BCA(Cloud Computing)

Semester-III

Semester – III	**Course Category**			
3BCC3-DM-002-T	**L**	**T**	**P**	**C**
Linux Administration	2	0	0	2

Course Objectives

- To provide an understanding of the historical development of Linux and its key contributors..
- To Navigate and utilize the vi editor for efficient text editing.
- To Describe the concepts of standard I/O, redirection, and pipes in Linux..
- To Develop and execute basic shell scripts with branching and loop-control structures.

Course Outcomes

Course Outcomes(COs)	**Level***
CO1: Describe the architecture, components, and installation process of the Linux operating system.	L1
CO2: Demonstrate the use of command-line tools and vi editor to manage files and directories in the Linux filesystem.	L2
CO3: Execute I/O redirection, pipes, and text processing tools to manipulate data effectively.	L3
CO4: Implement user account management, group configuration, and software package administration.	L2 & L3
CO5: Analyze and construct shell scripts to automate tasks using control structures and expressions.	L4 & L5

***Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*Outcome.

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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PSO 1	PSO 2
CO1	3	2	1	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	2
CO3	3	2	2	1	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	3
CO5	3	2	3	2	2	2	3	2	2	3	2	3

Course Contents:

Unit-I

A Brief History of Linux, Basic features of Linux OS, components of Linux System, Benefits of Linux, Linux Distributions, Free and Open-Source Software, Linux Kernel, booting process of Linux OS, Linux filesystem types ext2, ext3, ext4, Linux File Managers, Basic Linux directory structure, Managing Linux Files and Folders: Introduction, Linux Files and Folders, Creating Files and Folders, Managing Files and Folders, Searching for Files.

Unit-II

The vi Editor: vi Editor, editing with vi, Moving the Cursor, Editing, Copying and Moving Text, Pattern Searching, Repeating the Last Editor Command, Undoing Commands, Joining Lines, Writing Selected Lines onto a Separate File, Using the Shell from vi, Configuring the vi Environment.

File Attributes and Permissions: Ownership of Files, File Attributes, File Command, Changing File Permission, Changing the Owner of a File, Changing the group of a File, Times Associated with a File, umask Command

Unit-III

Standard I/O: stdout, stdin, stderr, Redirection Pipes and Filters: I/O, Redirection, Pipe and Pipeline, Regular Expressions: grep Family of Commands and sed: Regular Expressions, grep Family, egrep Command, fgrep Command, working with text: haid, tail, more, less, pipes, sort, wc, cut.

Unit-IV

User Environment: Use and configure user accounts and user groups-Identifying the Current User, Basics of Users and Groups, Adding and Removing Users, Adding and Removing Groups, The Root Accounts, Elevating to root Account, Startup files.

Managing Packages & Users: Installing and removing Software in Linux, Getting and Unpacking the Package, Configuring the Package, Compiling the Package, Installing the Package, Managing Users and Groups

Unit-V

Shell Programming: Shell Variables, export Command, profile File – A Script Run during Starting, The First Shell Script, read Command, Positional Parameters, The $? Variable – Knowing the Exit Status, more about the set Command, exit Command, Branching Control Structures, Loop-Control Structures, continue and break Statements, expr Command, Real Arithmetic in Shell Programs.

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, Assignment I-V, Q, MST-I & II #)**
Weightage (%)	60	40

*A-Attendance; Assignment I-V (Class Assignment/Home Assignments/Case Discussions/Term Papers/Mini Project); Q-Quiz (3 Quizzes), MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE office at MU).

Textbooks:

1. Linux Administration:A Beginner's Guide by Steve Shah,Wale Soyinka, ISBN 0072262591(0-07-226259-1), McGraw-Hill Education

Reference Books:

1. Unix and Linux System Administration Handbook, by Evi Nemeth,Garth Snyder, Trent R. Hein,Ben Whaley.
2. Red Hat Enterprise Linux 4: System Administration Guide Copyright 2005 Red Hat, Inc

MANDSAUR UNIVERSITY
DREAM. LEARN. LEAD.

BCA (Cloud Computing)

Semester – III	**Course Category**
3BCC3-DC-001-T	**L T P C**
Web Technology	4 0 0 4

Course Objectives

- Gain knowledge of web development fundamentals and design static web pages using HTML elements, tables, forms, lists, and multimedia.
- Learn to use Cascading Style Sheets (CSS) to control the visual presentation and layout of web pages.
- Understand client-side scripting using JavaScript to handle events, validate forms, and control program flow.
- Get familiar with server-side scripting using PHP with data types, control structures, loops, and array manipulation.
- Explore dynamic web application development by integrating PHP with MySQL database and managing user sessions and cookies.

