

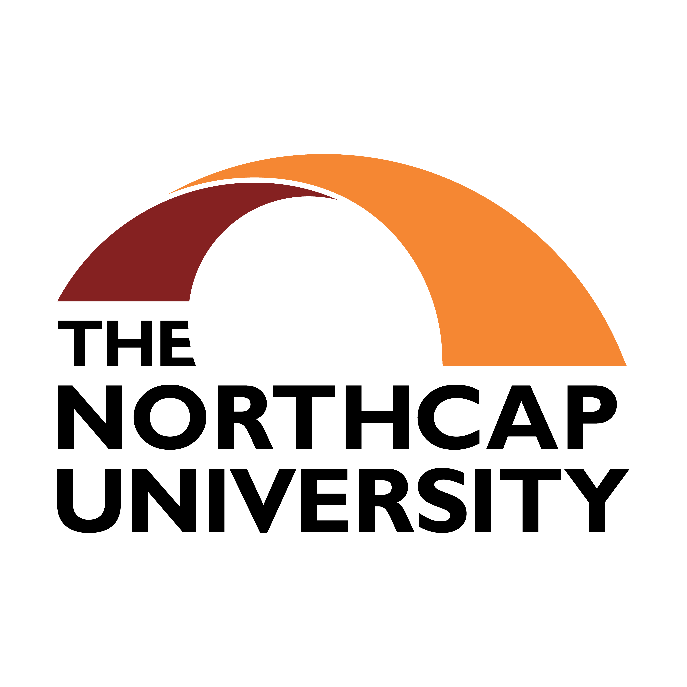
|  |  |
| --- | --- |
| **Web Framework** | |
| Lab Manual | |
| **Department of Computer Science and Engineering**  **The NorthCap University, Gurugram** | |
|  |

**Web Framework**

**CSL253**

**Dr. Sandeep Singh**

**Dr. Neeti Kashyap**



Department of Computer Science and Engineering

NorthCap University, Gurugram- 122001, India

Session 2022-23

*Published by:*

**School of Engineering and Technology**

**Department of Computer Science & Engineering**

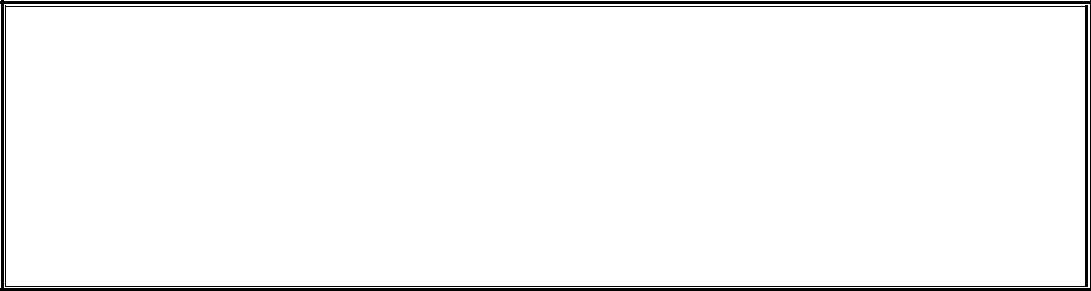
**The NorthCap University Gurugram**

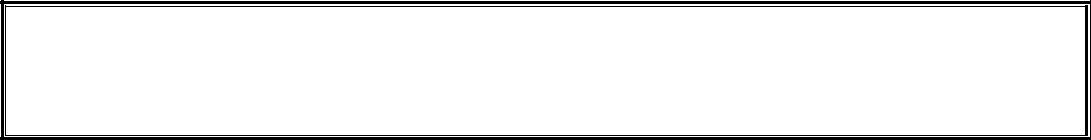
**• Laboratory Manual is for Internal Circulation only**

**© Copyright Reserved**

*No part of this Practical Record Book may be*

*reproduced, used, stored without prior permission of The NorthCap University*



Copying or facilitating copying of lab work comes under cheating and is considered as use of unfair means. Students indulging in copying or facilitating copying shall be awarded zero marks for that particular experiment. Frequent cases of copying may lead to disciplinary action. Attendance in lab classes is mandatory.

Labs are open up to 7 PM upon request. Students are encouraged to make full use of labs beyond normal lab hours.

**PREFACE**

Web Framework (CSL253) Lab Manual is designed to meet the course and program requirements of the NCU curriculum for B.Tech III year students of CSE branch. The concept of the lab work is to give brief practical experience for basic lab skills to students. It provides the space and scope for self-study so that students can come up with new and creative ideas.

The Lab manual is written on the basis of “teach yourself pattern” and expected that students who come with proper preparation should be able to perform the experiments without any difficulty. Brief introduction to each experiment with information about self-study material is provided. The laboratory exercises will include HTML tags, CSS2, CSS3, Bootstrap, type of scripting, client side scripting with JavaScript along-with security flaws. Further, it includes server side scripting with database connectivity and Angular framework. Groups of two shall be created for the students to work on the assigned project. Students are expected to come thoroughly prepared for the lab. General disciplines, safety guidelines and report writing are also discussed.

The lab manual is a part of the curriculum for The NorthCap University, Gurugram. Teacher’s copy of the experimental results and answer for the questions are available as sample guidelines.

We hope that lab manual would be useful to students of CSE, ECE, ME and Civil branches and author requests the readers to kindly forward their suggestions / constructive criticism for further improvement of the work book.

Author expresses deep gratitude to Members, Governing Body-NCU for encouragement and motivation.

**Authors**

**The NorthCap University**

**Gurugram, India**

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTENTS** | | | |
| S.N. | **Details** | Page No. | |
|  | **Syllabus** | 6 | |
| 1 | **Introduction** | 9 | |
| 2 | **Lab Requirement** | 10 |
| 3 | **General Instructions** | 11 |
| 4 | **List of Experiments** | 13 |
| 5 | **List of Flip Assignment** | 16 |
| 6 | **List of Projects** | 17 |
| 7 | **Rubrics** | 18 |
| 8 | **Annexure 1 (Format of Lab Report)** | 19 |
| 9 | **Annexure 2 (Format of Project Report)** | 20 |

**SYLLABUS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Department: | | **Department of CSE & IT** | | |
| 1. **Course Name:**   **Web Framework** | | 1. **Course Code** | 1. **L-T- P** | 1. **Credits** |
| **Code: CSL253** | 2 - 0 - 4 | 4 |
| 1. **Type of Course (Check one):** | | Programme Core Programme Elective Open Elective  **✓** | | |
| **✓**   1. **Frequency of offering (check one):** Odd Even Either Sem. Every Sem. | | | | |
| 1. **Brief Syllabus:**   This course is designed for beginners who have little or no programming expertise to develop interactive web pages with minimum effort. It covers basic HTML tags, CSS2, CSS3, Bootstrap, type of scripting, client side scripting with JavaScript along-with security flaws. Further, it includes server side scripting with database connectivity and Angular framework. By the end of this course, students acquire knowledge and Skills for creation of Web Site considering both client- and server-side Programming, learn database connectivity by JavaScript, perform create, read, update & delete operations in a Website using JavaScript and importance of single page application and technology behind it. | | | | |
| 1. **Total lecture and Practical Hours for this course**   30 Lectures 60 labs | | | | |
| 1. **Course Outcomes (COs)**   Possible usefulness of this course after its completion i.e. how this course will be practically useful to him once it is completed | | | | |
| **CO1** | Design a basic page by using HTML5, CSS2and CSS3 . | | | |
| **CO2** | Design a responsive website by using Bootstrap. | | | |
| **CO3** | Build Dynamic web pages using server side JavaScript Programming and Database | | | |
| **CO4** | Gain confidence to create dynamic website on real world problems. | | | |
| **CO5** | Designing single page application using Angular framework | | | |
| 1. **UNIT WISE DETAILS No. of Units: 6** | | | | |
| **Unit :I HTML** **Hours:3**  Introducing HTML4, HTML5, new features added in HTML5, Semantic Elements,Form Validations,Multimedia, flexbox | | | | |
| **Unit:II CSS2.0 and CSS3**  **Hours:3**  Introduction, Types of StyleSheet(Inline, Internal, External) ,CSS Selectors, Background Property, Margin, Padding, CSS BoxModel, Position, Overflow, Float, Display, Introduction, Advanced Selectors, Dynamic Selector Box Model, ColorModel, Transformation, Perspective, Animation, Flexbox, Introduction of Bootstrap, Page Layout, UI Components ,Responsive Web Design | | | | |
| **Unit :IIIJavaScript** **Hours:8**  Introduction to Client Side Scripting, Introduction to Java Script, DOM, Variables in JSOperators in JS, Conditions Statements, Java Script Loops, Java Script Arrays, Java Script Objects, Traversing Objects Traversing Objects,JavaScript Function,JavaScript Adding and Deleting Elements,JavaScript Methods,Mini Project on Salary Calculator, Cookies Management, Session Management | | | | |
| **Unit :IV Introduction to Node.JS and TypeScript** **Hours:3**  Installation, package manager, callbacks, event loop, os path, query string, URL, DNS, Net, UDP, process: child processes, buffers, streams, file systems, global objects, web modules.  Operators: mathematical operator, conditional operators, comparison operators, relational operator, working with arrays, list, object: object declaration, object literals, defining classes, creating and using modules. TypeScript features: type annotation, tuples, indexable types, access modifiers. | | | | |
| **Unit :V Database Connectivity with MySQL**  **Hours:3**  Introduction Use of MySQL in Node, Create Connection, Query a Database, Creating a Database, Creating a Table Insert into Table, Selecting from a Table, Select with a Filter, MySQL Order By, Delete Record, Update Table, Drop Table. | | | | |
| **Unit :VI Angular basics and project building using angular** **Hours: 10**  MVF framework, components, data binding, event binding, directives and pipes, routing, components life cycle hooks, Angular forms, Services, rxjs operators, REST API implement.  Installing npm package, preparing the RESTful web services, folder structure, HTML file, updating root component, root module, bootstrap file, model classes, data source preparation, model repository, feature module. Adding components and modules in the project. | | | | |
| 1. **Guided Project (No. of Hours 10):CRUD App and 7 minute exercise**   **Unguided Project (No. of Hours):NA**  **(Specify NA if not applicable)** | | | | |
| 1. **Brief Description of Self-learning component by students (through books/resource material etc.): NA** | | | | |
| 1. **Books Recommended:**   **Text Books**: [Marijn Haverbeke](https://www.amazon.com/Marijn-Haverbeke/e/B00409LR2W/ref=dp_byline_cont_book_1), “Eloquent JavaScript”, No Starch Press, 3rd Edition,2018. **Reference Books**: Thomas Powell, “HTML & CSS: The Complete Reference”, McGraw-Hill ,5th Edition,2017. **Reference websites:**  [**www.w3schools.com**](https://www.w3schools.com/js/)  [**https://developer.mozilla.org**](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5) | | | | |

1. **INTRODUCTION**

That ‘learning is a continuous process’ cannot be over emphasized. It is a more of learning practical concepts with little theoretical knowledge. Thus practical makes an integral part of a learning process.

The purpose of conducting experiments can be stated as follows:

* To familiarize the students with the basic concepts of web designing, different tools available for interactive web page designing, responsive pages, scripting languages – JavaScript & TypeScript and Single Page web application designing.
* Observing basic structure and importance of Angular framework.
* Backend database support in current scenario.
* Hands on experience on the single page web application designing.

1. **LAB REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Requirements** | **Details** |
| **1** | **Software Requirements** | Web browser – Firefox/ google Chrome, text editor – VS Code, Node Package Manager |
| **2** | **Operating System** | Windows 10/ Linux OS – 64 bit |
| **3** | **Hardware Requirements** | Windows/ Linux multimedia workstation: Intel 64 or AMD Athlon 64, or AMD Opteron/ Ryzen processor  8 GB RAM  80 GB hard disk space  UHD/ FHD display unit |
| **4** | **Required Bandwidth** | Good connectivity above 15Mbps |

1. **GENERAL INSTRUCTIONS** 
   1. **General discipline in the lab**
   * Students must turn up in time and contact concerned faculty for the experiment they are supposed to perform.
   * Students will not be allowed to enter late in the lab.
   * Students will not leave the class till the period is over.
   * Students should come prepared for their experiment.
   * Experimental results should be entered in the lab report format and certified/signed by concerned faculty/ lab Instructor.
   * Students must get the connection of the hardware setup verified before switching on the power supply.
   * Students should maintain silence while performing the experiments. If any necessity arises for discussion amongst them, they should discuss with a very low pitch without disturbing the adjacent groups.
   * Violating the above code of conduct may attract disciplinary action.
   * Damaging lab equipment or removing any component from the lab may invite penalties and strict disciplinary action.
   1. **Attendance**

* Attendance in the lab class is compulsory.
* Students should not attend a different lab group/section other than the one assigned at the beginning of the session.
* On account of illness or some family problems, if a student misses his/her lab classes, he/she may be assigned a different group to make up the losses in consultation with the concerned faculty / lab instructor. Or he/she may work in the lab during spare/extra hours to complete the experiment. No attendance will be granted for such case**.**
  1. **Preparation and Performance**
* Students should come to the lab thoroughly prepared on the experiments they are assigned to perform on that day. Brief introduction to each experiment with information about self-study reference is provided on LMS.
* Students must bring the lab report during each practical class with written records of thelast experiments performed complete in all respect.
* Each student is required to write a complete report of the experiment he has performed and bring to lab class for evaluation in the next working lab. Sufficient space in work book is provided for independent writing of theory, observation, calculation and conclusion.
* Students should follow the Zero tolerance policy for copying / plagiarism. Zero marks will be awarded if found copied. If caught further, it will lead to disciplinary action.
* Refer **Annexure 1** for Lab Report Format.

1. **LIST OF EXPERIMENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Title of the Experiment** | **Software used** | **Unit covered** | **CO**  **Covered** | **Time Required** |
|  | Create a webpage for the assigned theme, according to the template by using HTML5 | VS Code and Chrome web browser | 1 | CO1 | 2 hrs |
|  | Create a webpage for the assigned theme, according to the template by using HTML5 and CSS2 | VS Code and Chrome web browser | 1 | CO1 | 2 hrs |
|  | Create a webpage for the assigned theme, according to the template by using HTML5,CSS2 and CSS3 | VS Code and Chrome web browser | 1 | CO1 | 2 hrs |
|  | Create a front end of flip kart by using HTML5,CSS2, CSS3  and Bootstrap. | VS Code and Chrome web browser | 2 | CO2 | 2 hrs |
|  | Basic programs on JavaScript. | VS Code and Chrome web browser | 3 | CO2 | 2 hrs |
|  | Demonstrate the use of javascript to define a web server and API call for an application. | VS Code and Chrome web browser | 3 | CO2 | 2 hrs |
|  | Salary calculator using javascript | VS Code and Chrome web browser | 3 | CO3 | 2 hrs |
|  | **Mini Project on JavaScript**   1. Create a makemytrip online form 2. Create an online form for booking of hotel 3. Create an online job application form | VS Code and Chrome web browser | 3 | CO3 | 6 hrs |
|  | Write the Type Script code to test the syntaxes of the language | Node Package Manager, VS Code and Chrome web browser | 4 | CO3 | 2 hrs |
|  | Implement the class and object example using Type Script | Node Package Manager, VS Code and Chrome web browser | 4 | CO3 | 2 hrs |
|  | Implement the html template and lifecycle-hooks using Angular framework | Node Package Manager, VS Code and Chrome web browser | 5 | CO4 | 2 hrs |
|  | Implement the passing parameters/ messageexample among components in Angular framework | Node Package Manager, VS Code and Chrome web browser | 5 | CO4 | 2 hrs |
|  | Implement the service injection module in Angular framework | Node Package Manager, VS Code and Chrome web browser | 5 | CO4 | 2 hrs |
|  | Demonstrate the importance of routing in page access and show the features available in Angular framework | Node Package Manager, VS Code and Chrome web browser | 6 | CO5 | 2 hrs |
|  | | | | | |
|  | Create a makemytrip online form | VS Code and Chrome web browser | 1, 2 | CO1 -CO2 | 4hrs |
|  | Create an online form for booking of hotel | VS Code and Chrome web browser | 1,2,3 | CO1 -CO3 | 4 hrs |
|  | Create an online job application form | VS Code and Chrome web browser | 1,2,3 | CO1 -CO3 | 4 hrs |
|  | Create a weather API application | Node Package Manager, VS Code and Chrome web browser | 4, 5 | CO4 - Co5 | 4 hrs |
|  | Create a shopping application with Login and shopping cart facility | Node Package Manager, VS Code and Chrome web browser | 1, 2, 3, 4, 5 | CO1 – CO5 | 4 hrs. |

