

**COMPUTER SCIENCE DEPARTMENT**



**University  
of  
Calcutta**

**B.Sc (Honours and  
Honours with Research)  
4 - years degree program in  
Computer Science under  
credit framework.**

**(2023)**

**Semester – I & II**

<b>Semester - I</b>				
<b>Paper</b>	<b>Paper type</b>	<b>Paper name</b>	<b>Credit</b>	<b>Contact hours</b>
DSC/CC-1	Theory	Computer fundamentals and Digital Logic	3	45
	Practical	Computer fundamentals and Digital Logic lab	1	30
SEC – 1	Theory	Data visualization using spreadsheet	3	45
	Practical	Data visualization using spreadsheet Lab	1	30

<b>Semester - II</b>				
<b>Paper</b>	<b>Paper type</b>	<b>Paper name</b>	<b>Credit</b>	<b>Contact hours</b>
DSC/CC-2	Theory	Problem Solving using C	3	45
	Practical	Problem Solving using C Lab	1	30
SEC – 2	Theory	Web Development	3	45
	Practical	Web Development Lab	1	30

Semester - I				
Paper	Paper type	Paper name	Credit	Contact hours
DSC/CC-1	Theory	Computer fundamentals and Digital Logic	3	45
	Practical	Computer fundamentals and Digital Logic lab	1	30
SEC – 1	Theory	Data visualization using spreadsheet	3	45
	Practical	Data visualization using spreadsheet Lab	1	30

**CMSA- Theory: Computer Fundamentals and Digital Logic**  
**Core Course, Theory, Semester – 1, Credits - 03, Contact hours - 45.**

**Course description:**

The course introduces the fundamental principles and concepts of digital logic, which form the foundation of digital systems and computer architecture. Students will learn about Boolean algebra, logic gates, combinational and sequential circuits, and the design and analysis of digital systems.

**Course Objectives:**

By the end of the course, students should be able to:

1. Understanding of Computer fundamentals, generations, classification of computers and brief understanding of languages used.
2. Understand the principles and terminology of digital logic.
3. Analyze and simplify Boolean expressions using Boolean algebra.
4. Design and implement combinational logic circuits using logic gates.
5. Design and analyze sequential logic circuits, including flip-flops and registers.
6. Apply digital logic concepts to solve practical problems.
7. Utilizing discrete logic gates and integrated circuits on breadboards for the design of digital circuits to enhance hands-on experience and practical understanding.

<b>Computer Fundamentals</b>	
Central Processing Unit (CPU), Primary memory and Secondary Storage devices, I/O devices, generation and classification of Computers: Super, Mainframe, Mini and Personal Computer, System and Application Software, basic concepts on Machine, Assembly and High level Language.	2 hours
<b>Number Systems</b>	
Weighted and Non - Weighted Codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases, signed arithmetic, 1's, 2's complement representation, Parity bits. <b>Single bit error detection and correcting codes:</b> Hamming Code. <b>Fixed and floating point Arithmetic.</b>	3 hours
<b>Boolean Algebra</b>	
<b>Fundamentals of Boolean Expression:</b> Definition of Switching Algebra, Basic properties of Switching Algebra, Huntington's Postulates, Basic logic gates (AND, OR, NOT), De-Morgan's Theorem, Universal Logic gates (NAND & NOR), XOR and others, Minterm,	4 hours