Course Outcomes

Course Outcomes(COs)	**Level***
CO1: Describe the structure of HTML and design static web pages using text, images, forms, tables, and lists.	L1
CO2: Demonstrate the use of CSS to style and control the layout and presentation of web pages.	L2 & L3
CO3: Implement client-side scripting using JavaScript to handle events, validate forms, and control program flow.	L4
CO4: Apply PHP programming concepts including variables, control structures, loops, and arrays to develop server-side logic.	L4 & L4
CO5: Analyze and develop dynamic web applications by integrating PHP with MySQL and managing sessions and cookies.	L5

**Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*

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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PSO 1	PSO 2
CO1	3	2	1	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	2
CO3	3	2	2	1	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	3
CO5	3	2	3	2	2	2	3	2	2	3	2	3

Course Contents:

Unit-I

Static Web page Development: Introduction to Hyper TextMarkup Language (HTML), Elements of HTML, Basic structure of an HTML, document Head & Body Sections, inserting texts, Text alignment, using images in pages, Hyperlinks text, Forms in HTML, Backgrounds and Color controls, creating and using Tables in HTML, Creating Lists: Ordered List Tags, Unordered List Tag. Changing the Font Color; Marquee Tag.

Unit-II Creating Static Web Pages

Dynamic Web page Development: Cascading Style Sheet:CSS, Defining Style with HTML Tags, Features of Style Sheet, StyleProperties, CSS Styling (Background, Text Format, Controlling Fonts), Working with Lists and Tables, Style placement: Inline style, Span & div tags, header styles.CSS Id and Class Working with block elements and objects,CSS Color.

Unit-III Creating Dynamic Web page

JavaScript Overview: JavaScript and the WWW, Script, element,Functions: Functions introduction, calling functions, JavaScript Comments, Variables: Variables overview, declaring variables, Types of variables, Casting variables, Alert box, Prompt & confirm. Expressions: Arithmetic operators,Assignment operators, Logical operators, Expressions and precedence, Statements: If statement; for statement, while statement, Break/Continue.

Unit-IV JavaScript: Functions, Variables & Operators

Introduction to PHP, PHP data types, Variable, Constants, Variable scope, Operators, Variable manipulation, Dynamic variables, Static vs. Dynamic Optimization, If, else if, switch statement, loops: while, do while, for, for each, breaking out of loops: Break, Continue, and exit. **Array:** Indexed arrays, Associative arrays, Multidimensional arrays, Getting the size of an array, looping through an array, looping through an associative array, Sorting arrays, Sorting an associative array.

Unit-V PHP Essentials: Basics of Data, Variables, and Control

GET and POST data, Combine HTML and PHP code, create user Forms using database, File Inclusion: Include (), Require (), Importing user input, accessing user input,Connection with MySQL Database, performing basic database operations (DML) (Insert, Delete, Update, Select), Setting query parameter, Executing query. **Session and Cookies:** Introduction to Session Control, Creating Session, Set Session, Destroying Session**.**

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, Assignment I-V, Q, MST-I & II #)**
Weightage (%)	60	40

*A-Attendance; Assignment I-V (Class Assignment/Home Assignments/Case Discussions/Term Papers/Mini Project); Q-Quiz (5 Quizzes), MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE office at MU).

Reference Books:

1. HTML and Web designing - Kris Jamsa and Konrad King
2. Web Technology - N.P. Goplan, J. Akilandeswari
3. Internet Technology and Web Design - ISRD Group
4. Learning PHP, MySQL, By O' riley Press
5. Eric Filson , Erick Rosebrock, Setting up LAMP: Getting Linux, Apache, MySQL, and PHP Working Together, SyBex (2004).