1. **LIST OF FLIP EXPERIMENTS**
2. Generate html page for our own bio data (use all formatting tag).
3. Generate college admission form using html form tag.
4. Create html application for demo of <img> tag.
5. Write a java script to manipulate array.
6. Write a java script for blinking text on the screen.
7. Write a java script for calculating the bill and finding factorial of n numbers.
8. Write a java Script for counting Odd, even and prime numbers from 1 to 100.
9. Write a java script to find factorial of given number.
10. Write a java script program to generate Fibonacci series using do while loop.
11. Write a Java Script using for loop for displaying sum of first 150 numbers.
12. If a=10 and b=20 write a java script to swapping of these two numbers.
13. Write a java script which will greet according to the current timing
14. Write a java script for simple form processing
15. **LIST OF PROJECTS**
16. Online assessment system: This application will help to evaluate the online exam, quiz, minor test etc. It will have the login facility, question upload by admin/ faculty members, schedule to set for assessment. Later on, it will generate the total marks obtained by the user. At the admin panel, it will generate the report to show the marks of all the students.
17. Create an e-commerce platform for a book shop: At admin side, this will facilitate to maintain the inventory and at user side, it will work like an e-commerce web site to purchase the books.
18. Flower booking and delivery service: This application will provide an e-commerce platform to sell and purchase the flowers.
19. Weather recording and monitoring system: The application will provide the graphical view of the weather information for cities towns. The data can be fetch from openweather API. This data needs to be shown in graphs.
20. Online parking slot booking and management system: This application will help to find the parking place in nearby parking lots. The parking attendants will update the empty slots in the application which can be fetch by the users.
21. Online system for waste management: This application will help to manage the waste, trash bins and information to the authority for cleanliness.
22. Online platform for a restaurant food delivery: The application will connect the restaurants and the consumers. It will be for a particular restaurant who will setup the menu and costumers can order and buy the food.
23. Electrical appliances repair booking and home repair service: This application will provide the service to maintenance mechanics and the consumer who need some repair. The mechanics will create their profile and location and also update their current location and status like busy/ free etc. The consumers who needs help will raise the request which will be assigned to a mechanic who is free and located nearby.
24. Toys shop inventory system: It is an MIS based project with web and mobile based application to manage the inventory of a toy shop. Students can add the e-commerce component to facilitate the consumers to purchase toys online.
25. Financial Book System: The application is a virtual form of book keeping. It will be operated by the lender and borrower. The application can provide the printed form of report at any point of time.
26. COVID dashboard: This application will provide the COVID related data which are available in the form for APIs. The application can show the information for any city/ country level since the APIs are available at various sources, example: https://api.covidtracking.com
27. Family tree: This application will provide the information of a family in graphical form. On view will only show the parent and the immediate children and by clicking any of them, the view will change by showing the information of that particular person.
28. Online seminar hall/ auditorium booking system: The application will facilitate the users to book the seminar hall/ auditorium. It will be operated by the user and there will be the admin part who will allow and deny the user request and update the status.
29. Photo Album: The application will help to search a photo and show it as photo album. Now a days, because of handy camera, the drives and full of photos. This application can help to manage the photos. The application will store the pictures along with tags. The tags will be provided by the users at the time of clicking the pictures. These tags will be used as the search keys while searching the pictures in a pool of pictures.
30. **RUBRICS**

|  |  |
| --- | --- |
| **Marks Distribution** | |
| **Continuous Evaluation (50 Marks)** | **Project Evaluations (20 Marks)** |
| Each experiment shall be evaluated for 10 marks and viva at the end of the semester proportional marks shall be awarded out of total 50. | Both the projects shall be evaluated for 10 marks each and at the end of the semester viva will be conducted related to the projects as well as concepts learned in labs and this component carries 20 marks. |
| Following is the breakup of 10 marks for each  **6 Marks**: Observation & conduct of experiment. Teacher may ask questions about experiment.  **2 Marks:** For completing questions given at the end of each experiment  **2 Marks:** For timely submission. |

**Annexure 1**

**Web Framework**

**(CSL 253)**

**Lab Practical Report**



Faculty name: Dr. Neeti Kashyap

Student name: Avtar Singh

Roll No.: 20csu241

Semester: 5th

Group: FSB

**Department of Computer Science and Engineering**

**NorthCap University, Gurugram- 122001, India**

**Session 2022-23**

**INDEX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Experiment** | **Page No.** | **Date of Experiment** | **Date of Submission** | **Marks** | **CO Covered** | **Sign** |
| 1. | Create a webpage for the assigned theme, according to the template using HTML5. |  | 10-8-22 | 10-8-22 |  | 1 |  |
| 2. | Create a webpage for the assigned theme, according to the template by using HTML5 and CSS2. |  | 17-8-22 | 17-8-22 |  | 1 |  |
| 3. | Create a webpage for the assigned theme, according to the template by using HTML5,CSS2 and CSS3 |  | 24-8-22 | 24-8-22 |  | 1 |  |
| 4. | Create a front end of flip kart by using HTML5,CSS2, CSS3and Bootstrap. |  | 5-9-22 | 5-9-22 |  | 1 |  |
| 5. | Basic programs on JavaScript. |  | 12-9-22 | 12-9-22 |  | 2 |  |
| 6 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |
| **9** |  |  |  |  |  |  |  |

**EXPERIMENT NO. 1**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:10/08/2022** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective(s):**  Create a webpage for the assigned theme, according to the template using HTML5**.** |
| **Outcome:** |
| **Problem Statement:**  **Definition of Done**  **DoD 1.** The webpage should be created using the table tag.  **DoD 2.** The header of the webpage should display the theme on which the webpage is being created.  **DoD 3.**The main section should be divided into three sub-sections.  **DoD 4.** Each subsection should have its respective header information.  **DoD 5.** The leftmost section should be created using unordered list, check box, radio buttons, drop down menu and tool tip.  **DoD 6.** The middle section or the main section should contain the login form and different pictures giving the information on assigned theme.  **DoD 7.** The rightmost section should contain the different images along with their description aligned at the bottom of respective images.  **DoD 8.** The webpage should also contain footer representing the contact details for the webpage being created. |
| **Background Study:**  **HTML Element Structure**  A HTML document consists of HTML elements. Let’s take a closer look at the basic structure of an HTML element.  An HTML element consists of the following parts:  Start tag  Attributes  Element body  End tag  Each of these parts are explained throughout the rest of this text.  Element Names + Start and End Tags  HTML elements have a start tag and an end tag. For example, <html> and </html>.  The character < opens a tag and this > closes the tag. The slash-character before the tag name marks it as an end tag.  The element content is written in between this two tags. The element content is also called the element body. Here is an example:  <b>Text bla bla</b>  The example shows a <b> element consisting of a start tag (<b>), element content (Text bla bla), and an end tag (</b>).  The name of the element is written inside both the start and end tags. In the example above, the name of the element is b which is short for "bold". The b element marks its body to be displayed in bold. Later texts will get into more detail about the various elements available.  Empty Elements  Some HTML elements have no element body, meaning you cannot write any text between the start tag and end tag. An example is the br element which puts a line break into the HTML document. Here is an example:  <br>  Empty HTML elements do not need an end tag. You just write the start tag as shown in the example above.  Element Attributes  An HTML element can have attributes. Attributes are embeddes inside the start tag of the HTML element. Here is an example:  <table border="0"> ... </table>  The table element above has an attribute named border. The value of the attribute is 0.  The format of an attribute declaration is:  name="value"  The = and the quotes are part of the attribute declaration, but not part of neither attribute name nor value. You can use both the double or single quote character as quote. Here are two examples:  <table border="0"> ... </table>  <table border='0'> ... </table>  Both of these examples are valid.  It is actually allowed to omit the quotes around the attribute value. Here is an example:  <table border=0> ... </table>  This is the same table element with the same border attribute, but without quotes around the attribute value. If you omit the quotes, the attribute value cannot contain spaces. If the attribute value contains spaces, enclose the attribute value in quotes.  Some attributes have no value. Their presence or absence carries the meaning itself. Here is an example:  <input type="checkbox" checked>  This input element has an attributed named checked which has no value. If the checked attribute is present, it means that the checkbox should be displayed as checked. If the checked attribute is absent, it means that the checkbox should be displayed unchecked.  Nested Elements  Some HTML elements can have other HTML elements nested inside them. Here is an example:  <p>  Mary had a <b>little</b> lamb.  </p>  This example shows a <p> element with a text inside. Part of the text is surrounded by a <b> element. This <b> element is nested inside the <p> element. The <p> element is used around paragraphs of text, and the <b> element is used to mark text to be displayed as bold. Both of these elements will be explained in more detail in later texts.  Make sure that the nesting of elements is done correctly, meaning that the start and end tags are written in the correct order. Here is an example of a text that is both bold and italic:  <p><b><i>Bold italic text</i></b></p>  Notice how the inner element, <i> both starts and ends inside the <b> element. A common mistake is to swap the end tags, for instance like this:  <p><b><i>Bold italic text</b></i></p>  Notice how the end tags of the <i> and <b> elements are now swapped. The <b> element is ended before the <i> element. This is not valid HTML. Most browsers will still display the HTML correctly, but don't rely on it. Make sure your nesting of start and end tags is correct.  Nesting elements is often necessary to achieve more advanced document formatting, as you will see later in this book.  White Space and Line Breaks  The browser ignores extra white spaces and line breaks. White space characters include the space character, tab character, line break character and similar characters that represent blank (white) space in the document. Line breaks are characters that represent new lines in a normal text editor like notepad.  Here are two examples with different amounts of white space characters in:  <p>John is good</p>  <p>  John is good  </p>  These two paragraphs of text will be displayed similarly in the browser. The browser ignores white space before the first non-white-space character in a paragraph. The browser also ignores the line breaks inside the <p> element.  Additionally, the browser ignores the extra white space between the words, displaying only a single space between the words, and it ignores the extra white space and line break after the last word in the paragraph.  A single line break between two words will be displayed as a space. Extra line breaks in between words will be ignored, just like white space characters.  White Space Inside Tags  The browser also ignores extra white space and line breaks at some points inside HTML tags. Here are some examples:  <p > Text </p>  <table  width="100  height="200"  >  </table>  Both of these examples are valid HTML.  It not allowed to have white space between the < character and the element name though. Thus, this is not valid HTML:  < p>  The browser uses the text right after the first < character to determine what HTML element it has encountered. If a white space is found, the browser assumes that it is not an HTML tag, but instead just the < character, which is then just displayed as it is.  Uppercase and Lowercase Characters in Element Names  The HTML element names are case insensitive meaning it does not matter if you write them in uppercase or lowercase. That means, that the following two examples are displayed the same in the browser:  <b>My Text</b>  <B>My Text</B>  In many cases the attribute names and attribute values are also case insensitive.  Personally I always use lowercase characters when writing element names, attribute names and values. I find it easier to read, and easier to write too.  The content inside the element is not case insensitive though. The text inside the <b> element (My Text) will be displayed with uppercase and lowercase characters exactly as you write them. |
| **Question Bank:**   1. *What is a pseudo-class?*   *A pseudo-class is a selector that selects elements that are in a specific state, e.g. they are the first element of their type, or they are being hovered over by the mouse pointer.*   1. *What are The New doctype And Charset In HTML5?*   *To know about the html type and encoded character set*   1. *Which of The HTML5 Elements Support Media Content?*   *HTML5 supports AAC, MP3 and Ogg Vorbis for audio and Ogg Theora, WebM and MPEG-4 for video. Even though HTML5 supports these media formats, however, not every browser supports every format.*   1. ***How do I Remove The Border Around Frames?***   ***Remove border from iframe tag in the webpage could be done by using one of the CSS properties of the iframe tag called frameBorder and set its value to “0”. Syntax: frameBorder = "value"; Note: In the frameBorder property the B letter must be in capital otherwise it will not be recognized by the browser.***   1. ***How Do I Create Frames? What Is A Frameset?***   *The <frameset> tag defines, how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.*   1. ***How Is The Page Structure Of Html 5 Different From Html 4 Or Previous Html?*** 2. ***The main difference between HTML 4 and HTML 5 is that HTML 4 is the 4th and older version of HTML with fewer features and tags while HTML 5 is the extension of HTML4 and the 5th version of HTML which has new and simple features and has multiple new tags.*** 3. ***What Is Html 4 Header And Footer?***   ***The <header> tag in HTML is used to define the header for a document or a section as it contains the information related to the title and heading of the related content. The <header> element is intended to usually contain the section's heading (an h1-h6 element or an <hgroup> element), but this is not required.***   1. ***What Are The Authoring Documents With Html 4.0?***   ***What are the three HTML 4.0 flavors?***  ***To help in the transition from the era of using HTML tags that have evolved in an ad-hoc fashion to control the appearance of Web pages to a new era in which the appearance of Web pages will be controlled entirely by the use of styles sheets, the W3C has formulated three different flavors of HTML 4***   1. ***Which Code Can Be Used To Help Prevent A Page From Being Cached In The Users Browser?***   *You must send Cache-Control: no-store, must-revalidate (some browsers observe no-store and some observe must-revalidate* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Practical 1</title>**

**</head>**

**<body>**

**<table border=1 width=100% align=center>**

**<tr>**

**<td align="center" colspan="3">**

**<br>**

**<i><b>**

**<h1>Food Outlets in HUDA SEC 23A , GURURAM</h1>**

**</b></i>**

**<br>**

**</td>**

**</tr>**

**<tr>**

**<td align="center" width="15%">**

**<h6 size="1000px">Click on the names to visit : </h6>**

**<ul class="menu">**

**<li><a href="http://burmaburma.in/">Burma Burma Restaurant & Tea Room <br>**

**<img src="https://img.freepik.com/premium-photo/big-hamburger-with-double-beef-french-fries\_252907-8.jpg?w=2000"**

**border=1 alt="" srcset="" width="90%" height=90%>**

**</a>**

**</li>**

**<br><br>**

**<li><a**

**href="https://www.google.com/search?tbs=lf:1,lf\_ui:9&tbm=lcl&q=food+outlets+in+gurugram&rflfq=1&num=10&sa=X&ved=2ahUKEwj23eW236n5AhUsZWwGHdrgCh8QjGp6BAgJEAE&biw=1319&bih=657&dpr=1#">delectable**

**cafe <br><img**

**src="https://images.pexels.com/photos/1640777/pexels-photo-1640777.jpeg?cs=srgb&dl=pexels-ella-olsson-1640777.jpg&fm=jpg"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a**

**href="https://www.google.com/search?q=food%20outlets%20in%20gurugram&oq=foo&aqs=edge.0.69i59l2j69i57j0i433i512j0i131i433j0i433i512l2j0i131i433i512j0i433i512.2779j0j4&sourceid=chrome&ie=UTF-8&tbs=lf:1,lf\_ui:9&tbm=lcl&rflfq=1&num=10&rldimm=11159731253079895132&lqi=Chhmb29kIG91dGxldHMgaW4gZ3VydWdyYW1Ihp3ClM-rgIAIWigQABABGAMiGGZvb2Qgb3V0bGV0cyBpbiBndXJ1Z3JhbSoGCAMQABABkgESYnVybWVzZV9yZXN0YXVyYW50mgEkQ2hkRFNVaE5NRzluUzBWSlEwRm5TVVJOZFZBdFN6VkJSUkFCqgEeCggvbS8wMndibRABKhAiDGZvb2Qgb3V0bGV0cygA&phdesc=aYOq4IVZPPw&ved=2ahUKEwiy8bHn4Kn5AhVqSGwGHV1KDwgQvS56BAgDEAE&sa=X&rlst=f#">Take**

**Eat Easy**

**<br><img**

**src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTlqC4ZZdkC6LoR3823iWAp\_lz7hOjrs0bA6C1BkZ5OgNHnJs8QCCrqW1h3\_-rwG3zLXBo&usqp=CAU"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a href="https://stores.haldiram.com/sahara-mall-gurgaon/5305732432003072">Haldiram's <br>**

**<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT1aaC0tZmUlGjwMQgj3U-RDbfTPQn1hae5N-DBPCA6Mh3YIEO5DKaR-\_6fKB6RMNvwJJY&usqp=CAU"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a href="https://www.sodabottleopenerwala.in/cyberhub-gurgaon/">SodaBottle <br> OpenerWala <br>**

**<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQb4arHPf6NEddabnXO86hMblwzcohFzhpswBWO8uHR4gB8kRIBk1qmbDUsk4bkKLiqisk&usqp=CAU"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a**

**href="https://www.google.com/search?q=food%20outlets%20in%20gurugram&oq=foo&aqs=edge.0.69i59l2j69i57j0i433i512j0i131i433j0i433i512l2j0i131i433i512j0i433i512.2779j0j4&sourceid=chrome&ie=UTF-8&tbs=lf:1,lf\_ui:9&tbm=lcl&rflfq=1&num=10&rldimm=11159731253079895132&lqi=Chhmb29kIG91dGxldHMgaW4gZ3VydWdyYW1Ihp3ClM-rgIAIWigQABABGAMiGGZvb2Qgb3V0bGV0cyBpbiBndXJ1Z3JhbSoGCAMQABABkgESYnVybWVzZV9yZXN0YXVyYW50mgEkQ2hkRFNVaE5NRzluUzBWSlEwRm5TVVJOZFZBdFN6VkJSUkFCqgEeCggvbS8wMndibRABKhAiDGZvb2Qgb3V0bGV0cygA&phdesc=aYOq4IVZPPw&ved=2ahUKEwiy8bHn4Kn5AhVqSGwGHV1KDwgQvS56BAgDEAE&sa=X&rlst=f#">Shomils**

**food corner**

**<br>**

**<img src="https://thumbs.dreamstime.com/b/assorted-american-food-top-view-109748438.jpg"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**</ul>**

**</td>**

**<td align="center" width="70%">**

**<table border="1" align="center">**

**<tr>**

**<td>**

**<h1>Notify Me : </h1>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<center>**

**<form action=#><br><br> Your Name: <input**

**type="text" name="name"><br><br> Email Address:**

**<input type="text" name="email"><br><br> Message:**

**<textarea name="message"></textarea><br><br>**

**<br>**

**<br>**

**</center>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<h1>Choose Your Mood : </h1>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<input type="checkbox" id="mood1" name="mood" value="Bike">**

