in/
2.https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III

L-3 T-1 P-0 C-4

25MCA217T: Internet of Things

Course Objectives:

- To understand the basic concepts and Characteristics of IoT.
- To learn design principles for web connectivity.
- To understand sensor technology.
- To learn IOT design methodology.
- To learn how to develop IoT solutions

Course Outcomes:

- Understand the foundational concepts of IoT.
- Apply design principles for web connectivity and internet communication protocols.
- Analyze sensor technologies, participatory sensing, and communication protocols for sensor data.
- Implement IoT design methodologies, address privacy and security concerns.
- Develop IoT solutions using tools like Arduino and Raspberry Pi.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PSO-1	PSO-2
CO-1	1	3	2	-	-	-	-	-	-	-	-	-
CO-2	-	1	3	2	-	-	1	-	-	-	-	-
CO-3	-	1	2	3	2	1	-	1	-	-	-	-
CO-4	-	1	-	2	3	-	-	-	-	1	2	1
CO-5	2	-	1	-	2	3	-	-	1	-	1	2

High-3 Medium-2 Low-1

Unit-I: Introduction 12 Hours

Introduction: Definition, Characteristics of IOT, IOT Conceptual framework, IOT Architectural view, Physical design of IOT, Logical design of IOT, Application of IOT. Machine-to-machine (M2M), SDN (software-defined networking) and NFV(network function virtualization) for IOT, data storage in IOT, IOT Cloud Based Services.

Unit-II: Design Principles for Web Connectivity 12 Hours

Design Principles for Web Connectivity: Web Communication Protocols for connected devices, Message Communication Protocols for connected devices, SOAP, REST, HTTP Restful and Web Sockets. Internet Connectivity Principles: Internet Connectivity, Internet based communication, IP addressing in IOT, Media Access control.

Unit-III: Sensor Technology 12 Hours

Sensor Technology, Participatory Sensing, Industrial IOT and Automotive IoT, Actuator, Sensor data Communication Protocols, Radio Frequency Identification Technology, Wireless Sensor Network Technology.

Unit-IV: IOT Design Methodology 12 Hours

IOT Design methodology: Specification -Requirement, process, model, service, functional & operational view. IoT Privacy and security solutions, Raspberry Pi & Arduino devices. IOT Case studies: smart city streetlights control & monitoring.

Unit-V: Developing IoT solutions 12 Hours

Developing IoT solutions: Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Privacy and Security Issues in IoT.

Total Hours: 60

Reference Book:

1. Rajkamal,"Internet of Things", Tata McGraw Hill publication
2. Vijay Madisetti and ArshdeepBahga, "Internet of things(A-Hand-on-Approach)" 1st Edition ,Universal Press
3. HakimaChaouchi "The Internet of Things: Connecting Objects", Wiley publication.
4. Charless Bell "MySQL for the Internet of things", Apress publications.
5. Francis dacosta "Rethinking the Internet of things:A scalable Approach to connecting everything", 1st edition, Apress publications 2013.
6. Donald Norris"The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and BeagleBone Black", McGraw Hill publication.
7. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014. Syllabus for Bachelor of Technology Computer Engineering
8. Cuno Pfister, Getting Started with the Internet of Things, O‟Reilly Media, 2011, ISBN: 978-1-4493- 9357-1

List of E-Learning Resources:

1. https://www.nptel.com
2. https://www.coursera.com
3. https://www.javatpoint.com
4. https://www.simplilearn.com

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III

L-0 T-0 P-2 C-1

25MCA400P:ASP.Net with C#

Course Objectives

- To learn basics of ASP.Net
- To learn about form validations.
- To learn and practice ADO.Net.
- To implement about Web Services.
- To learn about C#.

Course Outcomes

1. Understand dynamic user interfaces using ASP.NET controls.
2. Apply input validation and algorithms using ASP.NET tools.
3. Analyze database CRUD operations with ADO.NET and SQL Server.
4. Evaluate custom user controls for input handling and validation.
5. Create Asp web applications with robust form handling and backend integration.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit 1: Working with Controls in ASP.NET

Basic Controls and Event Handling, ListBox, Button, Image, and Label controls, Advanced Controls and Data Interaction, Adding Labels, TextBox, and Button for quantity input.