List of e-Learning Resources:

1. https://www.coursera.org/courses?query=web%20technologies
2. https://onlinecourses.swayam2.ac.in/nou24_cs09/preview
3. https://www.udemy.com/course/web-technology-for-entrepreneurs/?couponCode=ST20MT50724
4. https://iisdt.in/product/certificate-in-web-technology

BCA (Cloud Computing)

Semester-III

Semester – III	**Course Category**			
3BCC3-DM-002-P	**L**	**T**	**P**	**C**
Linux Administration Lab	0	0	4	2

Course Objectives

- Understand the fundamental concepts, components, and architecture of the Linux operating system including installation and booting process.
- Create, manage, and manipulate files and directories using the Linux filesystem and vi editor.
- Perform system administration tasks such as user management, privilege handling, and software package installation.
- Configure and customize the Linux shell environment using startup files, profiles, and shell variables.
- Write, execute, and debug shell scripts using control structures, loops, and arithmetic operations.

Course Outcomes

Course Outcomes(COs)	**Level***
CO1: Describe the architecture, components, and installation process of the Linux operating system.	L1
CO2: Demonstrate the use of command-line tools and vi editor to manage files and directories in the Linux filesystem.	L2
CO3: Execute I/O redirection, pipes, and text processing tools to manipulate data effectively.	L3
CO4: Implement user account management, group configuration, and software package administration.	L2 & L3
CO5: Analyze and construct shell scripts to automate tasks using control structures and expressions.	L4 & L5

**Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*

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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PSO 1	PSO 2
CO1	3	2	2	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	1
CO3	3	1	2	2	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	2
CO5	3	2	3	2	2	2	3	2	2	3	2	3

List of Experiments

1. Exploring Linux OS: Understanding basic features, components, and distributions.
2. Installation and Booting Process: Installing Linux OS and understanding the booting process.
3. Filesystem Management: Creating, managing, and navigating Linux files and directories.
4. File Editing with vi Editor: Introduction to vi editor, editing, copying, and moving text.
5. Configuring vi Environment: Customizing vi editor settings and configurations.
6. File Attributes and Permissions: Understanding and modifying file permissions and attributes.
7. Redirection and Pipes: Using redirection and pipes for standard I/O and text processing.
8. Regular Expressions: Exploring grep family of commands and using regular expressions.
9. Text Manipulation Tools: Utilizing tools like head, tail, more, less, sort, wc, cut for text manipulation.
10. User Management: Adding, removing, and configuring user accounts and groups.
11. Root Account Management: Understanding root accounts and elevating privileges.
12. Package Management: Installing, removing, and managing software packages in Linux.
13. Compiling and Installing Software: Getting, unpacking, configuring, compiling, and installing packages.
14. Managing User Environment: Configuring user startup files and profiles.
15. Exploring Shell Variables: Understanding shell variables and the export command.

16. Shell Scripting Basics: Writing and executing basic shell scripts.
17. Working with Shell Scripts: Understanding positional parameters, exit status, and control structures.
18. Advanced Shell Scripting: Exploring branching and loop control structures.
19. Using expr Command: Performing real arithmetic in shell programs.
20. Script Testing and Debugging: Testing, debugging, and optimizing shell scripts.

Total 60 Hours

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, LR, MST-I & II #)**
Weightage (%)	50	50

*A-Attendance; Lab Record Submission, MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE office at MU).

BCA (Cloud Computing)

Semester – III	**Course Category**			
3BCC3-DE-003-P-03	**L**	**T**	**P**	**C**
Server-side Scripting Lab	0	0	8	4

Course Objective:

- To understand the basic concepts and architecture of Node.js and server-side scripting.
- To learn npm (Node Package Manager) and how to manage dependencies in Node.js applications.
- To develop web applications using Express.js and handle routing and middleware.
- To learn how to create dynamic web pages using templating engines like Pug and EJS.
- To understand MongoDB and use MongoDB Query Language (MQL) for database operations.

Course Outcomes:

Course Outcomes (COs)	**Level ***
CO1: Understand the concepts of Node.js, event-driven architecture, and non-blocking I/O model.	L1 & L2
CO2: Apply npm and CommonJS modules to build and manage Node.js applications.	L3
CO3: Develop server-side applications using Express.js with routing, middleware, and templating engines.	L4
CO4: Analyze and optimize application performance using caching and error handling techniques.	L5
CO5: Create and manage MongoDB databases and perform CRUD operations using MQL.	L6

**Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*

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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PSO 1	PSO 2
CO1	3	2	1	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	2
CO3	3	2	2	1	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	3
CO5	3	2	3	2	2	2	3	2	2	3	2	3

List of Experiments:

1. Install & Verify: Install Node.js and check versions using `node -v` and `npm -v`.
2. Project Init: Create a folder and run `npm init` to generate a `package.json`.
3. Hello World: Write a script to log "Hello, Node.js" to the terminal.
4. REPL Practice: Use the Node.js REPL to perform basic math and string operations.
5. Event Emitter: Create a custom event (e.g., 'greet') and emit it using the `events` module.
6. HTTP Server: Build a server that sends "Welcome to my Server" to a browser.
7. Variable Scope: Demonstrate the difference between `var`, `let`, and `const`.
8. Data Types: Write a script that identifies and logs different types (String, Number, Boolean, Object).
9. Basic Logic: Create a "Grade Calculator" using `if-else` and `switch` statements.
10. Loops: Use `for` and `while` loops to print prime numbers.
11. Functions: Write a function that calculates the area of a circle.
12. Closures: Create a counter function that "remembers" its value using closures.
13. Local Modules: Create a `math.js` file, export add/sub functions, and import them in `app.js`.
14. NPM Install: Install the `colors` package and use it to print colored text in the terminal.
15. Async File Read: Use `fs.readFile` to read a text file without blocking the script.
16. Sync File Write: Use `fs.writeFileSync` to create a log file.
17. Folder Ops: Write a script to create, rename, and delete a directory.
18. Buffers: Create a Buffer and convert it to a string/JSON.
19. Streams (Read): Read a large text file using `fs.createReadStream`.
20. Streams (Write): Pipe data from a readable stream to a writable stream (File Copy).
21. Callback Hell: Demonstrate nested callbacks for three sequential file operations.
22. Promises: Convert a callback-based file read into a Promise-based one.
23. Async/Await: Use `async/await` to fetch and log data from a mock function.
24. Global Objects: Experiment with `__dirname` and `__filename` to resolve paths.
25. Express Setup: Install Express and create a server listening on port 3000.
26. Basic Routing: Create routes for `/home`, `/about`, and `/contact`.
27. URL Params: Create a route like `/user/:id` and display the ID in the response.
28. Query Strings: Access data from a URL like `/search?term=node`.

29. Static Files: Use `express.static` to serve an `images` folder.
30. Middleware 1: Write a "Logger" middleware that prints the current time for every request.
31. Middleware 2: Create a middleware that restricts access based on a "secret key" in the header.
32. EJS Templates: Setup EJS and render a dynamic "User Profile" page.
33. Pug Templates: Setup Pug and pass an array of items to a list view.
34. Form Handling: Create an HTML form and use `body-parser` to read submitted data.
35. Redirects: Use `res.redirect()` to send users from `/old-page` to `/new-page`.
36. JSON Response: Create an API route that returns a list of products in JSON format.
37. Error Middleware: Create a 404 "Not Found" handler.
38. Global Error Handler: Write a middleware with four arguments to catch app errors.
39. Passport.js Setup: Implement a basic Local Strategy for login.
40. Session Management: Use `express-session` to keep a user logged in.
41. JWT Basics: Generate a JSON Web Token for an authenticated user.
42. API Fetch: Use `axios` inside an Express route to get weather data from a third-party API.
43. Environment Vars: Use `dotenv` to store and hide your Port and API keys.
44. CORS: Enable Cross-Origin Resource Sharing for your API.
45. Validation: Use `express-validator` to check if an email input is valid.
46. Socket.IO Setup: Create a basic "Ping-Pong" real-time connection.
47. Real-time Chat: Build a simple one-room chat where messages appear instantly.
48. Helmet.js: Use the Helmet module to secure your HTTP headers.
49. Mongo Install: Install MongoDB and verify it is running as a service.
50. Mongo Shell: Use `mongosh` to create a database named `college`.
51. Insert One: Create a document in a `students` collection.
52. Insert Many: Add five documents at once to a `products` collection.
53. Find Query: Use filters (e.g., `price: { $gt: 500 }`) to search data.
54. Update Docs: Use `$set` to update a student's email address.
55. Delete Docs: Remove a specific record using its `_id`.
56. Mongoose Setup: Connect your Express app to MongoDB using the Mongoose library.
57. Schema Design: Define a Mongoose Schema for a "Blog Post" (Title, Author, Date).
58. Full CRUD App: Create a "To-Do List" app where you can Add, View, and Delete tasks from the DB.
59. Indexing: Create an index on the "email" field to speed up searches.
60. Aggregation: Use the `$group` pipeline to find the average age of users in your database.

