Faculty of Science

Shree Ramkrishna Institute of Computer Education & Applied Sciences, Surat

B.Sc. (Computer Science)

SEMESTER- 4

Web Application Development - II (ASP.Net)

Course Code	
Course Title	**Major: Web Application Development - II (ASP.Net)**
Credit	3 (Theory) + 1 (Practical)
Teaching per Week	3 Hrs (Theory) + 2 Hrs (Practical)
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays, etc.)
Last Review / Revision	2021-22
Purpose of Course	The course aims to make students capable of developing basic web applications using the latest tools and technologies of C#.Net.
Course Objective	To provide an in-depth knowledge of the most recent server-side programming technology.
Pre-requisite	Basic understanding of Web, HTTP, HTML, JavaScript
Course Outcome	After completing this course, the student will be capable of developing basic web applications using the latest tools and technologies of C#.Net.
Course Content	**Unit-1: Overview of .net framework** **[5 Hrs]** 1.1 .Net framework & its benefits 1.2 Common Language Runtime (CLR) 1.2.1 Common Type System (CTS) 1.2.2 Common Language Specification (CLS) 1.2.3 Garbage Collector (GC) 1.2.4 Microsoft Intermediate Language (MISL) 1.3 ASP.NET - Event Driven Programming 1.4. Files & Directories 1.5 Page Lifecycle 1.6 Concept of Postback – Page.IsPostBack 1.7 Error Handling Concepts – Application Level, Page Level, Code Level **Unit 2: Client Server Communication & Application Management** **[5 Hrs]** 2.1 Communications with Web Browser 2.2. Response Object – Response.Write(), Response.Redirect() 2.3. State Management 2.3.1 Client-Side State Management – Cookies, QueryString ViewState, HiddenField 2.3.2 Server-Side State Management – Session, Application 2.4 Web.Config File 2.5 Global.asax File **Unit 3: Web Server Control** **[7 Hrs]** 3.1 Basic Web Server Controls 3.2 Validation Controls 3.3 Navigation Control-Menu 3.4 Master - Content Page **Unit-4: Data Access objects** **[8 Hrs]** 4.1 The Server Explorer 4.2 Connection Object – Connection string using Web.Config

	4.3 ADO.NET Architecture- Data provider, Data Adapter, DataReader 4.4 Command objects – ExecuteNonQuery(), ExecuteScalar(), ExecuteReader() 4.5 Disconnected Architecture – Dataset **Unit 5: Data access controls and operations** **[8 Hrs]** 5.1 Data binding with controls – DropDown, TextBox, Image 5.2 Basic CRUD Operations 5.3 Rich Data Controls - Grid View, Data list, Repeater **Unit 6: ASP.Net MVC Application** **[7 Hrs]** 6.1 Introduction to ASP.Net MVC Framework – Model, View, and Controller 6.2 Web Page Routing 6.3 Data Access and Modeling 6.4 View-ViewBag and ViewData 6.5 CRUD operation in MVC - Entity Data Model
Practical	List of practical will be prepared at the beginning of each semester
Text and Reference Literature	1. Pro ASP.NET 4 in C# 2010– Matthew MacDonald – Apress 2. Learning ASP.NET 3.5 – O'Reilly 3. ASP.Net 4.0 Black Book – dreamtech press 4. ASP.NET 4.0 Unleashed – Stephen Walther – Sams 5. Professional ASP.NET 3.5: In C# and VB (Programmer to Programmer)– by Bill Evjen – Wrox 6. Essential Windows Communication Foundation (WCF): For .Net Framework 3.5 - Steve Resnick – Pearson 7. Beginning ASP.NET 4.0 in C# and VB by Imar Spaanjaars Wrox Pubs. 8. Pro Asp.NET MVC 5 Adam Freeman - Apress **Web References for Practical:** https://www.aspsnippets.com/ https://www.c-sharpcorner.com/ http://www.tutorialspoint.com for ASP.NET http://www.asp.net
Teaching Methodology	Discussion, Independent Study, Seminars and Assignment, Internal Project Development Practical Assignments 80% (Approximately weekly)
Evaluation Method	50% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 50% assessment is based on end semester written examination