Unit 2: User Input and Validation

Input Validation Controls, Using CompareValidator for validating user input, Custom User Controls, Creating and validating user controls for login.

Unit 3: Database Integration with ASP.NET

Connecting to Databases, Establishing and testing SQL Server connections, Performing CRUD Operations, Inserting, updating, and deleting records in a database.

Unit 4: User Controls and Advanced Validation

Creating and Using User Controls, Validating user credentials with a user control, Advanced Input Validation, Validating complex user input such as age, email, and user ID.

Unit 5: ASP.NET Form Handling and Dynamic Content

Form Handling and Validation, Creating forms to collect and validate user input, Dynamic Content Display, Using controls to dynamically display content based on user actions.

List of Experiments

1. Write a program containing the following controls:

a) A ListBox
b) A Button
c) An Image
d) A Label

2. The list box is used to list items available in a store. When the user clicks on an item in the listbox, its image is displayed in the image control. When the user clicks the Button, the cost of the selected item is displayed in the control.

3. Extend the above program to add the following controls:

a) Two labels
b) A TextBox
c) A Button

one of the labels is displayed adjacent to the textbox , displaying the message “Enter the quantity:”.When the user enters the quantity in the textbox and clicks the button,
the total cost is evaluated and displayed in another label.

4. Write a program to get a user input such as the boiling point of water and test it to the appropriate value using CompareValidator.

5. Write a program to connect to the database in SQL Server, in the Page Load event. When the connection is established, the message “Connection has been established” should be displayed in a label in the form.

6. Write a program that updates the ename field of the emp table with the given name, where eno=102.

7. Write a program to display the records from the database as shown in the figure:

CustomerID	**CustomerName**
121	Baskar
122	Partha
123	Suresh
124	Vidya

8. Create a user control that receives the user name and password from the user and validates them. If the user name is “BCAV” and the password is “asp.net” then the user is authorized, otherwise not.

9. Database programs with ASP.NET and ADO.NET Create a web application to insert 3 records inside the SQL database table having following fields (DeptId, DeptName, EmpName, Salary). Update the salary for any one employee and increment it to 15% of the present salary. Perform delete operation on 1 row of the database table.

10. Create the application that accepts name, password , age , email id, and user id. All the information entry is compulsory. Password should be reconfirmed. Age should be within 21 to 30. Email id should be valid. User id should have at least a capital letter and digit as well as length should be between 7 and 20 characters.

11. Write programs using conditional statements and loops: Generate patterns (triangles and diamond) with numbers.

12. Write programs using conditional statements and loops: Test for prime numbers.

13. Write a program containing the following controls:

a) A ListBox
b) A Button
c) An Image
d) A Label

The list box is used to list items available in a store. When the user clicks on an item in the list box, its image is displayed in the image control. When the user clicks the button, the cost of the selected item is displayed in the control.

14. Write a program that gets user input such as the user name, mode of payment, appropriate credit card. After the user enters the appropriate values the Validation button validates the values entered.

15. Write programs using conditional statements and loops: Generate Fibonacci series

Total Hours 30

Reference Books:

1. Chris Ullman, John Kauffman, Beginning ASP.NET 1.1 with VB.NET 2003, Wrox Publication 2. ADO.NET Professional, Wrox Publication
2. Alex Homer, Dave Sussman, Professional ASP.NET 1.1, Wrox Publication
3. .NET Framework, OREILY Publication.
4. Crouch, ASP.NET and VB.NET Web Programming, Pearson Education
5. Richard Blair, Mathew Renolds, Beginning VB.NET 2003, 3rd edition, Wrox Publication
6. Bill Evjen, Billy, Hollis, et al, Professional VB.NET 2003, 3rd edition, Wrox Publication
7. Deitel and Deitel, Visual Basic.NET How to Program, Pearson Education,2nd edition Greg Buczek, ASP.NET Developer's Guide, Tata McGraw-Hill, 2002.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://www.udemy.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA
Semester-III