Total: 120 Hours

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, Assignment I-V, Q, MST-I & II #)**
Weightage (%)	50	50

*A-Attendance; Assignment I-V (Class Assignment/Home Assignments/Case Discussions/Term Papers/Mini Project); Q-Quiz (5 Quizzes), MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE

office at MU).

List of Books:

Textbook:

1. Node.js, Express and MongoDB Development, Ethan Brown, O'Reilly Media, 2nd Edition, 2019.

Reference Books:

1. Learning Node: Moving to the Server-Side with JavaScript, Shelley Powers, O'Reilly Media, 2nd Edition, 2016.
2. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage, Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, O'Reilly Media, 3rd Edition, 2019.

Important Websites:

1. **Node.js Documentation – Official Website** https://nodejs.org/en/docs/ (2026)
2. **Express.js Guide – Official Website** https://expressjs.com/en/guide/routing.html (2026)
3. **MongoDB Manual – MongoDB University** https://www.mongodb.com/docs/manual/ (2026)
4. **Node.js Tutorials – TutorialsPoint** https://www.tutorialspoint.com/nodejs/index.htm (2026)
5. **Mongoose ODM Reference – Mongoose Documentation** https://mongoosejs.com/docs/guide.html (2026)
6. **Passport.js Authentication Guide – PassportJS** https://www.passportjs.org/docs/ (2026)
7. **JavaScript Server-side Basics – MDN Web Docs** https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs (2026)

MANDSAUR UNIVERSITY
DREAM. LEARN. LEAD.

BCA (Cloud Computing)

Semester – III	**Course Category**			
3BCC3-DE-003-P-02	**L**	**T**	**P**	**C**
Network Security Lab	0	0	8	4

Course Objective:

- Define basic concepts of network security, including Confidentiality, Integrity, and Availability.
- Explain how different security protocols work and how network security has improved over time.
- Apply practical skills using tools like Nmap, Wireshark, and Metasploit for network analysis.
- Analyze network traffic and logs to identify security issues and unauthorized access.
- Evaluate network security and suggest methods to improve protection against cyber-attacks.

Course Outcomes:

Course Outcomes (COs)	**Level ***
CO1: Identify common network vulnerabilities and risks related to networking devices.	L1
CO2: Demonstrate secure communication using SSH, VPNs, and encryption methods.	L1 & L2
CO3: Perform vulnerability assessment and packet analysis to find data leaks and plain-text information.	L3
CO4: Analyze suspicious network traffic and perform basic digital forensics.	L5
CO5: Test firewall and IDS/IPS security by simulating network attacks like DoS and MITM.	L5

**Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*

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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PSO 1	PSO 2
CO1	3	2	1	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	2
CO3	3	2	2	1	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	3
CO5	3	2	3	2	2	2	3	2	2	3	2	3

List of Experiments:

1. IP Configuration: Use ipconfig / ifconfig to identify network interfaces and Gateway details.
2. Ping Testing: Use ping to check connectivity and analyze TTL (Time to Live) values.
3. Traceroute: Use tracert (Windows) or traceroute (Linux) to map the path to a website.
4. NSLookup: Find the IP address and DNS records of a specific domain.
5. Netstat: List all active network connections and listening ports on your machine.
6. ARP Table: View and clear the Address Resolution Protocol (ARP) cache.
7. Hostname Identification: Use hostname and whois to gather information about a target.
8. Physical Address: Identify the MAC address of your NIC and understand its structure.
9. File Sharing Security: Set up a shared folder with "Read Only" vs. "Full Control" permissions.
10. Browser Security: Configure privacy settings and clear SSL state in a web browser.
11. Network Services: Use services.msc to identify and stop unnecessary network services.
12. Basic Firewall: Enable/Disable Windows Firewall and create a rule to block a specific app.
13. Telnet Basics: Use Telnet to connect to a remote port and understand why it’s insecure.
14. Pathping: Combine Ping and Traceroute to analyze network latency at each hop.
15. Banner Grabbing: Use Netcat (nc) to identify the version of a running web server.
16. Port Scanning (Basic): Use a simple tool to check if Port 80 and Port 443 are open.
17. Resource Monitoring: Use Task Manager/Resource Monitor to see which apps use the most bandwidth.
18. WiFi Security: Identify different encryption types (WEP, WPA2, WPA3) in your range.
19. Host File Editing: Modify the hosts file to redirect a domain to a local IP.
20. MAC Filtering: Configure a router to allow/deny access based on MAC addresses.
21. Wireshark Basics: Capture "Live" traffic and filter by HTTP protocol.
22. Nmap Scanning: Perform a TCP Connect scan on a target IP.
23. OS Fingerprinting: Use Nmap (-O) to detect the operating system of a remote device.
24. Stealth Scanning: Perform an Nmap SYN scan (-sS) to avoid detection.
25. Packet Analysis: Capture a login attempt on an unencrypted (HTTP) site to see the password.