Maxterm, Minimization of Boolean Functions using Karnaugh-Map up to four (4) variables, Two level and multilevel implementation using logic gates, simplification of logic expressions.	
<b>Combinational Circuits</b>	
<b>Adder &amp; Subtractor:</b> Half adders (2-bit), half Subtractor (2-bit), Full Adder (3-bit), Full Subtractor (3-bit) realization using logic gates, Carry Look Ahead adders, BCD adder, 1's and 2's complement adders/subtractor unit using 4-bit parallel adders.	5 hours
<b>Data Selector/Multiplexer:</b> Realization of multiplexers (4 to 1 and 8 to 1) using logical gates, expansion (Cascading), realization of AND, OR and NOT using multiplexers, realization of different Boolean expressions (SOP) using multiplexers.	5 hours
<b>Data Distributor:</b> De-multiplexer, Cascading, realization of various functions.	2 hours
<b>Encoders:</b> Realization of simple and priority encoders using basic and universal logic gates.	2 hours
<b>Chip Selector/Minterm Generator:</b> Realization of decoders using logic gates, function realization, BCD Decoders, Seven Segment display and decoders, cascading.	3 hours
<b>Parity bit, Code Converters and magnitude comparators:</b> Parity bit generator/checker, Gray to binary code, binary to Gray code and Gray to Excess-3 code converter, 2 & 3 bit magnitude comparators.	2 hours
<b>Sequential Circuits</b>	
<b>Latch &amp; Flip-Flops:</b> Basic Set/Reset (SR) Latch using NAND and NOR gates, Gated S-R latches, Gated D Latch, Gated J-K Latch, race around condition, Master-Slave J-K flip flop, negative and positive clock edge detector circuits, edge triggered SR, D, JK, and T flip flop, flip-flop Conversions.	5 hours
<b>Registers:</b> Serial Input Serial Output (SISO), Serial Input Parallel Output (SIPO), Parallel input Serial Output (PISO), Parallel Input Parallel Output (PIPO), Universal Shift Registers.	3 hours
<b>Counters:</b> <b>Asynchronous Counter</b> UP/DOWN Counters, Mod - N Counters, BCD Counter (Counter Construction using J-K and T Flip Flops).	4 hours
<b>Synchronous Counter:</b> UP/DOWN Counters, Mod-N Counters, Ring & Johnson Counters.	3 hours

<b>Integrated Circuits (Qualitative Study):</b> DTL, TTL: Concepts of Fan in & out, TTL NOT, TTL NAND & NOR, NMOS, PMOS, CMOS, <b>IC fabrication (Concepts only):</b> SSI, MSI, LSI, VLSI, ULSI.	2 hours
--	---------

**Core Course/DSE, CMSA- Practical: Computer Fundamentals and Digital Logic Lab, Semester – 1, Credits - 01, Contact hours - 30.**

**Combinational Circuits**

1. Study and prove De-Morgan's Theorem.
2. Realization of Universal functions using NAND and NOR gates.
3. Implementation different functions (SOP, POS) using digital logic gates.
4. Implementation of half (2-bit) and full adder (3-bit) using basic (AND, OR and NOT) and Universal logic gates (NAND & NOR).
5. Design 4 to 1 multiplexer using basic or Universal logic gates and implement half and full adder/subtractor.
6. Design and implement half and full adder/subtractor and other functions using multiplexers 74151/74153 and other necessary logic gates.
7. Cascading of Multiplexers.
8. Design 2 to 4 decoder using basic or universal logic gates, study 74138 or 74139 and implement half and full Adder/Subtractor and other functions.
9. Design a display unit using Common anode or cathode seven segment display and decoders (7446/7447/7448)
10. Design and implement 4-input 3-output (one output as valid input indicator) priority encoder using basic (AND, OR & NOT) logic gates.
11. Design a parity generator and checker using basic logic gates.

**Sequential Circuits**

1. Realization of SR, D, JK Clocked/Gated, Level Triggered flip-flop using logic gates.
2. Master Slave flip-flop using discrete digital logic gates.
3. Conversion of flip-flops: D to JK, JK to D, JK to T, SR to JK, SR to D Flip-flop.
4. Design asynchronous counters MOD-n (upto 4 bits) UP/ DOWN.
5. Construction Synchronous UP/Down Counter (maximum 4 bits).

**Note:** The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

**Recommended Books**

1. Digital Fundamentals, 11th Edition by Pearson Eleventh Edition, Thomas L. Floyd.
  2. Digital Logic and Computer Design, M Morris Mano, Pearson.
  3. Digital Electronics, Principles, Devices and Applications, Anil K. Maini, John Wiley & sons.
  4. Digital Principles and Applications, Leach, Malvino, Saha, Tata McGraw Hill Education.
  5. Digital Systems, Principal and Applications, Widmer, Moss and Tocci, Pearson.
-

**CMSA- Theory: Data visualization using spreadsheet**  
**SEC-1, Theory, Semester – 1, Credits - 03, Contact hours - 45.**

**Course Description**

This Skill Enhancement Course (SEC) provides a comprehensive introduction to essential concepts and practical skills required for proficient utilization of spreadsheets. Students will gain proficiency in data management, visualization, analysis, and presentation using a widely-used open source spreadsheet software application such as Open Office, Libre Office, or Google Spreadsheets. Through this course, students will acquire the ability to proficiently create, format, manipulate, and analyze data within spreadsheets to meet a diverse range of needs.