L-0 T-0 P-2 C-1

25MCA350P: Data Science

Course Objectives

- To learn basics of python programming
- To learn object-oriented concepts of python
- To learn numpy basics
- To learn pandas module
- To learn Data Cleaning and Preparation

Course Outcomes

1. Understand Python data types and operations to solve diverse problems.
2. Apply mathematical, statistical, and file I/O techniques in Python.
3. Analyze Python functions, modules, classes, and handle exceptions for robust programming.
4. Evaluate and process data using NumPy and Pandas for data analysis.
5. Create preprocess data in Pandas, handling missing data and performing advanced data manipulations.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	-	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	-	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit 1: Python Data Types and Operations

Introduction to Python Built-in Data Types, Strings, Lists, Tuples, Dictionaries, Sets, Numerical Operations in Python, Math functions, Random number generation, Decision making statements, Looping statements.

Unit 2: Functions and Modules

User-Defined Functions, Function definition and arguments, Modules and Packages, Creating and importing modules and packages, File Manipulations, File operations: open, close, read, write, append, copy.

Unit 3: Object-Oriented Programming (OOP) in Python

Class and Object Creation, Class declarations, Object creation and manipulation, OOP Concepts, Data hiding and abstraction, Exception Handling, Handling built-in exceptions.

Unit 4: NumPy for Numerical Computing

Introduction to NumPy,Creating NumPy arrays from various data structures, Array Manipulation, Indexing, slicing, reshaping, joining, and splitting arrays, Computations with NumPy, Using universal functions and mathematical methods.

Unit 5: Data Analysis with Pandas

Pandas Basics, Creating Series and DataFrames, DataFrame Operations, Importing CSV files, Data Manipulation and Analysis, Selecting, deleting, ranking, sorting, and statistical operations, Advanced Data Handling, Counting and uniqueness of categorical values, Renaming columns.

List of Experiments

1. python built-in data types: Strings, List, Tuples, Dictionary, Set and their methods to solve any given problem
2. Handle numerical operations using math and random number functions
3. Create user-defined functions with different types of function arguments.
4. Create packages and import modules from packages.
5. Perform File manipulations- open, close, read, write, append and copy from one file to another.
6. Handle Exceptions using Python Built-in Exceptions
7. Solve problems using Class declarationImplement basic Python programs for reading input from the console.
8. Perform Creation, indexing, slicing, concatenation and repetition operations on Python built-in data types: Strings, List, Tuples, Dictionary, Set
9. Solve problems using decision and looping statements.
10. Apply Py and Object creation.
11. Implement OOP concepts like Data hiding and Data Abstraction.
12. Create NumPy arrays from Python Data Structures, Intrinsic NumPy objects and Random Functions.
13. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining and Splitting.
14. Computation on NumPy arrays using Universal Functions and Mathematical methods.
15. Import a CSV file and perform various Statistical and Comparison operations on rows/columns.
16. Load an image file and do crop and flip operation using NumPy Indexing.
17. Create Pandas Series and DataFrame from various inputs.
18. Import any CSV file to Pandas DataFrame and perform the following:
19. Visualize the first and last 10 records
20. Get the shape, index and column details
21. Select/Delete the records(rows)/columns based on conditions.
22. Perform ranking and sorting operations.
23. Do required statistical operations on the given columns.
24. Find the count and uniqueness of the given categorical values.
25. Rename single/multiple columns.