26. SSH Setup: Configure a secure shell (SSH) connection between two Linux machines.
27. Public vs. Private Keys: Generate an RSA key pair for secure communication.
28. Firewall Rules: Create an Inbound Rule to block all traffic from a specific IP range.
29. Snort Installation: Install Snort IDS (Intrusion Detection System) on a virtual machine.
30. Vulnerability Scanning: Use Nessus or OpenVAS to find holes in a Windows 7/10 VM.
31. Email Header Analysis: Inspect an email header to find the sender's IP and mail server.
32. Hashing: Use tools to calculate MD5 and SHA-256 hashes of a file to check integrity.
33. VPN Configuration: Set up a basic Point-to-Point VPN connection.
34. Service Version Detection: Use Nmap (-sV) to find vulnerable versions of running services.
35. Analyzing ICMP: Capture and analyze different ICMP message types (Type 8, Type 0).
36. Wi-Fi Password Auditing: Understand how WPA2 handshakes are captured (Theory/Tools).
37. Proxy Setup: Configure a browser proxy to hide your IP address.
38. Digital Signatures: Sign a document digitally and verify it using a tool.
39. Intrusion Detection: Configure Snort to detect a "Ping of Death" attack.
40. Process Analysis: Use TCPView to see which remote IPs your local processes are talking to.
41. ARP Spoofing: Use Ettercap or Arpspoof to demonstrate a Man-in-the-Middle (MITM) attack.
42. DNS Poisoning: Redirect a user from a legitimate site to a fake site using Kali Linux.
43. DoS Simulation: Perform a SYN Flood attack in a lab environment and monitor CPU usage.
44. SQL Injection: Use a vulnerable web app (like DVWA) to extract data from a database.
45. Cross-Site Scripting (XSS): Perform a basic reflected XSS attack on a test site.
46. Metasploit Basics: Use the Metasploit framework to test a known vulnerability in a VM.
47. Password Cracking: Use John the Ripper to crack a simple Linux password hash.
48. Brute Force Attack: Use Hydra to test the strength of a local FTP/SSH service.
49. HoneyPot Setup: Set up a "Pentbox" Honeypot to trap and log attacker IPs.
50. Network Forensics: Analyze a .pcap file to reconstruct a file downloaded by a user.
51. Encrypted Traffic Analysis: Try to identify application types in HTTPS traffic (without decrypting).
52. Rootkit Detection: Use Rootkit Hunter or Chkrootkit to scan a compromised system.
53. Steganography: Hide a text message inside an image file and extract it.
54. Social Engineering Toolkit (SET): Create a fake login page for educational purposes.
55. Exploit Research: Find an exploit on Exploit-DB for a specific service version.
56. Linux Hardening: Disable unnecessary ports and set up Fail2Ban to block repeated login failures.
57. Wireless De-authentication: Demonstrate how a device is kicked off a network (Lab only).
58. Traffic Shaping: Use a router or software to limit bandwidth for specific security protocols.
59. Packet Crafting: Use Scapy to create a custom TCP packet with specific flags.

60. Security Audit: Conduct a full "Black Box" audit of a test Virtual Machine and write a report.

Total: 120 Hours

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, Assignment I-V, Q, MST-I & II #)**
Weightage (%)	50	50

*A-Attendance; Assignment I-V (Class Assignment/Home Assignments/Case Discussions/Term Papers/Mini Project); Q-Quiz (5 Quizzes), MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE office at MU).

List of Books:

Textbook:

1. **Network Security Essentials: Applications and Standards**, William Stallings, Pearson, 6th Edition, 2017.
2. **Cryptography and Network Security: Principles and Practice**, William Stallings, Pearson, 8th Edition, 2020.