Data Structures and Algorithms

Course Code	
Course Title	**Major: Data Structures and Algorithms**
Credit	3(Theory) + 1(Practical)
Teaching per Week	3 Hrs (Theory)+2Hrs (Practical)
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Last Review / Revision	2021-22
Purpose of Course	This course imparts the knowledge of Data Structure. The concepts of Primitive and non-primitive data structures are covered in this course. It covers concepts of Arrays, Stack, Queue, Link list and sorting searching methods. The course is aimed to give inner depth of non-primitive data structures and its related applications.
Course Objective	To make students understand concepts of Primitive and non-primitive Data structure. To make students understand concepts of stack, queue and types of queues. To make students understand concepts of Link-list and related applications. To make students work with searching and sorting techniques.
Pre-requisite	Students are expected to have understanding of computer programming
Course Out come	At the end of the course, students are expected to have clear concepts about the primitive and non-primitive data structures
Course Content	**Unit 1: Introduction to Data Structure** **[5 Hrs]** 1.1 Concept of data, their characteristics, significance and digital storage mechanism, Concept of Data Structure 1.2 Types of data structures 1.2.1 Primitive Data Structures- integer, character, boolean 1.2.2 Non primitive Data structures- Linear Data Structures (Strings, Arrays, Records, Stack, Queue, Linked List), Non-linear Data Structures (Trees, Graphs) 1.3 Need and Applications of Data structure 1.4 Static and Dynamic memory allocation **Unit 2: Analysis of Algorithm** **[8 Hrs]** 2.1 Algorithm & Flowchart 2.2 Algorithm performance 2.3 Analysis criteria (Time / Space) 2.4 Concept of Algorithm analysis 2.5 Average case / Best Case / Worst case Scenarios **Unit 3: Searching & Sorting** **[8 Hrs]** 3.1 Searching in large amount of data 3.2 Linear search 3.3 Binary search 3.4 Sorting - Internal and external sort 3.5 Bubble sort 3.6 Selection sort 3.7 Insertion sort

	Unit 4: Stack **[8 Hrs]** 4.1 Stack data structure - memory organization and implementation 4.2 Operations on stack (Push, Pop, Peep, Display) 4.3 Applications of stack- Function calls, Recursion, Polish Notation (Evaluation to postfix expression) **Unit 5: Queue** **[8 Hrs]** 5.1 Simple Queue data structure-memory organization and implementation 5.2 Operations with Queue (insert, delete and view) 5.3 Circular queue(insert,delete and view) 5.4 Introduction to Dequeue and Priority queue 5.5 Applications of queue 5.5.1 Printer queue simulation, 5.5.2 Round robin algorithm simulation **Unit 6: Linked list** **[8 Hrs]** 6.1 Linked list - representation, advantages and disadvantages 6.2 Various operations on one way (singly) linked list[insert, delete, view] 6.3 Various operations on two way (doubly) linked list[insert,delete,view] 6.4 Applications of Linked list-Polynomial manipulation, Linked Dictionary
Practical	List of practical will be prepared at the beginning of each semester
Text and Reference Literature	**1.** An Introduction to Data Structure with Applications: Trembley & Sorenson – McGraw Hill **2.** Data Structures & Algorithms in Java , second edition:Robert Lafore SAMS PEARSON **3.** Data Structures and Algorithms in Java, 3rd edition: Goodrich and TamassiaJohn Wiley and Sons, Inc. 2004. ISBN 0-471-46983-1. **4.** Algorithms, 4th Edition: Robert Sedgewick and Kevin Wayne **5.** Fundamentals of Data Structures, Horwith E and Sahni S, Universities Press **6.** Data Structure and Algorithms, Aho A.V., Hopcroft and Ullman **7.** Data Structures and Algorithms in Java, 6th Edition Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser [ISBN: 978-1-118-77133-4] **8.** Data Structures and Algorithms Made Easy in Java, Narasimha Karumanchi [ISBN: 978-8192107554]
Teaching Methodology	The course is composed of readings, discussion, videos and presentation, code competition activity and assignments of computational problem solving. Practical Assignments 80% (Approximately weekly)
Evaluation Method	50% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination, etc. 50% assessment is based on the end semester written examination.