Total: 30 Hours

Reference Books

1. Y. Daniel Liang, “Introduction to Programming using Python”, Pearson,2012.
2. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython”, O’Reilly, 2nd Edition,2018.
3. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.
4. Wesley J. Chun, “Core Python Programming”, Prentice Hall,2006.
5. Mark Lutz, “Learning Python”, O’Reilly, 4th Edition, 2009.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III

L-0 T-0 P-8 C-4

25MCA410P: Server Side Scripting Lab

Course Objectives

- To learn the fundamental principles and architecture of Node.js.
- To learn Utilize npm (Node Package Manager) .
- To Learn how to serve static files and work with templating engines such as Pug and EJS to generate dynamic HTML content in Express.js applications.
- To Learn performance optimization techniques specific to Express.js applications, including caching strategies, code optimization, and server-side rendering, to improve application speed and scalability.
- To Learn the MongoDB Query Language (MQL)

Course Outcomes

1. Understand the concepts of event-driven, non-blocking I/O model in Node.js.
2. Apply the CommonJS module system to organize and modularize code effectively in Node.js applications.
3. Analyze and troubleshoot errors and implement middleware chaining in Express.js applications.
4. Evaluate performance optimization techniques in Express.js applications for scalability and efficiency.

5. Create a MongoDB database instance and configure it according to project requirements.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit 1: Introduction to Node.js

Overview of Node.js and its architecture, Installation and setup of Node.js, Understanding the event-driven, non-blocking I/O model, Basics of JavaScript in the context of Node.js, Variables, data types, operators, control structures, Functions, closures, callbacks.

Unit 2: Advanced Node.js Concepts

Modules and the Common JS module system, Using npm (Node Package Manager) to manage dependencies, Asynchronous programming in Node.js, Callbacks, Promises, Async/Await, Error handling and debugging techniques in Node.js, Working with the Node.js File System (fs) module, Introduction to streams for handling large data sets efficiently

Unit 3: Introduction to Express.js

What is Express.js and why use it, Setting up a basic Express application, Understanding middleware and how to use it in Express ,Routing in Express: handling HTTP requests and responses, Serving static files and working with templates (e.g., Pug, EJS),Error handling and middleware chaining in Express

Unit 4: Advanced Express.js Topics

Authentication and authorization using Passport.js, Express.js best practices and security measures, Using Express.js with databases (MongoDB, MySQL, etc.),Integration of third-party APIs with Express.js, Real-time communication with WebSockets and Socket.IO, Performance optimization techniques in Express.js applications

Unit 5: Introduction to MongoDB

Overview of MongoDB and NoSQL databases, Installation and setup of MongoDB,CRUD operations (Create, Read, Update, Delete) with MongoDB, Schema design and data modeling in MongoDB, Querying MongoDB with the MongoDB Query Language (MQL), Introduction to indexing and aggregation pipelines in MongoDB

List of Experiments

1. Install Node.js on your local machine and verify the installation.
2. Set up a basic Node.js project structure with a package.json file.
3. Create a simple "Hello World" program in Node.js.
4. Explore the event-driven architecture of Node.js by creating event listeners and emitting events.
5. Implement a basic HTTP server using the built-in 'http' module in Node.js.
6. Demonstrate the non-blocking I/O model of Node.js by performing asynchronous file operations.
7. Practice JavaScript basics in Node.js by creating variables, assigning values, and understanding data types.
8. Experiment with different JavaScript operators (+, -, *, /, %, ==, ===, &&, ||) in Node.js.
9. Implement control structures (if-else, switch-case) in Node.js to control program flow.
10. Create and invoke functions in Node.js, understanding scope and function hoisting.
11. Explore closures in JavaScript by defining and invoking nested functions in Node.js.
12. Implement callback functions to handle asynchronous operations in Node.js.
13. Practice error handling techniques in Node.js using try-catch blocks.
14. Debug Node.js applications using built-in debugging tools and the 'debugger' statement.
15. Read and write files asynchronously using the Node.js File System (fs) module.
16. Use streams to handle large data sets efficiently, such as reading from and writing to files or HTTP requests.
17. Install external Node.js modules using npm and manage dependencies in a Node.js project.
18. Create a custom module in Node.js and import/export functionality using module.exports.
19. Demonstrate asynchronous programming in Node.js using callbacks for file I/O operations.
20. Experiment with Promises to handle asynchronous operations more elegantly in Node.js.
21. Implement asynchronous functions using async/await syntax in Node.js.
22. Practice using the 'util' module to promisify callback-based functions in Node.js.
23. Perform unit testing on Node.js modules using popular testing frameworks like Mocha and Chai.