**<label for="mood1"> Want to party </label><br>**

**<input type="checkbox" id="mood2" name="mood" value="Car">**

**<label for="mood2">Sad </label><br>**

**<input type="checkbox" id="mood3" name="mood" value="Boat">**

**<label for="mood3"> Just want to Chill </label><br>**

**<center>**

**<br><br>**

**<input type="submit" value="Submit">**

**</center>**

**</form>**

**</td>**

**</tr>**

**</table>**

**</td>**

**<td align="center" width="15%">**

**<img src="https://images.unsplash.com/photo-1540189549336-e6e99c3679fe?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxleHBsb3JlLWZlZWR8N3x8fGVufDB8fHx8&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**<img src="https://images.unsplash.com/photo-1530554764233-e79e16c91d08?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MTB8fGZhc3QlMjBmb29kfGVufDB8fDB8fA%3D%3D&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**<img src="https://images.unsplash.com/photo-1555939594-58d7cb561ad1?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxleHBsb3JlLWZlZWR8M3x8fGVufDB8fHx8&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**<img src="https://images.unsplash.com/photo-1532980400857-e8d9d275d858?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MTF8fHl1bW15JTIwZm9vZHxlbnwwfHwwfHw%3D&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**</td>**

**</tr>**

**<tr>**

**<td align="center" colspan="3">**

**<hr>**

**<i><b>**

**<font color="red">**

**<h2>Copyright &copy; | Avtar Singh </h2>**

**</font>**

**</b></i>**

**<hr>**

**</td>**

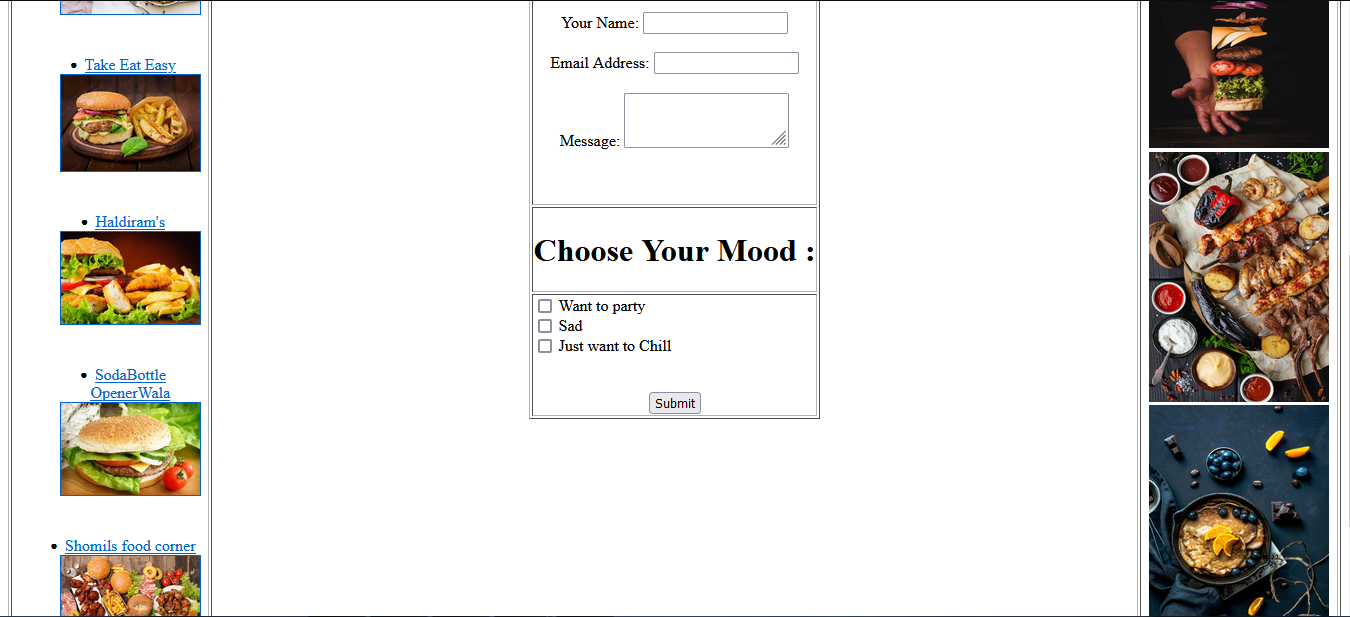
**</tr>**

**</table>**

**</body>**

**</html>**

**OUPUT :**

****

**EXPERIMENT NO. 2**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** **17-8-22** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Create a webpage for the assigned theme, according to the template by using HTML5 and CSS2**.** |
| **Outcome:** |
| **Problem Statement:**  **Definition Of Done :**  **DoD 1.** Page width should be 760px  **DoD 2.** The header height should be 263px  **DoD 3**. Columns: The contents are divided in two columns  **DoD 4**. Left side column contains a login form  **DoD 5.** Right side column contains the navigation bar, main contents paragraph  wise  **DoD 6**. A footer section is to be made to show contact information and important  links  **DoD 7**. A background images should overlap the area of two places  For e.g Background image covering header and left side column, and background image covering side bar and columns |
| **Background Study:**  **CSS Introduction:**  CSS stands for Cascading Style Sheets. It is a language designed to specify the overall appearance of webpages as well as the appearance and structure of the text and elements such as images and buttons on webpages and their layout. Styles can be specified with CSS using internal style sheet definitions which are placed right into HTML/XHTML code or in external files.  **CSS syntax**  There are three parts in the CSS syntax:  selector {property: value}   * selector - The selector is the HTML tag that a style will be applied to. For example, the <p> tag for paragraphs or the <h2> heading tag. * property - The property is an aspect of the HTML tag that will be affected by the stylesheet definition. For example, the background color of a webpage or the color of links. * value - The value that the property of a selector will have. For example, the value of the background color of a webpage can be green or the value of the color of links can be gray.   Example:  body {background-color: gray}  The property and value of a selector are separated by a colon, and sorrounded by curly braces.  If a value is more than one word, put quotes around it.  Example of value being more than one word:  h2 {font-family: "Trebuchet MS"}  If you specify more than one property, each property should be separated with a semicolon.  Example of multiple properties:  body {background-color: gray; color: yellow;}  You can make style definitions more easily readable by specifying each property on a separate line.  Example:  body{ background-color: gray; color: yellow; margin-top: 0; }  **Grouping styles**  You can specify that a group of tags will have the same styles by grouping them together.  Example:  p, h1, h2, h3, h4{ color: blue; font-family: courier; }  In this example, all text declared with the tags <p>, <h1>, <h2>, <h3>, and <h4> will be blue in color and Courier in font. |
| **Question Bank:**   1. ***How Does CSS3 Differ From CSS?***   ***CSS uses an old standard colour format. CSS3 provides a different gradient colour and schemes like RGBA, HSLA, HSL, and other colour gradients. CSS does not have the concept of modules. CSS3 incorporates a new feature where it can group CSS codes into convenient modules.***   1. ***What Are The Different Types Of CSS?***   ***There are three types of CSS which are given below: Inline CSS. Internal or Embedded CSS. External CSS.***   1. ***What Is The Use Of Float Property In CSS?***   *The CSS float property****controls the positioning and formatting of content on the page****. Its most common use is to wrap text around images. However, you can use the float property to wrap any inline elements around a defined HTML element, including lists, paragraphs, divs, spans, tables, iframes, and blockquotes* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**PROGRAM :**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Practical 1</title>**

**<link rel="stylesheet" href="style.css">**

**</head>**

**<body>**

**<table class="outertable" >**

**<tr >**

**<td class = "innertable" align="center" colspan="3">**

**<br>**

**<i><b>**

**<h1>Food Outlets in HUDA SEC 23A , GURURAM</h1>**

**</b></i>**

**<br>**

**</td>**

**</tr>**

**<tr>**

**<td class="setborder" id="tableleftimg" align="center" width="15%">**

**<h6 size="1000px">Click on the names to visit : </h6>**

**<ul class="menu">**

**<li><a href="http://burmaburma.in/">Burma Burma Restaurant & Tea Room <br>**

**<img src="https://img.freepik.com/premium-photo/big-hamburger-with-double-beef-french-fries\_252907-8.jpg?w=2000"**

**border=1 alt="" srcset="" width="90%" height=90%>**

**</a>**

**</li>**

**<br><br>**

**<li><a**

**href="https://www.google.com/search?tbs=lf:1,lf\_ui:9&tbm=lcl&q=food+outlets+in+gurugram&rflfq=1&num=10&sa=X&ved=2ahUKEwj23eW236n5AhUsZWwGHdrgCh8QjGp6BAgJEAE&biw=1319&bih=657&dpr=1#">delectable**

**cafe <br><img**

**src="https://images.pexels.com/photos/1640777/pexels-photo-1640777.jpeg?cs=srgb&dl=pexels-ella-olsson-1640777.jpg&fm=jpg"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a**

**href="https://www.google.com/search?q=food%20outlets%20in%20gurugram&oq=foo&aqs=edge.0.69i59l2j69i57j0i433i512j0i131i433j0i433i512l2j0i131i433i512j0i433i512.2779j0j4&sourceid=chrome&ie=UTF-8&tbs=lf:1,lf\_ui:9&tbm=lcl&rflfq=1&num=10&rldimm=11159731253079895132&lqi=Chhmb29kIG91dGxldHMgaW4gZ3VydWdyYW1Ihp3ClM-rgIAIWigQABABGAMiGGZvb2Qgb3V0bGV0cyBpbiBndXJ1Z3JhbSoGCAMQABABkgESYnVybWVzZV9yZXN0YXVyYW50mgEkQ2hkRFNVaE5NRzluUzBWSlEwRm5TVVJOZFZBdFN6VkJSUkFCqgEeCggvbS8wMndibRABKhAiDGZvb2Qgb3V0bGV0cygA&phdesc=aYOq4IVZPPw&ved=2ahUKEwiy8bHn4Kn5AhVqSGwGHV1KDwgQvS56BAgDEAE&sa=X&rlst=f#">Take**

**Eat Easy**

**<br><img**

**src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTlqC4ZZdkC6LoR3823iWAp\_lz7hOjrs0bA6C1BkZ5OgNHnJs8QCCrqW1h3\_-rwG3zLXBo&usqp=CAU"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a href="https://stores.haldiram.com/sahara-mall-gurgaon/5305732432003072">Haldiram's <br>**

**<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT1aaC0tZmUlGjwMQgj3U-RDbfTPQn1hae5N-DBPCA6Mh3YIEO5DKaR-\_6fKB6RMNvwJJY&usqp=CAU"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a href="https://www.sodabottleopenerwala.in/cyberhub-gurgaon/">SodaBottle <br> OpenerWala <br>**

**<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQb4arHPf6NEddabnXO86hMblwzcohFzhpswBWO8uHR4gB8kRIBk1qmbDUsk4bkKLiqisk&usqp=CAU"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**<br><br>**

**<li><a**

**href="https://www.google.com/search?q=food%20outlets%20in%20gurugram&oq=foo&aqs=edge.0.69i59l2j69i57j0i433i512j0i131i433j0i433i512l2j0i131i433i512j0i433i512.2779j0j4&sourceid=chrome&ie=UTF-8&tbs=lf:1,lf\_ui:9&tbm=lcl&rflfq=1&num=10&rldimm=11159731253079895132&lqi=Chhmb29kIG91dGxldHMgaW4gZ3VydWdyYW1Ihp3ClM-rgIAIWigQABABGAMiGGZvb2Qgb3V0bGV0cyBpbiBndXJ1Z3JhbSoGCAMQABABkgESYnVybWVzZV9yZXN0YXVyYW50mgEkQ2hkRFNVaE5NRzluUzBWSlEwRm5TVVJOZFZBdFN6VkJSUkFCqgEeCggvbS8wMndibRABKhAiDGZvb2Qgb3V0bGV0cygA&phdesc=aYOq4IVZPPw&ved=2ahUKEwiy8bHn4Kn5AhVqSGwGHV1KDwgQvS56BAgDEAE&sa=X&rlst=f#">Shomils**

**food corner**

**<br>**

**<img src="https://thumbs.dreamstime.com/b/assorted-american-food-top-view-109748438.jpg"**

**border=1 alt="" srcset="" width="90%" height=90%></a></li>**

**</ul>**

**</td>**

**<td class="setborder" id="tablecenterimg" align="center" width="70%">**

**<table class="base\_heading" border="1" align="center">**

**<tr>**

**<td>**

**<h1>Notify Me : </h1>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<center>**

**<form action=#><br><br> Your Name: <input**

**type="text" name="name"><br><br> Email Address:**

**<input type="text" name="email"><br><br> Message:**

**<textarea name="message"></textarea><br><br>**

**<br>**

**<br>**

**</center>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<h1 class="base\_heading">Choose Your Mood : </h1>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<input type="checkbox" id="mood1" name="mood" value="Bike">**

**<label for="mood1"> Want to party </label><br>**

**<input type="checkbox" id="mood2" name="mood" value="Car">**

**<label for="mood2">Sad </label><br>**

**<input type="checkbox" id="mood3" name="mood" value="Boat">**

**<label for="mood3"> Just want to Chill </label><br>**

**<center>**

**<br><br>**

**<input type="submit" value="Submit">**

**</center>**

**</form>**

**</td>**

**</tr>**

**</table>**

**</td>**

**<td id="tablerightimg" class="setborder" align="center" width="15%">**

**<img src="https://images.unsplash.com/photo-1540189549336-e6e99c3679fe?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxleHBsb3JlLWZlZWR8N3x8fGVufDB8fHx8&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**<img src="https://images.unsplash.com/photo-1530554764233-e79e16c91d08?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MTB8fGZhc3QlMjBmb29kfGVufDB8fDB8fA%3D%3D&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**<img src="https://images.unsplash.com/photo-1555939594-58d7cb561ad1?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxleHBsb3JlLWZlZWR8M3x8fGVufDB8fHx8&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**<img src="https://images.unsplash.com/photo-1532980400857-e8d9d275d858?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MTF8fHl1bW15JTIwZm9vZHxlbnwwfHwwfHw%3D&w=1000&q=80"**

**height="250px" width="180px">**

**<br>**

**</td>**

**</tr>**

**<tr>**

**<td id="footerimage" align="center" colspan="3">**

**<hr>**

**<i><b>**

**<font color="white">**

**<h2>Copyright &copy; | Avtar Singh </h2>**

**</font>**

**</b></i>**

**<hr>**

**</td>**

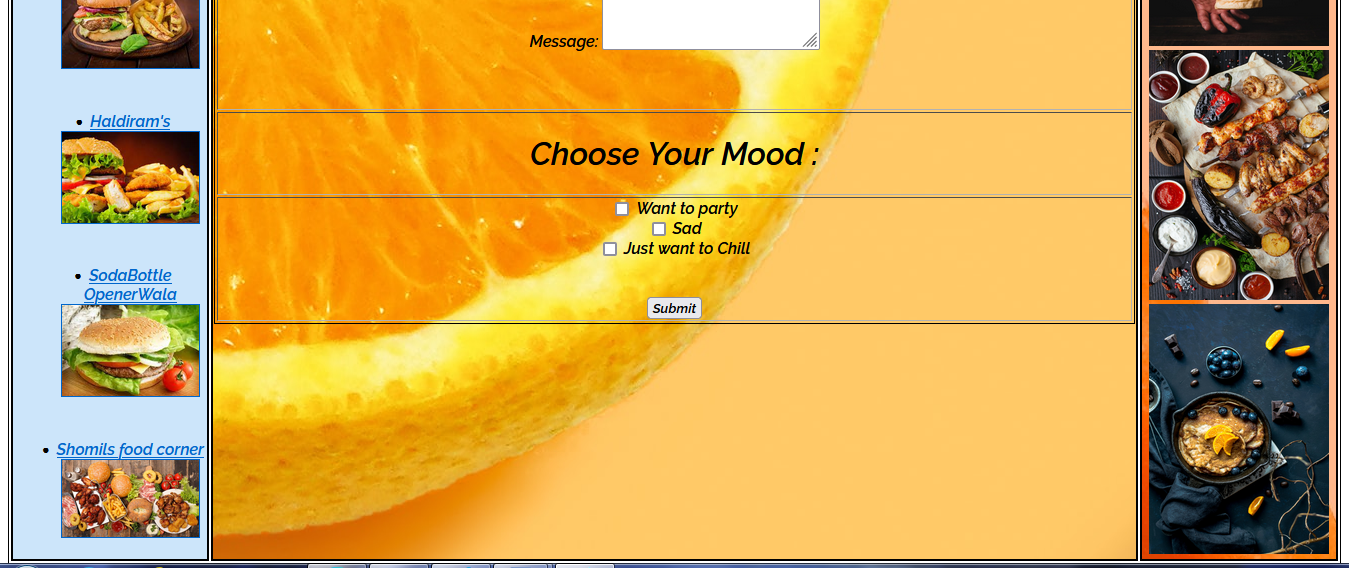
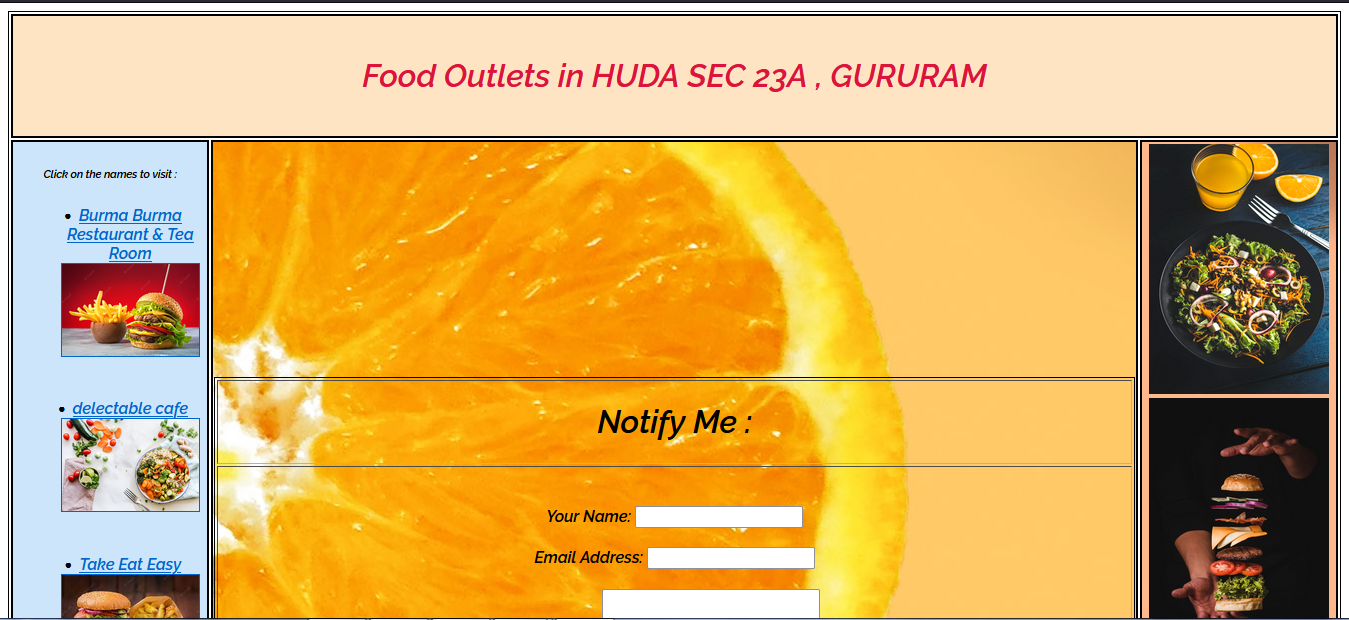
**</tr>**