Python Programming

Course Code	
Course Title	**Major: Python Programming**
Credit	3(Theory)+1(Practical)
Teaching per Week	3 Hrs (Theory)+ 2 Hrs (Practical)
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Last Review / Revision	2021-22
Purpose of Course	The purpose of the course is to make students capable of implementing concepts, methods and tools related to python programming and basic data science operations.
Course Objective	This subject aims to cover the python language programming with emphasis on various python data structures and various libraries like Pandas, NumPy, Matplotlib for performing various data science operations.
Pre-requisite	Basic concepts of Programming
Course Out come	After completion of this course, the student will be capable of performing application-level programming using python language with basic data science operations.
Course Content	**Unit 1: Introduction to Python** **[5Hrs]** 1.1 Python History and Usability 1.1.1 Application areas of Python 1.1.2 Technical Strengths of Python 1.2 Program Execution in Python - Program Execution, Python Virtual Machine (PVM) 1.3 IDLE of Python, Jupyter Notebook **Unit 2: Python Programming Basics** **[5 Hrs]** 2.1 Comments, Indentations, Operators 2.2 Assignment, Expressions and Data Types 2.3 Selection Control 2.4 Iterative Control **Unit 3: Python Object Types and Operations** **[9 Hrs]** 3.1 String: Indexing, Slicing, Text Parsing 3.2 List: Indexing, Slicing and Merging List 3.3 Dictionaries: Add, Update, Remove and Sort 3.5 Tuples and Sets **Unit 4: Python Functions & Modules** **[6 Hrs]** 4.1 Function Basics 4.1.1 Definition, Call, Passing Arguments 4.1.2 Lambda Functions 4.2 Function Parameter and Call

	4.2.1 Calling value returning function 4.2.2 Calling non-value returning function 4.2.3 Parameter Passing 4.2.4 Function arguments and variable scope 4.3 Modules 4.3.1 Modules and Name-spaces 4.3.2 Modules import, Load and execution 4.3.3 Top-Down Design 4.3.4 Built-in name-spaces in python **Unit 5: Text Files** **[5 Hrs]** 5.1 Opening, Reading and Writing Text file in python 5.2 File traversal 5.3 String and Text operations 5.4 Reading CSV, XML, JSON files **Unit 6: Python Libraries for Data Science** **[15 Hrs]** 6.1 Numeric Python - NumPy 6.1.1 Introduction to Numpy 6.1.2 Array Operations using Numpy 6.1.3 N-dimensional Array Processing 6.2 Data Analysis - Pandas 6.2.1 Introduction to Pandas 6.2.2 Pandas Objects - Series and Dataframes 6.2.3 Dataframe Operations 6.2.4 Reading and Writing Files 6.3 Plotting Graphs using Matplotlib 6.3.1 Plot Creation 6.3.2 Plot Routines 6.3.3 Saving, Showing and Clearing Graphs 6.3.4 Customize Matplotlib
Practical	List of practical will be prepared at the beginning of each semester
Text and Reference Literature	1. Learning Python -Mark Lutz : O'Reilly Media 2. Core Python Programming – by Wesley J Chun ISBN-13: 978- 0132269933 3. Introduction to Computer Science using Python - A computational problem solving focus - Charles Dierbach, Wiley ISBN-13 978-8126556014 4. Python for Everybody: Exploring Data in Python 3, by Charles Severance (Author), Aimee Andrion (Illustrator), Elliott Hauser (Editor), Sue Blumenberg (Editor) 5. An Introduction to Python - by van Rossum Guido ISBN: 9780954161767, 0954161769 6. Core Python Application Programming – by Wesley J Chun Prentice Hall 7. Introduction to Computer Programming with Python (Harris Wang)
Teaching Methodology	The course is composed of readings, discussion, videos and presentation, code competition activity and assignments of computational problem solving. Practical Assignments 80% (Approximately weekly)
Evaluation Method	50% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc.

	50% assessment is based on the end semester written examination.