**Course Objectives**

1. The purpose and potential applications of spreadsheets.
2. Create, format, and modify spreadsheets.
3. Use of formulas, functions, and calculations to perform data visualization.
4. Understanding and utilization of advanced spreadsheet features such as data validation, conditional formatting, and pivot tables.
5. Design visually appealing charts and graphs to represent data.
6. Collaborate and share spreadsheets with others.
7. Apply spreadsheet skills to real-world scenarios and problem-solving.
8. Role of spreadsheets in data analysis.
9. Import, clean, and transform data for analysis.
10. Applicability of statistical and mathematical functions for data visualization.
11. Advanced features and tools for data visualization.
12. Perform exploratory data analysis and identify patterns and trends.
13. Create informative reports and summaries based on data analysis.
14. Apply data analysis techniques to real-world problems.

Description	Teaching hours
<p><b>Introduction to Spreadsheets</b>            Spreadsheets and their applications, overview of spreadsheet software (e.g., Open office, Google Sheets), navigating the spreadsheet interface, entering and editing data in cells saving, opening, and closing spreadsheet files.</p>	3 hours
<p><b>Formatting and Organizing Data</b>            Formatting cells (e.g., font, alignment, and borders), adjusting column width and row height, using cell styles and themes, working with multiple worksheets, sorting and filtering data.</p>	3 hours
<p><b>Formulas and Functions</b>            Understanding formulas and cell references, basic mathematical operations, using common functions (e.g., SUM, AVERAGE, COUNT), applying absolute and relative cell references, nesting functions</p>	3 hours
<p><b>Data Analysis and Manipulation</b>            Working with text functions for data cleaning, Splitting and combining data, Data normalization and standardization, working with ranges and named ranges, conditional</p>	3 hours

formatting, data validation and error checking, using logical functions (e.g., IF, AND, OR), sorting and filtering data.	
<b>Advanced Spreadsheet Features</b> Creating and managing tables, creating and modifying pivot tables, using lookup functions (e.g., VLOOKUP, HLOOKUP), working with charts and graphs, importing and exporting data.	4 hours
<b>Collaboration and Sharing</b> Protecting worksheets and workbooks, sharing spreadsheets with others, tracking changes and commenting, collaborating in real-time, using version history and revision control.	4 hours
<b>Statistical Functions and Analysis</b> Descriptive statistics (mean, median, mode, variance, etc.), Calculating measures of central tendency and dispersion, Correlation and regression analysis, Hypothesis testing and confidence intervals, Analysis of variance (ANOVA).	5 hours
<b>Pivot Tables and Data Aggregation</b> Creating pivot tables for data summarization, grouping and aggregating data by categories, Applying filters and slicers to pivot tables, calculating calculated fields and items.	4 hours
<b>Advanced Data Visualization</b> Creating charts and graphs for data representation, Customizing chart elements (titles, axes, legends), Using sparklines and data bars for visual analysis, Creating interactive dashboards, Incorporating trendlines and forecasting in charts.	5 hours
<b>Exploratory Data Analysis</b> Identifying patterns and outliers in data, Creating histograms and box plots, Using conditional formatting for data visualization, Data segmentation and drill-down analysis, Applying data validation rules for data integrity.	3 hours
<b>Advanced Analysis Techniques</b> Using goal seek and solver for optimization problems, Performing "what-if" analysis with data tables, Simulating data using random number functions, Monte Carlo simulation for risk analysis. Creating scenario analysis models	3 hours
<b>Reporting and Presentation of Results</b> Designing informative reports and summaries, Creating interactive dashboards for data presentation, Data visualization best practices, Documenting data analysis processes Presenting findings to stakeholders.	3 hours

### **CMSA- Practical - Data visualization using spreadsheet**

#### **SEC, Laboratory, Semester – 1, Credits - 01, Contact hours - 30.**

1. Create a personal budget spreadsheet that tracks income, expenses, and savings over a specified period. Use formulas and functions to calculate totals, percentages, and remaining balances.
2. A dataset containing sales data for a company be provided. Create a spreadsheet that calculates monthly sales totals, identifies top-selling products, and visualizes sales trends using line charts or bar graphs. Use conditional formatting to highlight exceptional sales performances.