**</table>**

**</body>**

**</html>**

**OUTPUT :**

****

**EXPERIMENT NO. 3**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** **24-8-22** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Create a webpage for the assigned theme, according to the template by using HTML5 ,CSS2 and CSS3**.** |
| **Outcome:** |
| **Problem Statement:**  **Definition Of Done :**  **DoD1:** The website should be divided into four sections.  **DoD2:** Each section captures the space of one complete screen in vertical direction.  **DoD3:** The first section should have the headers of the website containing top bar, search bar and main navigation bar along with relevant icons.  **DoD4:** The second, third and fourth section of the website should contain the images relevant to the theme.  **DoD5.** The website should have the following HTML 5 and CSS3 features visibility in it:   * Form * Audio/Video * Responsive Website * Advance and Dynamic Selectors * Usage of Shadows either on different divisions of webpage or images on it. * Visibility of multiple backgrounds on any of the section apart from the first section. * Slider * Flex * Transformations applied on images on hovering * Transitions * Animations |
| **Background Study:**  **CSS3 Introduction:**  Properties – An updated list of valid properties. Click the bold text to highlight the text!  Color picker – Click the blue input field and use your mouse to select another color. The slider on the right side sets the transparency of the selected color. Use the generated code snippets.  CSS background generator – Optionally set a background image from URL, set the position, repeat, attachment and color. See the live preview as you adjust the settings then take the generated code when you're satisfied with the result.  Gradient generator – Sset two and optionally a third middle transition color to get the CSS gradient code which works almost in every web browser. Change the transition direction, the position of the middle color and choose between RGB or HEX codes.  Box shadow generator – Get the CSS code for any inset or outset box shadow, customizing the right and down offset, spread, blur, color and opacity.  CSS button generator – create styles for HTML buttons with this panel. Set the size, padding, radius, color and opacity. Adjust the gradient, box shadow, font style and border in other panels.  Text shadow generator – Get the CSS code for your text shadow by setting your own color, opacity, blur, right and down shift, or pick a predefined style from the effect gallery.  Font style generator – Select font-family, size, letter and word spacing, color, font-weight, decoration, style, variant and case for your font and get the CSS code instantly.  CSS transform generator – Scale the size, rotate, shift and skew HTML elements with the transform CSS3 property.  CSS border and outline generator – Set the properties for your box border or outline to get the CSS code. Adjust the width, style, color and position of the lines surrounding your box HTML elements.  Border radius generator – Apply border radius to the corners of your elements, setting it globally or for each individual corner.  Reset CSS – A short code to be added to the top of your stylesheet. This will assure that your page will look the same in all browsers.  Media queries – Declare different views for various media types and properties.  Pixel - EM calculator – Calculate the PX or EM values, knowing the pixel size of the parent element. |
| **Question Bank:**   1. *How does CSS3 differ from CSS?*   CSS uses an old standard colour format. CSS3 provides a different gradient colour and schemes like RGBA, HSLA, HSL, and other colour gradients. CSS does not have the concept of modules. CSS3 incorporates a new feature where it can group CSS codes into convenient modules.   1. *What is a CSS Selector?*   *A CSS selector is the first part of a CSS Rule. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.*   1. *What are Child Selectors in CSS?*   *Child Selector: Child Selector is****used to match all the elements which are children of a specified element****. It gives the relation between two elements. The element > element selector selects those elements which are the children of the specific parent.*   1. *What is Webkit In CSS3 and why is it used?*   *The term 'webkit' is used in the CSS syntax for rendering content in Safari and Chrome browsers. Webkit code may need to be added in CSS to ensure it renders correctly on Chrome and Safari due to the lack of cross-compatibility.* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**PROGRAM :**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Shopping Site - Avtar Singh 20csu241</title>**

**<link rel="stylesheet" href="style.css">**

**</head>**

**<body>**

**<header style="margin-bottom: -26px;">**

**<nav class="starting\_nav">**

**<span class="title\_text">**

**<a href="#" target="\_blank" rel="noopener noreferrer">**

**<span class="red\_word">S</span><span>hopping</span><span class="red\_word">C</span><span>art</span>**

**</a>**

**</span>**

**<a href="#"> <img class="mainlogo"**

**src="https://mir-s3-cdn-cf.behance.net/project\_modules/max\_1200/7a3ec529632909.55fc107b84b8c.png"**

**alt="logo">**

**</a>**

**<div class="nav\_locator">**

**<ul style="position: absolute;top: 8px;left: 421px;">**

**<li><a href="#">Home</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">Shop Locator</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">Track Order</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">+91 12334344</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">Sign in</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">Sign Up</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">Scroll Down</a></li><Span class="long\_divider">|</Span>**

**<li><a href="#">Skip To Main Content</a></li>**

**</ul>**

**<form id="nav\_bar\_form" action="fill.html" method="get">**

**<input type="text" name="req" id="req" placeholder="How can we help you Today ?">**

**<button type="submit">Submit</button>**

**</form>**

**<button id="cart\_button" type="submit"><img**

**src="https://media.istockphoto.com/id/815698766/photo/red-shopping-cart-icon.webp?s=612x612&w=is&k=20&c=KMM43v9YfY65uZgzRxz\_eSiKT4gNdLTJPnOlzxutK6Q="**

**alt=""></button>**

**</div>**

**</nav>**

**</header>**

**<div class="nav\_categories">**

**<ul>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">All Categories</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">Home</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">About Us</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">Kitchen</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">Household</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">FAQ</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">Pages</a></li><Span**

**class="long\_divider">|</Span>**

**<li class="navbarlist"><a href="#" target="\_blank" rel="noopener noreferrer">Contact</a></li>**

**</ul>**

**</div>**

**<main>**

**<h1 id="main\_header">Our Top Products</h1>**

**<div class="container">**

**<div class="left\_sidebar">**

**<h3 class="left\_box\_heading">Search For Products : </h3>**

**<form action="#" method="get">**

**<input type="text" name="name" placeholder="Search Here" id="name">**

**<button class="search\_me" type="submit">Submit</button>**

**</form>**

**<div class="widget">**

**<h3 class="left\_box\_heading">Price Range</h3>**

**<h3 class="left\_box\_heading\_low">₹ 541 -**

**₹ 1000</h3>**

**<input type="range" id="name\_scroll">**

**</div>**

**<div class="widget">**

**<h3 class="left\_box\_heading">Food Preference</h3>**

**<ul>**

**<li>**

**<input type="checkbox" name="food" id="veg">**

**<label class="left\_box\_heading\_low" for="veg">Vegetarian</label>**

**</li>**

**<li>**

**<input type="checkbox" name="food" id="non-veg">**

**<label class="left\_box\_heading\_low" for="non-veg">Non-Vegetarian</label>**

**</li>**

**</ul>**

**</div>**

**<div class="discount\_grid">**

**<h3 class="left\_box\_heading">Discount</h3>**

**<ul class="discount\_grid">**

**<li>**

**<input type="checkbox" id="discount\_1">**

**<label class="left\_box\_heading\_low" for="discount\_1">5% or more</label>**

**</li>**

**<li>**

**<input type="checkbox" id="discount\_2">**

**<label class="left\_box\_heading\_low" for="discount\_2">10% or**

**more</label>**

**</li>**

**<li>**

**<input type="checkbox" id="discount\_3">**

**<label class="left\_box\_heading\_low" for="discount\_3">20% or**

**more</label>**

**</li>**

**<li>**

**<input type="checkbox" id="discount\_4">**

**<label class="left\_box\_heading\_low" for="discount\_4">30% or**

**more</label>**

**</li>**

**<li>**

**<input type="checkbox" id="discount\_5">**

**<label class="left\_box\_heading\_low" for="discount\_5">40% or**

**more</label>**

**</li>**

**<li>**

**</ul>**

**</div>**

**<div class="widget">**

**<h3 class="left\_box\_heading">Customer Review</h3>**

**<ul class="discount\_grid" class="reviews">**

**<li>**

**<img class="star"**

**src="https://cdn3.iconfinder.com/data/icons/stars-5/512/5\_stars-512.png"></img><span**

**class="start\_heading">5.0</span>**

**</li>**

**<li>**

**<img class="star"**

**src="https://cdn3.iconfinder.com/data/icons/stars-5/512/4\_stars-512.png"></img><span**

**class="start\_heading">4.0</span>**

**</li>**

**<li>**

**<img class="star"**

**src="https://cdn3.iconfinder.com/data/icons/stars-5/512/3\_stars-512.png"></img><span**

**class="start\_heading">3.0</span>**

**</li>**

**<li>**

**<img class="star"**

**src="https://cdn3.iconfinder.com/data/icons/stars-5/512/2\_stars-512.png"></img><span**

**class="start\_heading">2.0</span>**

**</li>**

**</ul>**

**</div>**

**</div>**

**<div class="center\_sidebar">**

**<div class="inner\_bars bar\_position\_top">**

**<h3 class="center\_sidebar\_min\_heading">Nuts <span class="new">new</span></h3>**

**<h2 class="white">--------------------**

**</h2>**

**<img class="product\_image" src="https://m.media-amazon.com/images/I/71H8N3F7scL.\_SL1488\_.jpg"**

**alt="nuts">**

**<h2 class="white">--------------------**

**</h2>**

**<div class="text\_center">**

**<h2 class="white">Price :**

**</h2>**

**<h2 class="white line"> ₹ 140/Kg**

**</h2>**

**<h2 class="white"> ₹ 120/Kg**

**</h2>**

**<h2 class="white">--------------------**

**</h2>**

**<button class="buynow\_button" type="submit">Buy Now</button>**

**</div>**

**</div>**

**<div class="inner\_bars bar\_position\_top" style="position: relative;top: -597px;left: 286px;">**

**<h3 class="center\_sidebar\_min\_heading">Almonds**

**<span class="new">new <br>**

**</span>**

**</h3>**

**<h2 class="white">--------------------**

**</h2>**

**<img class="product\_image"**

**src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS77ZwQfidX9boKBphHnsly2LKCDVwF\_P-IJSE4MQqgKtUY8iI7XR1LOXf0qMiZr6MxlDc&amp;usqp=CAU"**

**alt="almonds">**

**<h2 class="white">--------------------**

**</h2>**

**<div class="text\_center">**

**<h2 class="white">Price :**

**</h2>**

**<h2 class="white line"> ₹ 200/Kg**

**</h2>**

**<h2 class="white"> ₹ 180/Kg**

**</h2>**

**<h2 class="white">--------------------**

**</h2>**

**<button class="buynow\_button" type="submit">Buy Now</button>**

**</div>**

**</div>**

**<div class="inner\_bars bar\_position\_top" style="position: relative;top: -1193px;left: 571px;">**

**<h3 class="center\_sidebar\_min\_heading">Cashew <span class="new">new</span></h3>**

**<h2 class="white">--------------------**

**</h2>**

**<img class="product\_image"**

**src="data:image/jpeg;base64 /2Q=="**

**alt="almonds">**

**<h2 class="white">--------------------**

**</h2>**

**<div class="text\_center">**

**<h2 class="white">Price :**

**</h2>**

**<h2 class="white line"> ₹ 240/Kg**

**</h2>**

**<h2 class="white"> ₹ 220/Kg**

**</h2>**

**<h2 class="white">--------------------**

**</h2>**

**<button class="buynow\_button" type="submit">Buy Now</button>**

**</div>**

**</div>**

**<div class="main\_bar bar\_position" >**

**<h3 class="center\_sidebar\_main\_min\_heading">Mix Dry Fruits**

**<span class="white"> [ Package ] <br>**

**</span>**

**</h3>**

**<h2 class="white">-----------------------------------------------------------------**

**</h2>**

**<img class="product\_image\_big" src="https://www.mumbailive.com/images/media/images/images\_1571394441925\_Sukameva.jpg?bg=ba7a45&crop=1368%2C768%2C0%2Cnull&fit=fill&fitToScale=w%2C1368%2C768&fm=webp&h=768&height=768&w=1368&width=1368" alt="complete">**