Reference Books:

1. **Network Security: Private Communication in a Public World**, Charlie Kaufman, Radia Perlman, and Mike Speciner, Prentice Hall, 2nd Edition, 2002.
2. **Applied Cryptography: Protocols, Algorithms, and Source Code in C**, Bruce Schneier, John Wiley & Sons, 20th Anniversary Edition, 2015.
3. **Firewalls and Internet Security: Repelling the Wily Hacker**, William R. Cheswick and Steven M. Bellovin, Addison-Wesley, 2nd Edition, 2003.
4. **Network Forensics: Tracking Hackers through Cyberspace**, Sherri Davidoff and Jonathan Ham, Prentice Hall, 1st Edition, 2012.

Important Websites:

1. **Network Security Overview** – *TutorialsPoint* https://www.tutorialspoint.com/network_security/index.htm (2026)
2. **Cryptography & Network Security Concepts** – *GeeksforGeeks* https://www.geeksforgeeks.org/cryptography-and-network-security/ (2026)
3. **Cryptography and Network Security** – *Prof. Sourav Mukhopadhyay, NPTEL (IIT Kharagpur)* https://nptel.ac.in/courses/106105162 (2026)
4. **Cisco Networking Academy: Cybersecurity Essentials** – *Cisco* https://www.netacad.com/courses/cybersecurity/cybersecurity-essentials (2026)
5. **Wireshark User's Guide (Network Analysis Tool)** – *Wireshark.org* https://www.wireshark.org/docs/wsug_html_chunked/ (2026)

BCA(Cloud Computing)

Semester-III

Semester – III	**Course Category**			
3BCC3-DE-003-P-01	**L**	**T**	**P**	**C**
Server Operating System Lab	0	0	4	2

Course Objectives

- To understand the fundamentals of Windows Server installation, configuration, and system management.
- To develop skills in managing storage systems, file systems, and access permissions.
- To gain practical knowledge of basic networking services such as DHCP and DNS.
- To learn the installation and administration of Active Directory, including users, groups, and domain management.
- To implement basic security measures using Group Policy and Windows Firewall.

Course Outcomes

Course Outcomes(COs)	**Level***
CO1: Understand and explain the basic concepts of Windows Server installation, system configuration, and networking fundamentals.	L1
CO2: Apply knowledge to configure disk management, file systems, and access permissions in a Windows Server environment.	L1 & L2
CO3: Implement and manage basic network services such as DHCP and DNS.	L3
CO4: Apply and manage Active Directory components including users, groups, and organizational units.	L4
CO5: Analyze and configure security settings using Group Policy and Windows Firewall to secure the server environment.	L5

**Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*

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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PSO 1	PSO 2
CO1	3	2	1	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	2
CO3	3	2	2	1	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	3
CO5	3	2	3	2	2	2	3	2	2	3	2	3

List of Experiments

1. Install Windows Server (GUI mode) and explore Server Manager.
2. Configure system settings (hostname, date/time, updates).
3. Configure static IP address and verify connectivity.
4. Use basic network tools (ping, ipconfig, tracert, nslookup).
5. Initialize disks using MBR and GPT.
6. Create and format partitions with NTFS.
7. Create and attach VHD/VHDX virtual disks.
8. Configure NTFS permissions on files and folders.
9. Configure shared folders and share permissions.
10. Compare and test NTFS vs Share permissions.
11. Implement disk quotas for users.
12. Enable file compression and encryption (EFS).
13. Create and manage basic Storage Spaces (simple setup).
14. Perform IPv4 addressing and subnetting exercises.
15. Install and configure DHCP Server.
16. Create and manage DHCP scopes.
17. Configure DHCP reservations.
18. Install and configure DNS Server.
19. Create forward lookup zones and basic DNS records.
20. Test name resolution using nslookup and ping.
21. Install Active Directory Domain Services (AD DS).
22. Promote server to Domain Controller.
23. Create and manage users and groups.
24. Design and manage Organizational Units (OUs).
25. Join a client machine to the domain.
26. Perform user account management (reset password, lock/unlock).
27. Create and apply Group Policy Objects (GPOs).
28. Configure password and account lockout policies.
29. Apply simple desktop restrictions using GPO.
30. Configure basic Windows Firewall rules.

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, LR, MST-I & II #)**
Weightage (%)	50	50

*A-Attendance; Lab Record Submission, MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE office at MU).

MANDSAUR UNIVERSITY
DREAM. LEARN. LEAD.