3. Design a grade book spreadsheet that calculates students' final grades based on assignments, exams, and participation. Incorporate weighted grading systems, formulas for calculating averages, and conditional formatting to indicate performance levels. Generate reports to track individual student progress.
4. Create a spreadsheet that tracks inventory for a hypothetical business. Include columns for item names, quantities, prices, and total values. Use formulas to automatically update inventory totals, generate alerts for low stock, and create visualizations to represent inventory levels over time.
5. Loan parameters, such as principal amount, interest rate, and loan term to be provided. Create a spreadsheet that calculates monthly loan payments, remaining balances, and interest paid over time using appropriate formulas. Create a chart to visualize the loan's repayment schedule.
6. Dataset to be provided which will allow various data analysis tasks using spreadsheets. Calculation of summary statistics, sorting and filtering data, creating pivot tables for deeper insights, and generation of charts or graphs to visualize patterns or trends within the data.
7. A dataset to be selected (e.g., stock prices, weather data, population growth, etc) and create line charts or area charts to visualize trends over time. Students should choose appropriate chart types, label axes, and add titles and legends to make the visualization clear and informative.
8. A dataset containing information about different products or variables (e.g., sales data, customer satisfaction ratings) to be provided and following to be done; create bar charts or column charts to compare the performance or rankings of the items. Use color, data labels, and chart elements to enhance the visual comparison.
9. Design an interactive dashboard using a spreadsheet. Combine various chart types, slicers, and drop-down menus to allow users to explore and interact with the data dynamically. Create an intuitive and user-friendly interface.
10. A dataset containing time-series data for multiple variables (e.g., monthly sales data for different products) to be provided and the following task to be performed; to create a combo chart with lines and columns to compare the trends of the variables and identify any relationships or patterns.
11. To create a unique visualization using advanced spreadsheet features and tools. For example, an experiment with sparklines, radar charts, or treemaps to represent specific types of data or explore innovative ways to visualize information.

**Note:** The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

### **Recommended Text books**

1. Data Analysis and Decision Making with Microsoft Excel" by S. Christian Albright.
2. Microsoft Excel 2019 Data Analysis and Business Modeling, Sixth Edition, Wayne L. Winston, Pearson education.
3. Excel 2019 Bible, Michael Alexander, 11<sup>th</sup> edition, Wiley.



4. Microsoft Office 2019 for Dummies, Wallace Wang, Wiley.

### Recommended Application Software

1. Google Spreadsheets
2. Libre Office
3. Open Office.

Semester - II				
Paper	Paper type	Paper name	Credit	Contact hours
DSC/CC-2	Theory	Problem Solving using C	3	45
	Practical	Problem Solving using C Lab	1	30
SEC – 2	Theory	Web Development	3	45
	Practical	Web Development Lab	1	30

### PROGRAMMING AND PROBLEM SOLVING THROUGH ‘C’ LANGUAGE

#### Objective of the Course

The objectives of this course are to make the student understand programming language, programming, concepts of Loops, reading a set of Data, stepwise refinement, Functions, Control structure, Arrays. After completion of this course the student is expected to analyze the real life problem and write a program in ‘C’ language to solve the problem. The main emphasis of the course will be on problem solving aspect i.e. developing proper algorithms.

After completion of the course the student will be able to;

1. Develop efficient algorithms for solving a problem.
2. Use the various constructs of a programming language viz. conditional, iteration and recursion.
3. Implement the algorithms in “C” language.
4. Use simple data structures like arrays, stacks and linked list in solving problems.
5. Handling File in “C”.