**<div class="text\_center">**

**<h2 class="white">Price :**

**</h2>**

**<h2 class="white line"> ₹ 1200/Kg**

**</h2>**

**<h2 class="white"> ₹ 999/Kg**

**</h2>**

**<h2 class="white">--------------------**

**</h2>**

**<button class="buynow\_button\_main " type="submit">Add to Cart</button>**

**</div>**

**</div>**

**</div>**

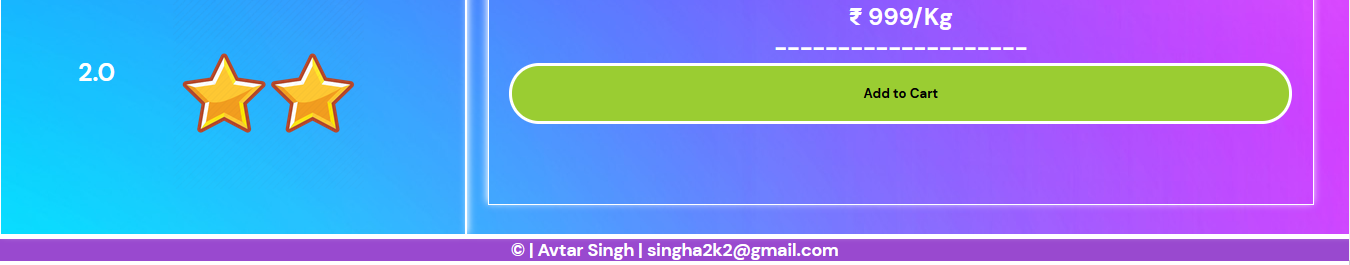
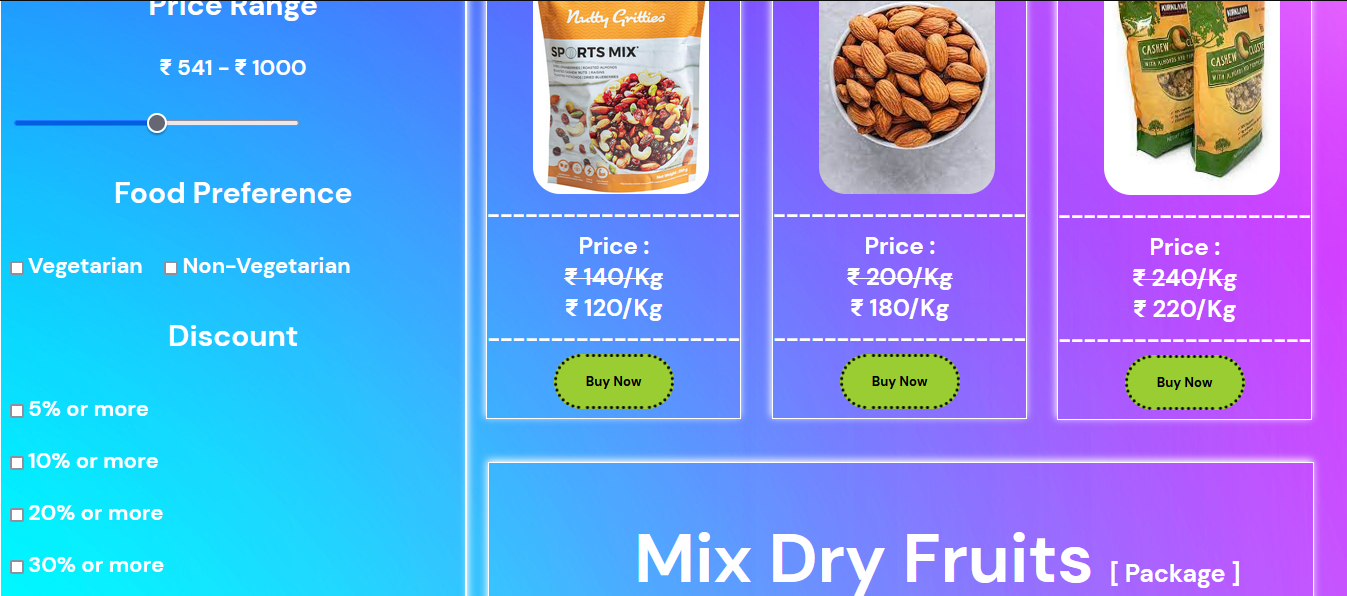
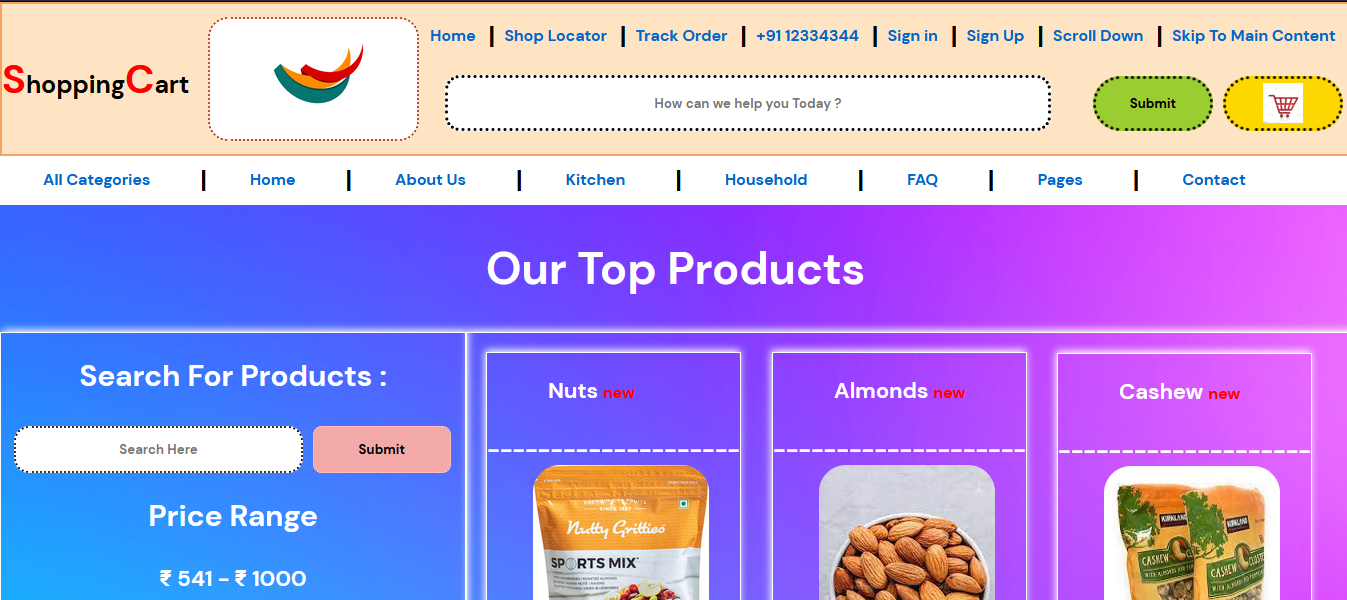
**</main>**

**<footer>**

**<h3> &copy; | Avtar Singh | singha2k2@gmail.com</h3>**

**</footer>**

**</body>**

**</html>**

**EXPERIMENT NO. 4**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** **5-9-22** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Create a front end of flipkart Website by using HTML5,CSS2 CSS3 and Bootstrap. |
| **Outcome:** |
| **Problem Statement:**  **Definition Of Done :**  **DoD1:** There must be the header part, footer part and navigation bar.  **DoD2:** Each section captures the space of one complete screen in vertical direction.  **DoD3:** The headers of the website will have search bar and main navigation bar along with relevant icons.  **DoD4:** The contents may include a carousal and box for each items with appropriate background etc. |
| **Background Study:**  **CSS Introduction:**  CSS stands for Cascading Style Sheets. It is a language designed to specify the overall appearance of webpages as well as the appearance and structure of the text and elements such as images and buttons on webpages and their layout. Styles can be specified with CSS using internal style sheet definitions which are placed right into HTML/XHTML code or in external files.  **CSS syntax**  There are three parts in the CSS syntax:  selector {property: value}   * selector - The selector is the HTML tag that a style will be applied to. For example, the <p> tag for paragraphs or the <h2> heading tag. * property - The property is an aspect of the HTML tag that will be affected by the stylesheet definition. For example, the background color of a webpage or the color of links. * value - The value that the property of a selector will have. For example, the value of the background color of a webpage can be green or the value of the color of links can be gray.   Example:  body {background-color: gray}  The property and value of a selector are separated by a colon, and surrounded by curly braces.  If a value is more than one word, put quotes around it.  Example of value being more than one word:  h2 {font-family: "Trebuchet MS"}  If you specify more than one property, each property should be separated with a semicolon.  Example of multiple properties:  body {background-color: gray; color: yellow;}  You can make style definitions more easily readable by specifying each property on a separate line.  Example:  Body{background-color: gray; color: yellow; margin-top: 0;}  **Grouping styles**  You can specify that a group of tags will have the same styles by grouping them together.  Example:  p, h1, h2, h3, h4{color: blue; font-family: courier;}  In this example, all text declared with the tags <p>, <h1>, <h2>, <h3>, and <h4> will be blue in color and Courier in font. |
| **Question Bank:**   1. *How does CSS3 differ from CSS?*   CSS uses an old standard colour format. CSS3 provides a different gradient colour and schemes like RGBA, HSLA, HSL, and other colour gradients. CSS does not have the concept of modules. CSS3 incorporates a new feature where it can group CSS codes into convenient modules.   1. *What is a CSS Selector?*   *A CSS selector is the first part of a CSS Rule. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.*   1. *What are Child Selectors in CSS?*   *Child Selector: Child Selector is****used to match all the elements which are children of a specified element****. It gives the relation between two elements. The element > element selector selects those elements which are the children of the specific parent.*   1. *What is Webkit In CSS3 and why is it used?* 2. *The term 'webkit' is used in the CSS syntax for rendering content in Safari and Chrome browsers. Webkit code may need to be added in CSS to ensure it renders correctly on Chrome and Safari due to the lack of cross-compatibility.* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**PROGRAM :**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<title>flipkart</title>**

**<meta charset="utf-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"**

**integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">**

**<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>**

**<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>**

**<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"**

**integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"**

**crossorigin="anonymous"></script>**

**<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"**

**integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"**

**crossorigin="anonymous"></script>**

**<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"**

**integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"**

**crossorigin="anonymous"></script>**

**<link rel="stylesheet" href="landing.css">**

**</head>**

**<body>**

**<nav class="navbar navbar-expand-lg navbar-light bg-primary m-0 p-0">**

**<div class="container-fluid flex-row justify-content-center">**

**<div class="d-flex flex-row m-1 p-1" style="width: 50%;">**

**<a class="navbar-brand m-0 p-0" href="#"><img**

**src="https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRdd-qzkVUo0dBoxPSfHqX6f1uOkJfLOcKdExSY-s0PiK3Ox3DY"**

**style="width: 100px; height: 45px; "></a>**

**<div class="form-group m-0 p-0 col-9">**

**<div class="m-0 p-0 input-group-text text-primary bg-white p-0">**

**<input type="text" id="searchCat" class="form-control border-0"**

**placeholder="Search for products,brands and more">**

**<span id="searchBtn" class="fa fa-search form-control-feedback bg-white px-2"></span>**

**</div>**

**</div>**

**</div>**

**<div class="justify-content-between pl-3 ml-3">**

**<button class="btn btn-default text-primary bg-white font-weight-bold"><a href="http://google.com" target="\_blank" rel="noopener noreferrer"> Login</a></button>**

**<div class="btn-group">**

**<button type="button" class="btn text-primary bg-white dropdown-toggle font-weight-bold" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">**

**Subscribe**

**</button>**

**<div class="dropdown-menu">**

**<a class="dropdown-item" href="#">Github</a>**

**<a class="dropdown-item" href="#">Google+</a>**

**<a class="dropdown-item" href="#">Instagram</a>**

**</div>**

**</div>**

**<button class="btn text-white font-weight-bold" style="font-size: 19px;"> <span id="searchBtn"**

**class="fa fa-shopping-cart px-1"></span>Cart</button>**

**</div>**

**</div>**

**</nav>**

**<nav class="navbar navbar-expand-lg navbar-light border m-0 p-0">**

**<div class="container-fluid text-dark justify-content-center m-0 p-0">**

**<div class="dropdown"><button id="Electronic" class="btn dropbtn dropdown-toggle font-weight-bold m-1"**

**type="button" data-toggle="dropdown" style="font-size : 14px;">Electronics<span**

**class="caret"></span></button>**

**<div class="dropdown-content">**

**<a href="#">Mobile</a>**

**<a href="#">Headphone</a>**

**<a href="#">trimmer</a>**

**</div>**

**</div>**

**<div class="dropdown"><button id="Tv" class="btn dropbtn dropdown-toggle font-weight-bold m-1" type="button"**

**data-toggle="dropdown" style="font-size : 14px;">TV&Appliances<span class="caret"></span></button>**

**<div class="dropdown-content">**

**<a href="#">samsung</a>**

**<a href="#">Lg</a>**

**<a href="#">OLED</a>**

**</div>**

**</div>**

**<div class="dropdown"><button id="Men" class="btn dropbtn dropdown-toggle font-weight-bold m-1"**

**type="button" data-toggle="dropdown" style="font-size : 14px;">Men<span**

**class="caret"></span></button>**

**<div class="dropdown-content">**

**<a href="#">casuals</a>**

**<a href="#">formals</a>**

**<a href="#">party</a>**

**</div>**

**</div>**

**<div class="dropdown"><button id="Women" class="btn dropbtn dropdown-toggle font-weight-bold m-1"**

**type="button" data-toggle="dropdown" style="font-size : 14px;">Women<span**

**class="caret"></span></button>**

**<div class="dropdown-content">**

**<a href="#">saree</a>**

**<a href="#">party wear</a>**

**<a href="#">jackets</a>**

**</div>**

**</div>**

**<div class="dropdown"><button id="Baby" class="btn dropbtn dropdown-toggle font-weight-bold m-1"**

**type="button" data-toggle="dropdown" style="font-size : 14px;">Baby&Kids<span**

**class="caret"></span></button>**

**<div class="dropdown-content">**

**<a href="#">cosmetics</a>**

**<a href="#">creams</a>**

**<a href="#">toys</a>**

**</div>**

**</div>**

**<div class="dropdown"><button id="Home" class="btn dropbtn dropdown-toggle font-weight-bold m-1"**

**type="button" data-toggle="dropdown" style="font-size : 14px;">Home&Furniture<span**

**class="caret"></span>**

**<div class="dropdown-content">**

**<a href="#">kitchen</a>**

**<a href="#">home lights</a>**

**<a href="#">home decor</a>**

**</div>**

**</button>**

**</div>**

**<div class="dropdown"><button id="Sports" class="btn dropdown-toggle font-weight-bold m-1" type="button"**

**data-toggle="dropdown" style="font-size : 14px;">Sports,books&more<span**

**class="caret"></span></button>**

**<div class="dropdown-content">**

**<a href="#">cricket</a>**

**<a href="#">badminton</a>**

**<a href="#">more</a>**

**</div>**

**</div>**

**<div><button id="add" class="btn font-weight-bold m-1">Add+</button></div>**

**<div><button id="showData" class="btn font-weight-bold m-1">Show Products</button></div>**

**<div><button id="sort" class="btn font-weight-bold m-1">Price Up</button></div>**

**<div><button id="reverse" class="btn font-weight-bold m-1">Price Down</button></div>**

**</div>**

**</nav>**

**<div class="container-fluid bg-primary text-light mt-1 m-0 ">**

**<div class="row">**

**<form id="form" class="m-2">**

**<h1 class="p-2">ADD PRODUCT DETAILS</h1>**

**<div class="row">**

**<div class=" container justify-content-center">**

**<div class="form-group">**

**<label class="p-2">Add image Url</label>**

**<input type="text" class="form-control p-2" id="Image" placeholder="Enter Image Url"**

**required>**

**</div>**

**<div class="form-group">**

**<label class="p-2">Add product name</label>**

**<input type="text" class="form-control p-2" id="Product" placeholder="Enter Product Name"**

**required>**

**</div>**

**<div class="form-group">**

**<label class="p-2">Add Price</label>**

**<input type="number" class="form-control p-2" id="Price" placeholder="Enter Price" required>**

**</div>**

**<div>**

**<label class="p-2">Category</label>**

**<select class="p-2" id="Category">**

**<option value="electronic">Electronics</option>**

**<option value="tv">TV&Appliances</option>**

**<option value="men">Men</option>**

**<option value="women">Women</option>**

**<option value="baby">Baby&Kids</option>**

**<option value="home">Home&Furniture</option>**

**<option value="sports">Sports,books&more</option>**

**</select>**

**</div>**

**</div>**

**</div>**

**<button class="btn btn-success" id="submit">Submit</button>**

**</form>**

**</div>**

**</div>**

**<div class="container-fluid m-0">**

**<div id="carouselExampleIndicators" class="carousel mb-0 mx-4 slide data-ride=" carousel">**

**<ol class="carousel-indicators">**

**<li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>**

**<li data-target="#carouselExampleIndicators" data-slide-to="1"></li>**

**<li data-target="#carouselExampleIndicators" data-slide-to="2"></li>**

**</ol>**

**<div class="carousel-inner">**

**<div class="carousel-item active">**

**<img class="d-block w-100"**

**src="https://rukminim1.flixcart.com/flap/3376/560/image/de3e826deb18f627.jpg?q=50 "**

**alt="First slide">**

**</div>**

**<div class="carousel-item">**

**<img class="d-block w-100"**

**src="https://rukminim1.flixcart.com/flap/3376/560/image/a202fb36f2ed03de.jpg?q=50"**

**alt="Second slide">**

**</div>**

**<div class="carousel-item">**

**<img class="d-block w-100"**

**src="https://rukminim1.flixcart.com/flap/1688/280/image/55b9b02b1222cfc8.jpg?q=50"**

**alt="Third slide">**

**</div>**

**</div>**

**<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">**

**<span class="carousel-control-prev-icon" aria-hidden="true"></span>**

**<span class="sr-only">Previous</span>**

**</a>**

**<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">**

**<span class="carousel-control-next-icon" aria-hidden="true"></span>**

**<span class="sr-only">Next</span>**

**</a>**

**</div>**

**</div>**

**<div id="showDiv" class="container-fluid m-0 mt-1 p-0">**

**<div id="cardsDiv" class="container-fluid row mt-1 p-0"></div>**

**</div>**

**<div class="container-fluid">**

**<div class="mainflex mx-4">**

**<div class="flexbox border">**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/flap/150/150/image/973a70fff3bb0962.jpg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Bike Accessories</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">From ₹99+Extra10%Off</p>**

**<p class="card-text text-muted ">Top Selling</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/flap/150/150/image/5a4a08e3aa382826.jpg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Casual Shirts</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Min 65%Off+Extra 5%</p>**

**<p class="card-text text-muted ">BestSelling Collection</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/juwzf680/headphone-refurbished/e/g/m/z-rockerz-255-boat-original-imaf8pudy3ebmssn.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Headphone&Speaker</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto60%Off+Extra10%Off</p>**

**<p class="card-text text-muted ">Jbl,sony&more </p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/flap/150/150/image/5a23ef3e7d789a91.jpg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Designer Back Covers</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Just ₹179</p>**

**<p class="card-text text-muted ">For All Top Models</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/k2urhjk0/shoe/2/9/s/32-6-lancer-maroon-original-imafm425fzr9nufs.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Sparx,Lancer</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">From ₹199</p>**

**<p class="card-text text-muted ">Men;s Footwear</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/chopper/p/u/k/new-handy-chopper-pigeon-original-imaemn3bdyjfrthy.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Kitchen Tools</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">From ₹99</p>**

**<p class="card-text text-muted ">Pigeon&more</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/flap/150/150/image/b80fa78f1eaa3141.jpg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Heels</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 70%Off</p>**