Computer Networks

Course Code	
Course Title	**Major: Computer Networks**
Credit	04(Theory)
Teaching per week	4 Hrs
Minimum weeks per semester	15 (Including Class work, examination, preparation, holidays etc.)
Last Review / Revision	2021-22
Purpose of Course	The purpose of the course is to make students understand the basics of networking and how one can design the same.
Course Objective	1. Learn about how computer networks are organized and it's protocol with the concept of layered approach 2. Learn concept of network management & its tool 3. Learn more about basic of Information & network security
Pre-requisite	Knowledge of basics of operating systems and any programming language.
Course Out come	After successful completion of the course students should be able to: • Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies; • Analyse, specify and design the topological and routing strategies for an IP based networking infrastructure • Analyse the requirements of company or organization security and its purpose and select to apply appropriate tools needed for the same; • Evaluate the authentication and encryption needs of an information system
Course Content	**Unit 1: Introduction to Computer Networks [6 Hrs]** 1.1 Basics of computer network, 1.1.1 Advantages and Disadvantages 1.1.2 Analogue and Digital Signals, Frequency, bandwidth, datagram, packets, frames, message, 1.1.3 Synchronous and Asynchronous communication, 1.1.4 Simplex, half-duplex and full-duplex transmission 1.2 Connecting devices: NIC, Bridges, router, switches, Repeater, Access Points 1.3 Network types: LAN, MAN, PAN, WAN 1.4 Network topology: Bus, Star, Ring, Mesh, Hybrid 1.5 Different Types of Transmission Media 1.5.1 Guided and unguided data transmission **Unit 2: Reference Model [8 Hrs]** 2.1 Physical Layer: End to end data transmission 2.2 Data Link Layer - MAC & LLC Sub layers 2.3 Error classification-Delay distortion Attenuation, noise 2.4 Types of errors – Single bit error and burst errors. 2.5 Error detection – Parity check (VRC) , LRC, CRC 2.6 CSMA/CD, CSMA/CA, IEEE Standards **Unit 3: Upper Level Layers [7 Hrs]** 3.1 Network Layer: Addressing - IP Address, subnet, gateway

	3.1.1 Physical and logical address, IP address Classes A to E 3.1.2 Use of different class at various places 3.1.3 IP Binding and IP cloning, Proxy, Static IP address and its benefits, routing table 3.2 Concept of Unicast, Broadcast, Multicast and anycast 3.3 Protocols and its use: NAT, ARP, RARP, ICMP, IPv4, IPv6, DHCP **Unit 4: Transport layer [7 Hrs]** 4.1 UDP and TCP-packet heads, services, communication, Flow and Error & congestion control 4.2 Session, Presentation, and Application Layers: SMTP, IMAP, H TTP, FTP, DNS, VOIP. **Unit 5: Network Management Concepts [8 Hrs]** 5.1 Creating and Managing ACL 5.2 VPN: Concept, Protocols, Applications 5.3 VLAN: Concept, Protocols, Applications 5.4 Concept of Broadcasting, Looping, Routing **Unit 6: Network Management Models [8 Hrs]** 6.1 OSI Network Management Model (FCAPS) 6.1.1 SNMP (Simple Network Management Protocol) 6.1.2 SNMP Working and Tools 6.2 overview of Performance Management 6.2.1 Performance metrics and monitoring 6.2.2 Performance tuning and optimization **Unit 7: Configuring servers with Server OS [8 Hrs]** 7.1 PDC, BDC and File server 7.2 Configuration of web server 7.3 Configuration of DHCP Server 7.4 Configuring Proxy Server 7.5 Configuring print server **Unit 8: Network Security [8 Hrs]** 8.1 Network security issues 8.1.1 Sniffing, IP spoofing, Common threats, E-Mail security, IPSec, SSL, TSL, PGP, Intruders, Virus, Worms 8.2 Firewalls-need and features of firewall, Types of firewalls 8.3 Overview: Symmetric Key Cryptography, Asymmetric Key Cryptography, Authentication
Text and Reference Literature	1. Data communications and network Behrouz A Forouzan, McGraw Hill 2. Data communications and networks, Achyut S Godbole, McGraw Hill 3. Fundamentals of computer networks, Sudakshina Kundu, PHI 4. Data communications and networking, Jain, BPB 5. Introduction to networking, McMahon, McGraw Hill 6. Data communications and networks, D B Rathod, K R Vishwa Jhananic, Himaliya publishing 7. Computer Security: Principles and Practice, 4th Edition, William Stallings, Lawrie Brown, Pearson