#### Outline of Course

S. No.	Topic	Minimum number of hours
1	Introduction to Programming	03
2	Algorithm/ Flowchart for Problem Solving	06
3	Introduction to ‘C’ Language	02
4	Conditional Statements and Loops	05
5	Arrays	05
6	Functions	04
7	Storage Classes	02
8	Structures and Unions	05
9	Pointers	05
10	Self-Referential Structures and Linked Lists	04
11	File Processing	02
12	Organizing C Projects	02
Lectures = 45		
Practical/tutorials = 30, Total = 75		

## Detailed Syllabus

Description	Teaching hours
<b>Introduction to Programming</b> The Basic Model of Computation, Algorithms, Flow-charts, Programming Languages, Compiler, Interpreter, Assembler, Linker and Loader, Testing and Debugging, Documentation.	06 hours
<b>Algorithms/ Flowchart for Problem Solving</b> Exchanging values of two variables, summation of a set of numbers, Decimal Base to Binary Base conversion, Reversing digits of an integer, GCD (Greatest Common Division) of two numbers, Test whether a number is prime, Organize numbers in ascending order using Bubble Sort, Find integer square root of a number, factorial computation, Fibonacci sequence, Evaluate 'sin x' as sum of a series, Reverse order of elements of an array, Find largest number in an array, Print elements of upper triangular matrix, multiplication of two matrices, Evaluate a Polynomial.	06 hours
<b>Introduction to 'C' Language</b> Character set, Variables, Identifiers and their nomenclature, Built-in Data Types, Variable Declaration, Arithmetic operators and Expressions, Constants and Literals, Simple assignment statement, Basic input/output statement, Simple 'C' programs.	02 hours
<b>Conditional Statements and Loops</b> Decision making within a program, Conditions, Relational Operators, Logical Connectives, if statement, if-else statement, Loops: while loop, do while, for loop, Nested structure, Infinite loops, Switch-case, break, continue statement, Structured Programming.	05 hours
<b>Arrays</b> One dimensional arrays: Array manipulation; Searching, Insertion, Deletion of an element from an array; Finding the largest/smallest element in an array; Two dimensional arrays, Addition/Multiplication of two matrices, Transpose of a square matrix; Null terminated strings as array of characters, Standard library string functions	05 hours
<b>Functions</b> Top-down approach of problem solving, Modular programming and functions, Standard Library of C functions, Prototype of a function: Formal parameter list, Return Type, Function call, Block structure, Passing arguments to a Function: call by reference, call by value, Recursive Functions, arrays as function arguments.	04 hours
<b>Storage Classes</b> Scope and extent, Storage Classes in a single source file: auto, extern and static, register, Storage Classes in a multiple source files: extern and static	02 hours
<b>Structures and Unions</b> Structure variables, initialization, structure assignment, nested structure, structures and functions, structures and arrays: arrays of structures, structures containing arrays, unions	05 hours
<b>Pointers</b> Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, Array of Pointers, pointer to an array, pointers and structures, dynamic memory allocation.	05 hours

<b>Self-Referential Structures and Linked Lists</b> Creation of a singly connected linked list, Traversing a linked list, Insertion into a linked list, Deletion from a linked list	02 hours
<b>File Processing</b> Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file, Appending to a file.	02 hours
Organizing C projects, working with multiple source directories, makefiles.	

### Recommended books main reading

1. Byron S Gottfried “Programming with C” Second edition, Tata McGrawhill, 2007 (Paper back)
2. R.G. Dromey, “How to solve it by Computer”, Pearson Education, 2008.
3. Kanetkar Y, “Let us C”, BPB Publications, 2007.
4. Hanly J R & Koffman E.B, “Problem Solving and Programm design in C”, Pearson Education, 2009.
5. Kashi Nath Dey and Samir Bandyopadhayay “C Programming Essentials” Pearson India Education, 2010.

### Supplementary reading.

1. E. Balagurusamy, “Programming with ANSI-C”, Fourth Edition,2008, Tata McGraw Hill.
2. Venugopal K. R and Prasad S. R, “Mastering ‘C’”, Third Edition, 2008, Tata McGraw Hill.
3. B.W. Kernighan & D. M. Ritchie, “The C Programming Language”, Second Edition, 2001, Pearson education.
4. ISRD Group, “Programming and Problem Solving Using C”, Tata McGraw Hill,2008.
5. Pradip Dey , Manas Ghosh, “Programming in C”, Oxford University Press, 2007.

## CMSA- Theory: Web development

**SEC, Theory, Semester – 2, Credits - 03, Contact hours - 45.**

### Course Description

This course provides an introduction to web development using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). Students will learn the core concepts and practical skills needed to create and style web pages. The course covers the fundamentals of HTML structure, CSS styling properties, and responsive web design principles.