**<p class="card-text text-muted ">Top Selling</p>**

**</div>**

**</div>**

**</div>**

**<div class="posterDiv">**

**<img class="poster" src="https://rukminim1.flixcart.com/flap/464/708/image/7326117459d6663c.jpg?q=70">**

**</div>**

**</div>**

**</div>**

**<div class="container-fluid">**

**<div class="cardDiv row mx-4 border">**

**<img class="col-sm-12 col-lg-4 mt-1"**

**src="https://rukminim1.flixcart.com/flap/480/480/image/af2926cf02a3c3c9.jpg?q=50">**

**<img class="col-sm-12 col-lg-4 mt-1"**

**src="https://rukminim1.flixcart.com/flap/480/480/image/9581455d64cad13b.jpg?q=50">**

**<img class="col-sm-12 col-lg-4 mt-1 "**

**src="https://rukminim1.flixcart.com/flap/480/480/image/92b2b7267e3c9cf4.jpg?q=50">**

**</div>**

**</div>**

**<div class="container-fluid">**

**<div class="mx-4 border">**

**<div class="row justify-content-between border m-1">**

**<div>**

**<h5 class="font-weight-bold">Top Rated Appliances For You</h5>**

**<p class="text-muted">Shop Now</p>**

**</div>**

**<div>**

**<button type="button" class="btn font-weight-bold btn-primary btn-block">VIEW ALL</button>**

**</div>**

**</div>**

**<div class="flexbox2">**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/jv6zi4w0/fan/v/r/n/renesa-smart-1200-ceiling-fan-gorilla-original-imafepmhzz3ekvak.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Energy Efficient Fans</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">From ₹1,099</p>**

**<p class="card-text text-muted ">Orient,Usha&more</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/jialea80/television/v/m/w/samsung-65q7fn-original-imaf64f2fzug5wbp.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Samsung QLED 8k TVs</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Up to 60%Off</p>**

**<p class="card-text text-muted ">Brighter Colors</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/k20r8nk0/roti-khakra-maker/r/x/m/puri-press-roti-maker-khakhra-maker-chapati-maker-machine-medigo-original-imafhgrhf7bmns64.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Roti Makers</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 75% Off</p>**

**<p class="card-text text-muted ">Prestige,Aiwa&More</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/jingcy80/electric-kettle/w/y/v/philips-hd9329-06-original-imaf6e6vz6seyhzd.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Electric Kettle</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 60% off</p>**

**<p class="card-text text-muted ">Prestige,Singer&more</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/jjkbhjk0/sandwich-maker/5/q/g/nova-2-slice-sandwich-maker-original-imaf74a8zrqzfgug.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Sandwich Makers</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 80% Off</p>**

**<p class="card-text text-muted ">Prestige,Nova&more</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/jd4u5jk0/mixer-grinder-juicer/h/k/8/prestige-41115-original-imaffff4thqwbq9t.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Juicers</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 50% Off</p>**

**<p class="card-text text-muted ">Phillips,Prestige&more</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/jqwny4w0/room-heater/4/k/v/fksbrhqrg-flipkart-smartbuy-original-imafctdfzb5dhh5y.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Room Heaters</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 75% Off</p>**

**<p class="card-text text-muted ">Usha,Bajaj&more</p>**

**</div>**

**</div>**

**<div class="card text-center">**

**<img src="https://rukminim1.flixcart.com/image/150/150/k0sgl8w0/landline-phone/r/j/j/c51-beetel-original-imafkg2wrahtjgfz.jpeg?q=70"**

**height="150px" class="card-img-top">**

**<div class="card-body text-dark text-center">**

**<h6 class="card-title font-weight-bold">Landline</h6>**

**<p class="card-subtitle mb-2 text-success" style="font-size: 16px;">Upto 30%Off</p>**

**<p class="card-text text-muted ">Best Seller</p>**

**</div>**

**</div>**

**</div>**

**</div>**

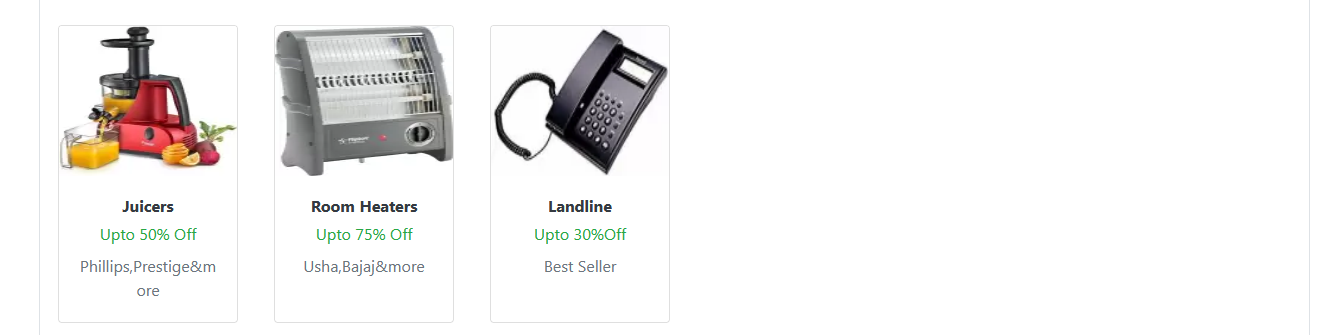
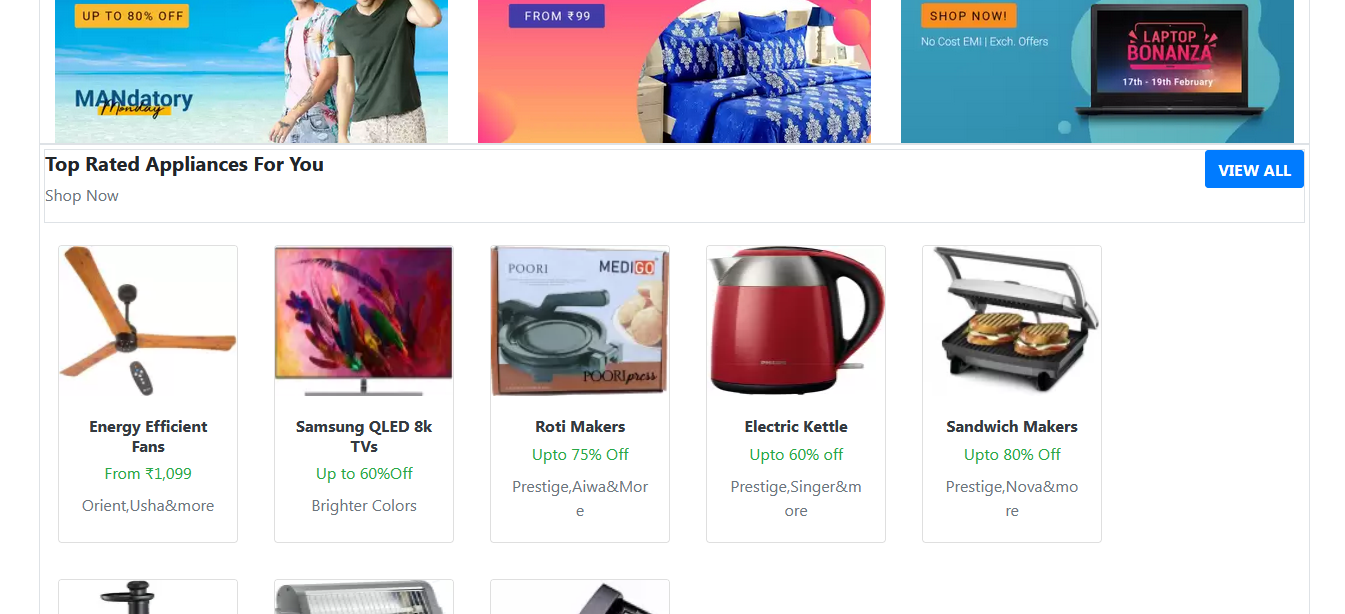
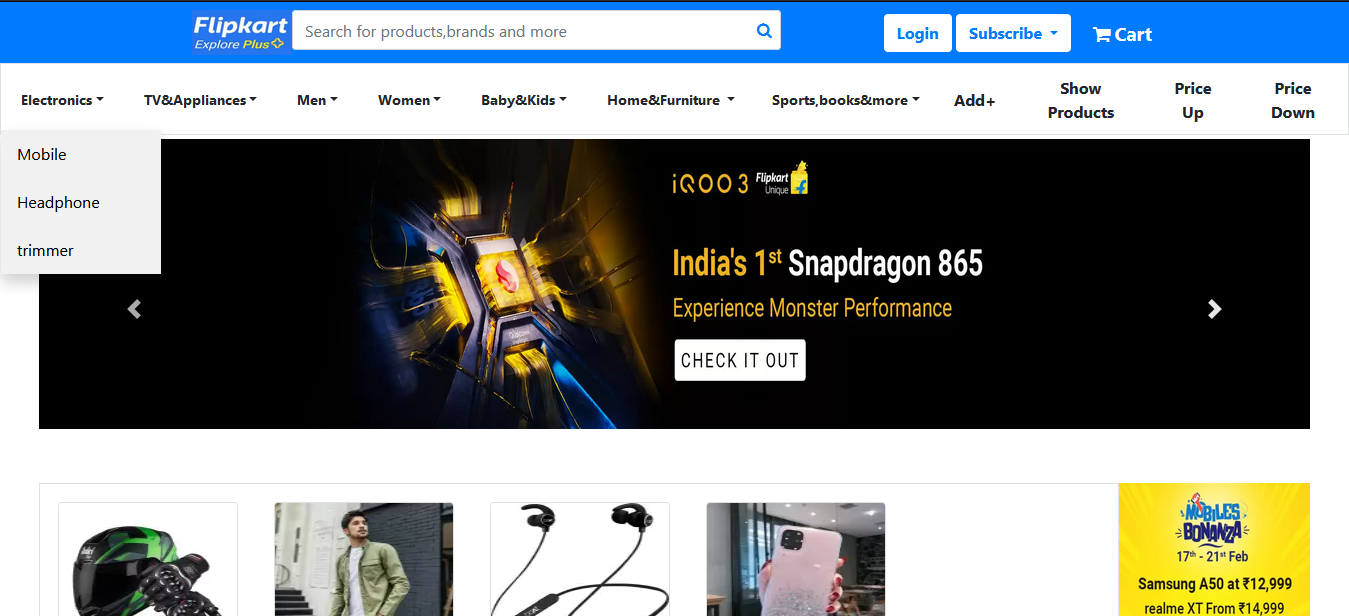
**</div>**

**<script src="landing.js"></script>**

**</body>**

**</html>**

**OUTPUT :**

****

**EXPERIMENT NO. 5**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** **12-9-22** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Basic programs on JavaScript |
| **Outcome:**  The output of the commands and functions written and executed on console of web browser or console of VS code |
| **Problem Statement:**  **Definition Of Done :**  **DoD1:** There must be the header part, footer part and navigation bar.  **DoD2:** Each section captures the space of one complete screen in vertical direction.  **DoD3:** The headers of the website will have search bar and main navigation bar along with relevant icons.  **DoD4:** The contents may include a carousal and box for each items with appropriate background etc. |
| **Background Study:**  **Introduction**  JavaScript is a language used in HTML (Web) pages, and understood by most Web browsers, to add interactivity to Web pages.  JavaScript has been largely standardized by ECMA (the European Computer Manufacturers Association) as a language formally named ECMAScript v3, usually known as JavaScript 1.5 (or ActionScript, JScript, QtScript, JScript .NET, DMDScript, or InScript). All modern browsers implement EMCAScript v3 reasonably well, with possibly some omissions and errors, and most browers add features of their own. In this paper I attempt to discuss primarily platform-independent features, but make no attempt to keep track of what doesn't work in various browsers.  Because JavaScript is used almost exclusively in Web pages, it cannot access files and has very limited networking capabilities. The only graphics capability it has is the ability to create HTML programmatically.  **Data types**  JavaScript has three primitive types: number, string, and boolean, and two special values, null and undefined. Everything else is an object. An array is a type of object.  A number may be written as an integer or as a real number; however written, numbers are stored as floating-point. Special predefined numbers include Infinity, NaN ("not a number"), Number.NaN (same as NaN), Number.MIN\_VALUE, Number.MAX\_VALUE, Number.NEGATIVE\_INFINITY, and Number.POSITIVE\_INFINITY (same as Infinity). A hexadecimal literal begins with 0x or 0X. A nonzero number should never be written with a leading zero (e.g. 0123), because some implementations will treat this as an octal number and others won't.  A string is a sequence of zero or more characters enclosed in either single quotes ('hello') or double quotes ("hello"). There is no "character" data type.  A boolean has one of two values: true or false. When other values are converted into boolean (typically, for use as a test), the values 0, "0", the empty string, undefined, null, and NaN are converted to false, and other values are converted to true.  An object is a collection of named values, called its properties. If the name of a property is known to the programmer, it can be referenced by the syntax object.property\_name, for example, image.width. If the name can be computed, it can be referenced by the syntax object[expression], for example, image["width"].  The special value null is an object with no properties. A similar value, undefined, means that the variable has never been assigned a value or that the requested property does not exist.  Functions, arrays, regular expressions, errors, and dates are all special types of object. In addition, there are "wrapper" types Number, String, and Boolean, to allow object-like operations on primitive values; but JavaScript automatically converts between primitives and wrappers as needed, so the programmer almost never sees these wrapper types.  **Identifiers**  Identifiers consist of letters (as defined by Unicode), digits, underscores, and/or dollar signs; the first character may not be a digit. Case is significant in JavaScript but not in HTML; since JavaScript names are frequently used in HTML, this can lead to some strange errors if you are not careful about case. By convention, the names of variables, properties, functions, and methods should begin with a lowercase letter; names of constructors should begin with a capital letter. Names may be any length.  **Comments**  // introduces a comment that extends to the end of the line. Multi-line comments start with /\* and end with \*/. Inside a comment, // and /\* have no special meaning (so you cannot "nest" comments). Inside a quoted string or regular expression, // and /\* do *not* start a comment.  **Operators**  **Arithmetic operators**  + (Addition), - (Subtraction), \* (Multiplication), / (Division), % (Modulus), ++ (Increment), -- (Decrement)  **Comparison Operators**  = = (Equal), != (Not Equal), > (Greater than), < (Less than), >= (Greater than or Equal to), <= (Less than or Equal to)  **Logical Operators**  && (Logical AND), || (Logical OR), ! (Logical NOT)  **Assignment Operators**  = (Simple Assignment ), += (Add and Assignment), −= (Subtract and Assignment), \*= (Multiply and Assignment), /= (Divide and Assignment), %= (Modules and Assignment)  **Miscellaneous Operator**  Conditional Operator (? :)  **Expressions**  An expression is any JavaScript code that results in a value. Expressions may be as simple as a single constant or single variable.  Operators with higher precedence (indicated by larger numbers) are performed before those with lower precedence. For example, in the expression 2\*3+4\*5, the multiplications are done before the addition, because multiplication has higher precedence than addition.  When operators have equal precedence, the associativity ("left to right," or "right to left") determines which operations are done first. For example, substraction has left to right associativity, so 10-5-3 means (10-5)-3 rather than 10-(5-3). Assignment has right to left associativity, so a=b=c+5 means a=(b=c+5) rather than (a=b)=c+5. Most common operators, other than the assignment operators, associate left to right.  Parentheses can be used to override the above rules and specify an explicit order of evaluation. Parentheses are also used to show the order of evaluation when it might not be obvious.  **Statements**  Statements are terminated by a semicolon or by the end of the line. Semicolons are required only to separate two or more statements on the same line. If a statement needs to extend over two lines, take care to separate it in a place where the first line cannot be understood as a complete statement.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Good:** | w = a + b +     c + d |  | **Bad:** | w = a + b   + c + d |   **Functions**  A function is a block of code that can be "invoked" (called). Functions in HTML should normally be defined in the<head>element, to ensure that they will be defined before they are needed. Functions may be recursive. Functions may be nested within other functions.  Example definition:  function average(x, y) { // x and y are formal parameters  var sum; // this declares sum as a local variable  g = x; // this declares g as a global variable  sum = g + y;  return sum / 2; }  Example call:  avg = average(5, 10);  **Object prototypes**  Each JavaScript object belongs to a class, and each class has a prototype. The properties of an object belong to that object alone. If you want a property or function shared across all instances of a class, you need to attach them to the class prototype. Prototype properties are similar to static values and methods in Java.  For example, to ensure that every Car has four wheels, you might write:      Car.prototype.wheelCount = 4;  You can attach a method to a single object like this:      myCar.age = function() { return this.year - 2005; }  **Arrays**  An array is an object with some special features.  You can create an array (say, a) as follows:  var a = new Array();     // an array with no elements (initially)  **Array methods**  join()  Converts all the elements of the array to strings and concatenates them, with a comma separator.  join(string)  Converts all the elements of the array to strings and concatenates them, using the given string as a separator.  reverse()  Reverses the elements of the array in place, that is, no new array is created.  sort()  Sorts the array in place, in alphabetical order.  sort(function)  Sorts the array in place. The function takes two arguments, and returns a negative number if the first element is judged "less than" the second, zero if the two elements are "equal," and a positive number if the first element should be considered "larger."  concat(arg1, ..., argN) or concat([arg1, ..., argN])  Creates and returns a new array with all the elements of the old array, and also the elements arg1, ..., argN.  slice(start, end)  Returns a new array consisting of elements start through end - 1 of the original array.  splice(start)  Removes all elements from index location start to the end of the array, and returns them in a new array.    splice(start, count)  Removes count elements from the array, starting at index location start, and returns them in a new array.  splice(start, count, value1, ..., valueN)  Removes count elements from the array, starting at index location start, and replaces them with new elements value1, ..., valueN. The removed elements are returned in a new array.    push(value1, ..., valueN)  Adds value1, ..., valueN to the end of the array; the return value is the new length of the array.  pop()  Removes and returns the last value in the array, and decrements the length of the array.  unshift(value1, ..., valueN)  Adds value1, ..., valueN to the beginning of the array; the return value is the new length of the array.  shift()  Removes and returns the first value in the array. Remaining elements are all shifted down one place.  toString()  Creates and returns a comma-separated list of the string representations of each element in the array.  **Strings**  A string is a primitive value consisting of a sequence of zero or more characters. A String (capitalized) is a wrapper object for a string. The methods in this section are actually String methods, but JavaScript converts between the primitive and the wrapper as needed.  Strings have a length property, which is the number of characters in the string.  Strings can be concatenated (producing a new string) with the + operator.  String methods  Here are some of the more important methods on strings.  charAt(n)  Returns the nth character of a string.  charCodeAt(n)  Returns the numeric Unicode value of the nth character of a string.  concat(string1, ..., stringN)  Concatenates the string arguments to the given string.  fromCharCode(c1, ..., cN)  Creates a string from the given numeric Unicode values.  indexOf(substring)  Returns the position of the first character of substring in the given string, or -1 if not found.  indexOf(substring, start)  Returns the position of the first character of substring in the given string that begins at or after position start, or -1 if not found.  lastIndexOf(substring) lastIndexOf(substring, start)  Like indexOf, but searching starts from the end of the given string.  localeCompare(string)  Returns a negative, zero, or positive number according to whether string is less than, equal to, or greater than the given string, according to locale-specific ordering.  match(regexp)Returns an array containing the results of the match, or null if no match is found. On a successful match:  If g (global) is set, the array contains the matched substrings.  If g is not set, array location 0 contains the matched text, and remaining locations contain text matched by any parenthesized subexpressions. The array index property gives the position of the first character of the match.  replace(regexp, replacement)Returns an new string that has the matched substring replaced with the replacement.  search(regexp)Returns the position of the first matched substring in the given string, or -1 if not found. |
|  |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

Write a JavaScript program to display the current day and time in the following format.