	8. Introduction to Computer Security, Matt Bishop, Addison Wesley 9. Information security, Dhiren Patel
Teaching Methodology	• Class lecture and discussions, independent study, assignments. Seminar, expert lectures, group projects and / or case studies (wherever applicable)
Evaluation Method	• 50% internal assessment is based on class attendance and participation, unit test, quiz, assignments, seminar, group project / case study evaluation, internal examination etc. 50% external assessment is based on semester end written examination.

Software Testing

Course Code	
Course Title	**SEC: Software Testing**
Credit	2
Teaching per Week	1 Hr (Theory) + 2 Hrs (Practical)
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Last Review / Revision	2021-22
Purpose of Course	This course imparts the knowledge of Software Testing. The concepts of software testing, role of testing, testing tools and reporting are covered in this course. The course is aimed to give inner depth of Software testing.
Course Objective	• To make students understand concepts of testing and testing practices. • To make students understand test automation process. • To make students understand Testing tools. • To make students writing and tracking test cases.
Prerequisite	Basic knowledge of Computer Programming and Application Development
Course Out come	After completion of this course, the student will be capable of understanding software development principles, processes to develop and design software systems.
Course Content	**Unit 1: Testing for Applications** **[8 Hrs]** 1.1 Testing life cycle, Test Exit criteria 1.2 Component level Testing 1.3 Navigation Testing 1.4 Configuration Testing 1.4.1 Server-side issues 1.4.2 Client-side issues 1.5 Security Testing 1.6 Performance Testing 1.6.1 Performance testing objectives 1.6.2 Load Testing 1.6.3 Stress Testing **Unit 2: Software Testing Oracles** **[6 Hrs]** 2.1 Role and objective of Testing 2.2 Testing Principles 2.3 Central Issue in Testing, Testing Activities 2.4 Testing Approaches- Whitebox, Blackbox and Grey box 2.5 Levels of testing: Unit, Integration, System and Acceptance **Unit 3: Test Automation Tools - I** **[8 Hrs]** 3.1 Manual Testing Vs. Automated Testing 3.2 Unit Testing with Microsoft.Net

	3.2.1 MStest, NUnit, xUnit 3.2.2. NUnit framework, Test Fixture, Test, Sepup & Tear Down, Asserts and Exception 3.3 Test case generation Tool-case study of TestCaseGenerator 3.4 Tool for Bug Tracking- case study of Bugzilla **Unit 4: Test Automation Tools - II** **[8 Hrs]** 4.1 Tool for Test Management-case study of Tlink 4.2 Renorex - Desktop and Web Automation 4.3 Jmeter / LoadRunner - Load & Stress Testing 4.4 Selenium - Web Application Test automation tool 4.5 Appium - Mobile Automation 4.6 Test Sigma – Scriptless Test Automation 4.7 Jira – Project Management Tool
	1. Ron Patton ―Software Testing, Techmedia Publication, 2000 2. Dr. K.V.K.K prasad, ―Software Testing Tools , Dreamtech, 2006 3. Srinivas D and Gopalswamy R, ―Software Testing: Principles and Practices. Pearson Education, 2013 4. K. Mustafa and R.A Khan, ―Software Testing -concepts and practices‖, Narosa, 2012 5. Bill Hamilton, ―NUnit: pocket Referance, SDP-OReilly, 2004 6. Andrew Hunt and David Thomus, ―Pragmatic Unit Testing in Java with JUnit, PD, 2006 7. Testing with JUnit by Frank appeal PACKT Publishing 8. Software testing Principal and practices by Naresh Chauhan – OXFORD 9. Software testing (A Practical approach) by Rajiv Chopra – S K Kataria & Sons (KATSON Books) 10 Software testing and quality assurance Theory and practice by Kashirasagar Naik and Priyadarshini Tripathy – Wiley india Pvt Ltd.
Teaching Methodology	Discussion, Independent Study, Seminars and Assignment
Evaluation Method	50% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 50% assessment is based on end semester written examination