BCA(Cloud Computing)

Semester – III	**Course Category**			
3BCC3-DC-001-P	**L**	**T**	**P**	**C**
Web Technology Lab	0	0	4	2

Course Objectives

- Gain knowledge of web development fundamentals and design static web pages using HTML elements, tables, forms, lists, and multimedia.
- Learn to use Cascading Style Sheets (CSS) to control the visual presentation and layout of web pages.
- Understand client-side scripting using JavaScript to handle events, validate forms, and control program flow.
- Get familiar with server-side scripting using PHP with data types, control structures, loops, and array manipulation.
- Explore dynamic web application development by integrating PHP with MySQL database and managing user sessions and cookies.

Course Outcomes

Course Outcomes(COs)	**Level***
CO1: Describe the structure of HTML and design static web pages using text, images, forms, tables, and lists.	L1
CO2: Demonstrate the use of CSS to style and control the layout and presentation of web pages.	L2 & L3
CO3: Implement client-side scripting using JavaScript to handle events, validate forms, and control program flow.	L4
CO4: Apply PHP programming concepts including variables, control structures, loops, and arrays to develop server-side logic.	L4 & L4
CO5: Analyze and develop dynamic web applications by integrating PHP with MySQL and managing sessions and cookies.	L5

**Level of Learning: Level 1 (L1) - Remember ; Level 2 (L2) – Understand; Level 3 (L3) –Applye; Level 4 (L4) –Analyze; Level 5 (L5) -Evaluate;.Level 6 (L6) -Create. Mention the highest level that will be attained in the Course Outcome.*

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	3	2	1	1	-	-	2	1	-	2	2	1
CO2	3	2	2	2	1	2	2	1	1	2	3	2
CO3	3	2	2	1	1	-	2	-	-	2	3	2
CO4	3	2	3	3	2	2	2	1	1	2	2	3
CO5	3	2	3	2	2	2	3	2	2	3	2	3

List of Experiments

1. Create a basic HTML page demonstrating the head and body structure with text formatting tags.
2. Design a webpage using text alignment, font color, and marquee tag.
3. Create a webpage with ordered and unordered lists.
4. Insert images and create hyperlinks (internal and external) in a webpage.
5. Design an HTML form with text fields, radio buttons, checkboxes, dropdowns, and submit buttons.
6. Create a webpage with background colors, background images, and color controls.
7. Design a webpage using HTML tables with rowspan and colspan.
8. Apply inline, internal, and external CSS to style a webpage.
9. Design a webpage using CSS background and text formatting properties.
10. Style fonts and control font size, weight, family, and style using CSS.
11. Create and style ordered and unordered lists using CSS.
12. Design a styled table using CSS border, padding, and spacing properties.
13. Demonstrate the use of CSS id and class selectors with div and span tags.
14. Create a webpage layout using block elements with CSS styling.
15. Write a JavaScript program to demonstrate the use of alert, prompt, and confirm boxes.
16. Create a webpage demonstrating variable declaration, data types, and type casting in JavaScript.
17. Write a JavaScript program using arithmetic, assignment, and logical operators.
18. Implement if-else conditional statements using JavaScript with form input.
19. Write JavaScript programs using for, while loop and break/continue statements.
20. Create a webpage with JavaScript functions that are called on button click events.
21. Write a JavaScript program to validate an HTML form (name, email, password fields).
22. Write a PHP program to demonstrate variables, constants, data types, and variable scope.
23. Implement if, else-if, and switch statements in PHP with user input.
24. Write PHP programs using while, do-while, for, and foreach loops with break and continue.

25. Create and manipulate indexed, associative, and multidimensional arrays in PHP.
26. Write a PHP program to sort indexed and associative arrays using built-in functions.
27. Create an HTML form and process GET and POST data using PHP.
28. Connect PHP with MySQL database and perform Insert, Delete, Update, and Select operations.
29. Design a user registration and login form using PHP combined with MySQL database.
30. Demonstrate session creation, setting session variables, and destroying sessions using PHP with cookies.

Total 120 Hours

Examination Scheme: Total – 100 marks

Components Continuous Internal Assessment*	**External Assessment (EST #)**	**(A, LR, MST-I & II #)**
Weightage (%)	50	50

*A-Attendance; Lab Record Submission, MST-I, MST-II, EST. (# MST-I & IIconducted at Department Level & EST (External Assessment) will be conducted by the CoE office at MU).