**Program :**

**<!doctype html>**

**<html lang="en">**

**<head>**

**<meta charset="utf-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" ">**

**<title>Buyme.com - The Hybrid Dukan</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">**

**</head>**

**<body>**

**<nav class="navbar navbar-expand-lg bg-light">**

**<div class="container-fluid">**

**<a class="navbar-brand" href="index.html"><img width="250" src="logo.png" alt="#" /></a>**

**<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">**

**<span class="navbar-toggler-icon"></span>**

**</button>**

**<div class="collapse navbar-collapse" id="navbarSupportedContent">**

**<ul class="navbar-nav me-auto mb-2 mb-lg-0" style="margin-left: 20px;">**

**<li class="nav-item" style="margin-left: 48px;margin-right: 48px;">**

**<a class="nav-link active" aria-current="page" href="#">Home</a>**

**</li>**

**<li class="nav-item" style="margin-left: 48px;margin-right: 48px;">**

**<a class="nav-link active" aria-current="page" href="#">Contact Us</a>**

**</li>**

**<li class="nav-item dropdown" style="margin-left: 48px;margin-right: 48px;">**

**<a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">**

**Social Media**

**</a>**

**<ul class="dropdown-menu" style="margin-left: 48px;margin-right: 48px;">**

**<li ><a class="dropdown-item" href="#">Instagram</a></li>**

**<li><a class="dropdown-item" href="#">Facebook</a></li>**

**<li><hr class="dropdown-divider"></li>**

**<li><a class="dropdown-item" href="#">GitHub</a></li>**

**</ul>**

**</li>**

**<li class="nav-item" style="margin-left: 48px;margin-right: 48px;">**

**<a class="nav-link disabled">Win Iphone</a>**

**</li>**

**</ul>**

**<form class="d-flex" role="search">**

**<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">**

**<button class="btn btn-outline-success" type="submit">Search</button>**

**</form>**

**</div>**

**</div>**

**</nav>**

**<!-- Carosuel -->**

**<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="true">**

**<div class="carousel-indicators">**

**<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>**

**<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>**

**<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3"></button>**

**</div>**

**<div id="carouselExampleFade" class="carousel slide carousel-fade" data-bs-ride="carousel">**

**<div class="carousel-inner">**

**<div class="carousel-item active">**

**<img src="https://images.unsplash.com/photo-1593412578702-3332c86f3db3?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=725&q=80" class="d-block w-100" alt="...">**

**</div>**

**<div class="carousel-item">**

**<img src="https://images.unsplash.com/photo-1487744480471-9ca1bca6fb7d?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=891&q=80" class="d-block w-100" alt="...">**

**</div>**

**<div class="carousel-item">**

**<img src="https://images.unsplash.com/photo-1534452203293-494d7ddbf7e0?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=872&q=80" class="d-block w-100" alt="...">**

**</div>**

**</div>**

**<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleFade" data-bs-slide="prev">**

**<span class="carousel-control-prev-icon" aria-hidden="true"></span>**

**<span class="visually-hidden">Previous</span>**

**</button>**

**<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleFade" data-bs-slide="next">**

**<span class="carousel-control-next-icon" aria-hidden="true"></span>**

**<span class="visually-hidden">Next</span>**

**</button>**

**<h2 style="font-size: 2.6rem;letter-spacing: 0.05em;line-height: 1;font-weight: 600;font-style: normal;align-items: center;text-align: center;margin-top: 27px;margin-bottom: 27px;">Why <span style="color: red;"> Us</span></h2>**

**<div class="row row-cols-1 row-cols-md-3 g-4">**

**<div class="col">**

**<div class="card h-100" style="background-color: #002c3e;color: white;">**

**<img src="badge.png" class="card-img-top" style="width: 69%;margin-left: 57px;" alt="...">**

**<div class="card-body">**

**<h5 class="card-title">Best Quality</h5>**

**<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</p>**

**</div>**

**<div class="card-footer">**

**<small class="text-muted">Last updated 3 mins ago</small>**

**</div>**

**</div>**

**</div>**

**<div class="col">**

**<div class="card h-100" style="background-color: #002c3e;color: white;">**

**<img src="Dilvery.png" class="card-img-top" style="width: 69%;margin-left: 57px;" alt="...">**

**<div class="card-body">**

**<h5 class="card-title">Fast Dilvery</h5>**

**<p class="card-text">This card has supporting text below as a natural lead-in to additional content.</p>**

**</div>**

**<div class="card-footer">**

**<small class="text-muted">Last updated 3 mins ago</small>**

**</div>**

**</div>**

**</div>**

**<div class="col">**

**<div class="card h-100" style="background-color: #002c3e;color: white;">**

**<img src="free.png" class="card-img-top" style="width: 69%;margin-left: 57px;" alt="...">**

**<div class="card-body">**

**<h5 class="card-title">Free Shipping</h5>**

**<p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This card has even longer content than the first to show that equal height action.</p>**

**</div>**

**<div class="card-footer">**

**<small class="text-muted">Last updated 3 mins ago</small>**

**</div>**

**</div>**

**</div>**

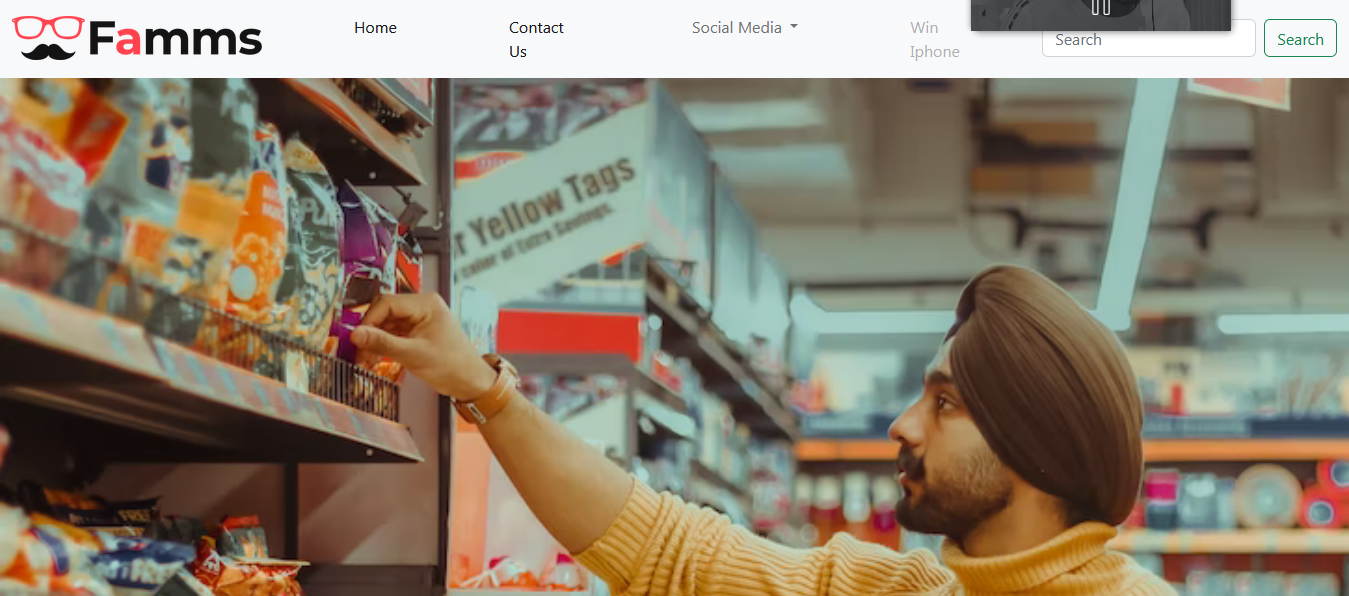
**</div>**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vIsF0jrRAoQmDKKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"></script>**

**</body>**

**</html>**

**OUTPUT :**

****

**Program (index.html) :**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Document</title>**

**</head>**

**<body>**

**<script src="script.js"></script>**

**</body>**

**</html>**

**PROG1:**

**const matches = (obj, source) =>**

**Object.keys(source).every(key => obj.hasOwnProperty(key) && obj[key] === source[key]);**

**document.writeln(" { age: 25, hair: 'long', beard: true }, { hair: 'long', beard: true } : Result :: "+matches({ age: 25, hair: 'long', beard: true }, { hair: 'long', beard: true })+"\n"); // true**

**document.writeln("{ hair: 'long', beard: true }, { age: 25, hair: 'long', beard: true } : Result :: "+matches({ hair: 'long', beard: true }, { age: 25, hair: 'long', beard: true })+"\n"); // false**

**document.writeln("{ hair: 'long', beard: true }, { age: 26, hair: 'long', beard: true } : Result :: "+matches({ hair: 'long', beard: true }, { age: 26, hair: 'long', beard: true })+"\n"); // false**

**OUTPUT : **

**PROG2:**

**var today = new Date();**

**// Creating object refernce of Date**

**var day = today.getDay();**

**// Getting Current Date**

**console.log("Index of Today's Day is : "+day);**

**var daylist = ["Sunday","Monday","Tuesday","Wednesday ","Thursday","Friday","Saturday"];**

**// Providing the daylist**

**console.log("Today's Day is : " + daylist[day] + ".");**

**// Getting Time from Refernced Var**

**var hour = today.getHours();**

**var minute = today.getMinutes(); var second = today.getSeconds();**

**let prepand = (hour >= 12)? " PM ":" AM ";**

**hour = (hour >= 12)? hour - 12: hour;**

**if (hour===0 && prepand===' PM ')**

**{**

**if (minute===0 && second===0)**

**{**

**hour=12;**

**prepand=' Noon';**

**}**

**else**

**{**

**hour=12;**

**prepand=' PM';**

**}**

**}**

**if (hour===0 && prepand===' AM ')**

**{**

**if (minute===0 && second===0)**

**{**

**hour=12;**

**prepand=' Midnight';**

**}**

**else**

**{**

**hour=12;**

**prepand=' AM';**

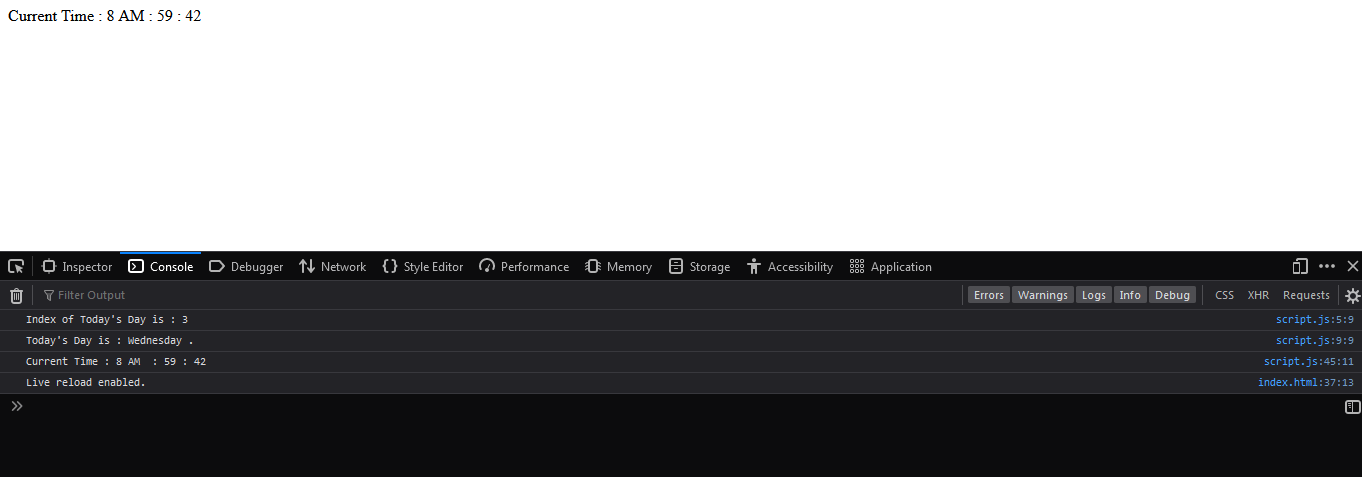
**}**

**}**

**console.log(`Current Time : ${hour}${prepand} : ${minute} : ${second}`);**

**document.write(`Current Time : ${hour}${prepand} : ${minute} : ${second}`);**

**OUTPUT:**

****

**PROG3:**

**var num = prompt("Enter Number You Want To Reverse : ")**

**function reverse\_a\_number(n)**

**{**

**n = n + "";**

**return n.split("").reverse().join("");**

**}**

**var rev = reverse\_a\_number(num);**

**window.onload = function() {**

**document.getElementById("heading").innerHTML=rev;**

**}**

**document.write("Entered Number is : "+num);**

**OUTPUT :**

****

**PROG 4:**

**// program to display the sum of natural numbers**

**// take input from the user**

**const number = parseInt(prompt('Enter a positive integer: '));**

**let sum = 0;**

**// looping from i = 1 to number**

**// in each iteration, i is increased by 1**

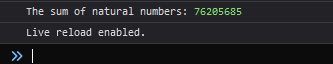
**for (let i = 1; i <= number; i++) {**

**sum += i;**

**}**

**console.log('The sum of natural numbers:', sum);**

**OUTPUT :**

****

**PROG 5 :**

**var mail = prompt("Enter Your Mail : ");**

**ValidateEmail(mail);**

**function ValidateEmail(mail)**

**{**

**if (/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/)**

**{**

**alert("You have entered Valid email address!")**

**return (true)**

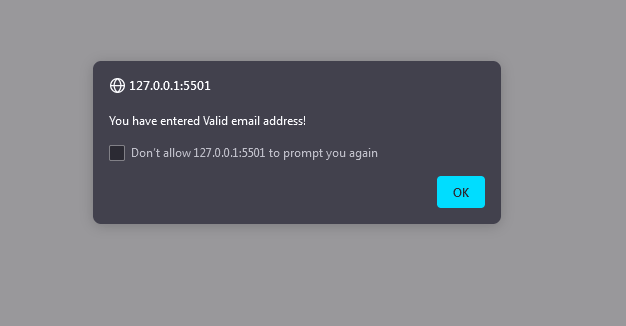
**}**

**alert("You have entered an invalid email address!")**

**return (false)**

**}**

**OUTPUT :**

****

**PROG 6 :**

**function phonenumber(inputtxt)**

**{**

**var phoneno = /^\+?([0-9]{2})\)?[-. ]?([0-9]{4})[-. ]?([0-9]{4})$/;**

**if((inputtxt.value.match(phoneno)))**

**{**

**alert("Valid Mob Number");**

**return true;**

**}**

**else**

**{**

**alert("Mobile Number is not valid");**

**return false;**

**}**

**}**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="utf-8">**

**<title>JavaScript form validation - checking non-empty</title>**

**<link rel='stylesheet' href='style.css' type='text/css' />**

**</head><body onload='document.form1.text1.focus()'>**

**<div class="mail">**

**<h2>Input an Phone No.[xxxxxxxxxx] and Submit</h2>**

**<form name="form1" action="#">**

**<ul>**

**<li><input type='text' name='text1'/></li>**

**<li>&nbsp;</li>**

**<li class="submit"><input type="submit" name="submit" value="Submit" onclick="phonenumber(document.form1.text1)"/></li>**