24. Set up a basic Express.js application and define routes for handling HTTP requests.
25. Use middleware functions in Express.js to modify request and response objects.
26. Serve static files (HTML, CSS, JavaScript) using Express.js middleware.
27. Implement error handling middleware in Express.js to handle runtime errors gracefully.
28. Use template engines like Pug (formerly Jade) or EJS to render dynamic content in Express.js.
29. Set up authentication and authorization using Passport.js in an Express.js application.
30. Experiment with third-party API integration in Express.js, such as fetching data from a RESTful API and displaying it in a web application.

Total Hours:60

Reference Books:

1. "Node.js Design Patterns" by Mario Casciaro

2. "Node.js 10.x Blueprints: Create Stunning Web and Server-Side Applications Using the Latest Features of Node.js, 3rd Edition" by Fernando Monteiro
3. "Express.js Guide: The Comprehensive Book on Express.js" by S. H. Khan
4. "Node.js Web Development: Server-side Development with Node 10 made easy, 5th Edition" by David Herron
5. "The Little MongoDB Book" by Karl Seguin

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

MCA

Semester-III

L-0 T-0 P-8 C-4

25MCA200P: Minor Project

Course Objectives

- To solve industrial (or society or research) problems.
- To plan, schedule, and monitor the software project.
- To learn Development, coding, and testing of a large project cohesively.
- To learn Documentation of project.

Course Outcomes

1. Understand fundamental concepts and principles relevant to their project domain.
2. Apply appropriate methodologies and technologies to design and develop a software solution or system addressing a real-world problem or need.
3. Analyze the effectiveness, efficiency, and quality of their software solution through testing, debugging, and optimization processes.
4. Evaluate their knowledge, skills, and experiences to innovate and propose enhancements or extensions to their project, demonstrating creativity and problem-solving abilities.
5. Develop solutions for contemporary problems using modern tools for sustainable development.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

GUIDELINES FOR MCA MINOR PROJECT

1. Every student, (in a group of maximum two) will be asked to select a particular project listed by the department, on which he/she will have to develop a working module in semesters.
2. The students are expected to work on real-life project. However, it is not mandatory for a student to work on a real-life project.
3. No student will be allowed to change the topic of the project once allotted.
4. Not more than two students are permitted to work on a project.
5. The student can formulate a project problem with the help of her/his supervisor and if approved, the students commence working on it.
6. A candidate is required to present the progress of the project work during the

semester as per the schedule.

PROJECT SYNOPSIS FORMAT

The project proposal should be prepared and approved in consultation with supervisor. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. The project proposal should contain complete details in the following form:

1. Title of the project
2. Name of the supervisor (external supervisor(company)from / internal supervisor (teacher of the MCA)
3. Introduction and objectives of the project
4. Analysis (DFD, ER diagrams, class diagrams, time line etc. As per the project requirements).
5. A complete structure which includes:
    - Name of modules and their description
    - Database / data structures description
    - Process logic of each module (flow chart)
    - Reports generation. (repot format)
6. Tools / platform, hardware and software requirement specifications
7. Organization/ company details with profile of supervisor (if project is carried out outside the department)

PROJECT REPORT FORMULATION

Good quality white executive bond paper A4 size should be used for typing and duplication. Care should be taken to avoid smudging while duplicating the copies.

Page specification: leftmargin-3.0cms, right margin- 2.0 cm, top margin 2.54 cm, bottom margin 2.54 cm, line spacing – single, font size – 12 for normal text, 14 for headings, 16 for chapter heading, page numbers - all text pages as well as program source code listing should be numbered at the bottom of the pages. Employ MS-Word or open-source software.

The project report should contain the following:

1. Front page – black color with golden or white text.
2. Certificate from the supervisor with her/his signature and date.
3. Certificate of originality/ self-certificate

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**