### Course Objectives

1. Understanding the basics of web development and the role of HTML and CSS.
2. Create well-structured HTML documents using proper tags and elements.
3. Apply CSS to style web pages, including layout, typography, colors, and images.
4. Implement responsive design techniques to ensure optimal display on different devices.
5. Incorporate multimedia elements, such as images, videos, and audio, into web pages.
6. Understand best practices for organizing and maintaining code in web development projects.
7. Develop and deploy a basic website using HTML and CSS.

Description	Teaching hours
<b>Introduction to Web Development</b> Overview of web technologies and the role of HTML and CSS, understanding the structure of a web page, introduction to web browsers and developer tools.	3 hours
<b>HTML Fundamentals</b> Introduction to HTML tags and elements, creating headings, paragraphs, lists, and links, working with images and multimedia content, creating forms for user input.	3 hours
<b>CSS Basics</b> Introduction to CSS and its role in web page styling, selectors, properties, and values, applying inline, internal, and external style sheets, formatting text, backgrounds, and borders.	3 hours
<b>CSS Layout and Box Model</b> Understanding the box model and its impact on layout, working with margins, padding, and borders, positioning elements using floats, positioning properties, and flexbox, creating responsive layouts with media queries.	3 hours
<b>Typography and Colors</b> Styling text with fonts, sizes, weights, and styles, formatting text using CSS properties, understanding color models and applying colors to elements.	4 hours
<b>Images and Multimedia</b> Working with images: sizing, aligning, and optimizing, incorporating videos and audio into web pages, implementing responsive images and media.	4 hours
<b>CSS Selectors and Specificity</b> Understanding CSS selectors and specificity, applying styles to specific elements and classes, using pseudo-classes and pseudo-elements.	5 hours
<b>Responsive Web Design</b> Introduction to responsive design principles, creating fluid layouts using CSS media queries, adapting web pages for different screen sizes and devices.	4 hours
<b>CSS Frameworks and Libraries</b> Overview of popular CSS frameworks (e.g., Bootstrap, Foundation), using pre-built CSS components and grids, customizing and integrating CSS frameworks into web projects.	5 hours
<b>Web Development Best Practices</b> Organizing and structuring code files and directories, validating HTML and CSS code, optimizing web pages for performance, introduction to version control with Git.	3 hours
<b>Building and Deploying a Website</b> Planning and designing a basic website structure, Implementing HTML and CSS to create the website, testing and debugging the website across different browsers, deploying the website to a local host/web server	6 hours

## CMSA- Web development

**SEC, Laboratory, Semester – 2, Credits - 01, Contact hours - 30.**

1. Creating a personal portfolio website using HTML and CSS. There should be sections for an about me, projects, skills, and contact information's. Using CSS to style the layout, typography, and colors to create a visually appealing and professional-looking portfolio.

2. To design a responsive website that adapts to different screen sizes. They should create a layout that adjusts fluidly using CSS media queries and responsive design techniques.
3. To create a product landing page for a fictional product or an existing one. HTML to be used to structure the page and CSS to style the layout, typography, buttons, and images. Main focus to be on creating an engaging page that effectively showcases the chosen product.
4. To incorporate CSS animation effects into a web page. Use CSS transitions, transforms, and keyframe animations to add interactive and engaging elements to the website. Create animations for hover effects, scrolling effects, image sliders, or menu transitions.
5. Redesign an existing website using HTML and CSS. Analyze the original design and propose improvements to the layout, typography, color scheme, and overall user experience.
6. Create a webpage layout using CSS Flexbox or CSS Grid. Design a responsive layout that organizes content in a visually appealing way. Experiment can be performed with different grid or flexbox properties to create flexible and responsive designs.
7. To design and style an interactive form using HTML and CSS. They should incorporate various form elements such as text inputs, checkboxes, radio buttons, and select dropdowns. Apply CSS styling to improve the form's visual appearance and user experience.

**Note:** The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

### **Recommended books**

1. Mastering HTML, CSS & Java Script Web Publishing, Laura Lemay, Rafe Colburn, Jennifer Kyrnin, BPB Publication.
2. Web designing and development, Satish Jain, BPB Publications.
3. HTML & CSS: The complete reference, Thomas Powell, McGraw Hill education.