**<li>&nbsp;</li>**

**</ul>**

**</form>**

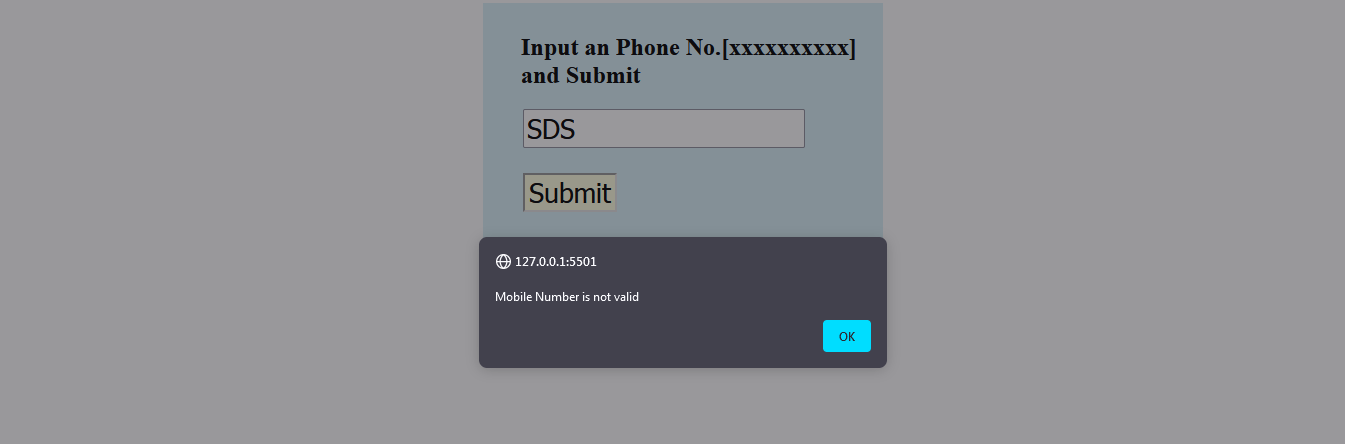
**</div>**

**<script src="script.js"></script>**

**</body>**

**</html>**

**OUTPUT :**

****

**EXPERIMENT NO. 6**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Demonstrate the use of Java script to define a web server and API call for an application. |
| **Outcome:**  A web application with server connectivity provided using Ajax |
| **Problem Statement:**  Definition of Done  1. Create a weather application taking state name from user  2. It will display the weather details |
| **Background Study:**  A Promise is a proxy for a value not necessarily known when the promise is created. It allows you to associate handlers with an asynchronous action's eventual success value or failure reason. This lets asynchronous methods return values like synchronous methods: instead of immediately returning the final value, the asynchronous method returns a promise to supply the value at some point in the future.  A Promise is in one of these states:   * pending: initial state, neither fulfilled nor rejected. * fulfilled: meaning that the operation was completed successfully. * rejected: meaning that the operation failed.   A pending promise can either be fulfilled with a value or rejected with a reason (error). When either of these options happens, the associated handlers queued up by a promise's then method are called. If the promise has already been fulfilled or rejected when a corresponding handler is attached, the handler will be called, so there is no race condition between an asynchronous operation completing and its handlers being attached.  As the [Promise.prototype.then()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/then) and [Promise.prototype.catch()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/catch) methods return promises, they can be chained. |
| **Question Bank:**   1. *Why are promises?* 2. *Why do we need Ajax?* 3. *Explain the term API.* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 7**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Implement Salary calculator using JavaScript and HTML5. |
| **Outcome:** |
| **Problem Statement:**  **Definition Of Done :**  **DoD1:**A simple salary calculator to show the different components of the salary  **DoD2:** The salary should be calculated based upon the given basic salary in the box  **DoD3:**All the calculated variables such as TA, HRA, DA and taxes should be displayed in the salary calculator box. |
| **Background Study:**  **Document Object Model**  The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects. That way, programming languages can connect to the page.  For example, the standard DOM specifies that the querySelectorAll method in the code below must return a list of all the <p> elements in the document:  const paragraphs = document.querySelectorAll("p");  // paragraphs[0] is the first <p> element  // paragraphs[1] is the second <p> element, etc.  alert(paragraphs[0].nodeName);  **getElementById, innerHTML**   1. getElementById:  To access elements and attributes whose id is set. 2. innerHTML: To access the content of an element.   **Event handler**   1. createElement:  To create new element 2. removeChild: Remove an element 3. You can add an event handler to a particular element |
| **Question Bank:**   1. *Is there any difference between window and document?* 2. *Does document.onload and window.onload fire at the same time?* 3. *What are the different ways to get an element from DOM?* 4. *What is the fastest way to select elements by using css selectors?* 5. *How would you add a class to an element by query selector?* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 8**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Mini Project. |
| **Outcome:**  Specific output based upon the project specification. |
| **Problem Statement:**  **Project-1**  Create a makemytrip online form with the below mentioned details:-  • Id  • Name  • Gender (required, (Male, Female)) – Radio Button  • Password  • Confirm Password  • DOB (required)  • Journey From Place-Drop Down list  • Journey To Place- Drop Down list  • Email Address  • Address(Textbox)  • Phone Number  • Password  • No .of person(Text box)  Validations should be applied for the input fields:  a. Email Address:-Basic email validation – “@”and “.” symbols should be included.  b. DOB, Journey from date and Journey to date cannot be greater than the Current Date.  c. The values entered in the Password and confirm Password should be same  Create a HTML form for the students to fill in the details, with the following DoDs:  1) Form should be responsive.  2) Should be created using HTML5 + CSS3 + Bootstrap + JavaScript.  3) Application should be designed using MVC Framework.  4) Once the person has entered the above details and presses the Submit button, person Details should appear in a Table below.  5) Apply the cookie handling mechanism for remembering the username for revisiting of the web page  **Project-2**  Create a online form for booking of hotel with the below mentioned details:-  • Id  • Name  • Password  • Confirm Password  • Gender (required, (Male, Female)) – Radio Button  • DOB (required)  • Place :-Combo box  • Date of check-in  • Date of check-out  • Meal-Combobox(breakfast, lunch, Dinner)  • Email Address  • Phone Number  • No .of person(Textbox  • Address(Textbox)  Validations should be applied for the input fields:  a. Email Address  Basic email validation – “@”and “.” symbols should be included.  b. DOB, Check-in, check-out cannot be greater than the Current Date.  c. The values entered in the Password and confirm Password should be same  Create a HTML form for the students to fill in the details, with the following DoDs:  1) Form should be responsive.  2) Should be created using HTML5 + CSS3 + Bootstrap + JavaScript.  3) Application should be designed using MVC Framework.  4) Once the person has entered the above details and presses the Submit button, person Details should appear in a Table below  5) Apply the cookie handling mechanism for remembering the username for revisiting of the web page  **Project-3**  Create an online job application form, with the below mentioned detail:-  1. Id  2. Name  3. Password  4. confirm password  5. Age  6. DOB  7. Gender(Radio button)  8. Email  9. Phone Number  10. Address(Textbox)  11. Qualification(Dropdown list)  12. Previous Employment(Textbox)  Validations should be applied for the input fields:  a. Email Address  Basic email validation – “@”and “.” symbols should be included.  b. DOB cannot be greater than the Current Date.  c. The values entered in the Password and confirm Password should be same  Create a HTML form for the students to fill in the details, with the following DoDs:  1) Form should be responsive.  2) Should be created using HTML5 + CSS3 + Bootstrap + JavaScript.  3) Application should be designed using MVC Framework.  4) Once the person has entered the above details and presses the Submit button, person Details should appear in a Table below  5) Apply the cookie handling mechanism for remembering the username for revisiting of the web page |
| **Background Study:** |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 9**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Write the Type Script code to test the syntaxes of the language. |
| **Outcome:**  Command and snippets program output |
| **Problem Statement:**  Definition of Done  DoD1: Define variables of different type and show the type of values the variables can store  DoD2: Demonstrate a looping construct to print the values of an array  DoD3: Define the control flow statement and show the true and false cases |
| **Background Study:** Basic types any  void  boolean  number  string  null  undefined  string[] /\* or Array<string> \*/  [string, number] /\* tuple \*/  string | null | undefined /\* union \*/  never /\* unreachable \*/  enum Color {Red, Green, Blue = 4}  let c: Color = Color.Green Declarations let isDone: boolean  let isDone: boolean = false  function add (a: number, b: number): number {  return a + b  }  // Return type is optional  function add (a: number, b: number) { ... } Type assertionsVariables let len: number = (input asstring).length  let len: number = (<string> input).length /\* not allowed in JSX \*/ Functions function object(this: {a: number, b: number}, a: number, b: number){  this.a = a;  this.b = b;  returnthis;  }  // this is used only for type declaration  let a = object(1,2);  // a has type {a: number, b: number} InterfacesInline functionprintLabel(options: { label: string }){  console.log(options.label)  }  // Note the semicolon  functiongetUser(): { name: string; age?: number } {  } Explicit interfaceLabelOptions{  label: string  }  functionprintLabel(options: LabelOptions){ ... } Optional properties interfaceUser {  name: string,  age?: number  } Read only interfaceUser {  readonly name: string  } Dynamic keys {  [key: string]: Object[]  } Type aliases type Name = string | string[] Function types interfaceUser{ ... }  functiongetUser(callback: (user: User) => any){ callback({...}) }  getUser(function(user: User){ ... }) Classes classPoint {  x: number  y: number  static instances = 0  constructor(x: number, y: number) {  this.x = x  this.y = y  }  } Inheritance classPoint{...}  classPoint3DextendsPoint{...}  interfaceColored{...}  classPixelextendsPointimplementsColored{...} Short fields initialisation classPoint {  static instances = 0;  constructor(  public x: number,  public y: number,  ){}  } Fields which do not require initialisation classPoint {  public someUselessValue!: number;  ...  } Generics classGreeter<T> {  greeting: T  constructor(message: T) {  this.greeting = message  }  }  let greeter = new Greeter<string>('Hello, world') Modules export interfaceUser { ... } Type extraction interface Building {  room: {  door: string,  walls: string[],  };  }  type Walls = Building['room']['walls']; // string[] |
| **Question Bank:**   1. **What are the Advantages and Disadvantages of TypeScript?** 2. **What are the Components of TypeScript?** 3. **Can we combine multiple .ts files into a single .js file?** 4. **What are Variables in TypeScript and how to create them?** 5. **What are the different ways of declaring a Variable?** 6. **What are Classes in TypeScript? List out some of the features.** 7. **What are the access modifiers supported by TypeScript?** 8. **What are modules in TypeScript?** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 10**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Implement the class and object example using Type Script. |
| **Outcome:**  A classobject and class variable values to print using set ad get methods |
| **Problem Statement:**  Definition of Done  DoD1: Create a class product with class variables – pid, pname, price and pdate  DoD2: Create a function to instantiate the object of item class to set and get the values of some of the items  DoD3: Define a function to create an array of this item object. Function should add or delete the object in the array |
| **Background Study:** Classes classPoint {  x: number  y: number  static instances = 0  constructor(x: number, y: number) {  this.x = x  this.y = y  }  } Inheritance classPoint{...}  classPoint3DextendsPoint{...}  interfaceColored{...}  classPixelextendsPointimplementsColored{...} Short fields initialisation classPoint {  static instances = 0;  constructor(  public x: number,  public y: number,  ){}  } Fields which do not require initialisation classPoint {  public someUselessValue!: number;  ...  } Generics classGreeter<T> {  greeting: T  constructor(message: T) {  this.greeting = message  }  }  let greeter = new Greeter<string>('Hello, world') Modules export interfaceUser { ... } Type extraction interface Building {  room: {  door: string,  walls: string[],  };  }  type Walls = Building['room']['walls']; // string[] |
| **Question Bank:**   1. *How to Call Base Class Constructor from Child Class in TypeScript?* 2. *How do you implement inheritance in TypeScript?* 3. *What is the difference between the internal module and the external module?* 4. *What is namespace in Typescript? How to declare a namespace in Typescript?* 5. *What is default visibility for properties/methods in TypeScript classes?* 6. *How does TypeScript support optional parameters in function as in JavaScript every parameter is optional for a function?* 7. *Does TypeScript supports function overloading as JavaScript doesn't support function overloading?* 8. *What are method overriding in TypeScript?* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 11**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Implement the html template and lifecycle-hooks using Angular framework. |
| **Outcome:**  A web application showing different buttons to demonstrate the life cycle of the angular control flow. |
| **Problem Statement:**  Definition of Done  DoD1: Create a HTML template in the default component in Angular  DoD2: Write messages on different events triggered during the life cycle of the Angular project  DoD3: Create different functions to update a static variable to demonstrate the value change on every life cycle hook calls |
| **Background Study:**  A component in Angular has a life-cycle, a number of different phases it goes through from birth to death. We can hook into those different phases to get some pretty fine grained control of our application. To do this we add some specific methods to our component class which get called during each of these life-cycle phases, we call those methods hooks.  The hooks are executed in this order:  Lifecycle Hooks  These phases are broadly split up into phases that are linked to the component itself and phases that are linked to the children of that component. Hooks for the Component constructor  This is invoked when Angular creates a component or directive by calling new on the class.  ngOnChanges  Invoked **every** time there is a change in one of th input properties of the component.  ngOnInit  Invoked when given component has been initialized. This hook is only called **once** after the first ngOnChanges  ngDoCheck  Invoked when the change detector of the given component is invoked. It allows us to implement our own change detection algorithm for the given component.  ngOnDestroy  This method will be invoked just before Angular destroys the component. [Hooks for the Component’s Children](https://codecraft.tv/courses/angular/components/lifecycle-hooks/#_hooks_for_the_component_s_children) ngAfterContentInit  Invoked after Angular performs any content projection into the component’s view  ngAfterContentChecked  Invoked each time the content of the given component has been checked by the change detection mechanism of Angular.  ngAfterViewInit  Invoked when the component’s view has been fully initialized.  ngAfterViewChecked  Invoked each time the view of the given component has been checked by the change detection mechanism of Angular. Adding Hooks In order to demonstrate how the hooks work we’ll adjust the joke application we’ve been working with so far.  Firstly let’s change the JokeComponent so it hooks into all the phases.  All we need to do is to add functions to the component class matching the hook names above, like so:  TypeScript  class JokeComponent {  @Input('joke') data: Joke;  constructor() {  console.log(`new - data is ${this.data}`);  }  ngOnChanges() {  console.log(`ngOnChanges - data is ${this.data}`);  }  ngOnInit() {  console.log(`ngOnInit - data is ${this.data}`);  }  ...  } |
| **Question Bank:** When will ngInit be called?How would you make use of onNgInit()?Why we use - ngAfterContentInit lifecycle method?Which lifecycle method should we use if we execute the code only once?Which lifecycle method should we use if we execute the code only once at the end of the project. |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 12**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Implement the passing parameters/ messageexample among components in Angular framework. |
| **Outcome:**  A message and value passed from one component to other component will be displayed on page as well as in console window |
| **Problem Statement:**  Definition of Done  DoD1: There should be two components and one component should be the child of the other parent  DoD2: A message should be passed from parent component to child component  DoD3: A message should be passed from child component to parent component  DoD4: The message should be written on the web page as well as in console window |
| **Background Study:** Introduction Communication between the Components in Angular will help you to pass data from child components to parent components and vice-versa. Passing Data from Parent to Child Component with Input Binding When we want to pass the data from the parent component to the child component, we use input binding with @Input decorations.  Let's consider an example where PersonChildComponent has two input properties with @Input decorations. As we can see in the below example, we must import Input from '@angular/core' library.   |  | | --- | | import{Component,Input}from'@angular/core';  import{Person}from'./Person';  @Component({  selector:'app-person-child',  template:`  <h3>{{person.name}} says:</h3>  <p>I, {{person.name}}, welcome to Pluralsight, {{masterName}}.</p>  `  })  exportclassPersonChildComponent{  @Input() person:Person;  @Input('master') masterName:string;  } |   We can use aliasing with @Input binding. As we see in the above example, masterName is aliased with the master. Intercepting Input Property Changes with a Setter and ngOnChanges() Intercepting input property helps to act upon a value from the parent. **Changes with setter:** Let's consider an example where we are setting the personname of the input property in the child PersonChildComponent that trims the whitespace from a name and replaces an empty value with default text.  The PersonParentComponent below demonstrates name variations in the personname, including a personname with all spaces.   |  | | --- | | **Child\_Component.ts**  import { Component, Input } from '@angular/core';  @Component({  selector: 'app-personname-child',  template: '<h3>"{{personname}}"</h3>'  })  export class PersonChildComponent {  private \_personname = '';  @Input()  set personname(personname: string) {  this.\_personname = (personname && personname.trim()) || '<no personname set>';  }  get personname(): string { return this.\_personname; }  }  **Parent\_Component.ts**  import { Component } from '@angular/core';  @Component({  selector: 'app-person-parent',  template: `  <h2>Master have {{personnames.length}} personnames</h2>  <app-person-child \*ngFor="let personname of personnames" [personname]="personname"></app-person-child>`  })  export class PersonParentComponent {  // Displays 'Yallaling', '<no person set>', 'Goudar'  personnames = ['Yallaling', ' ', ' Goudar '];  } |  Changes with ngOnChanges(): ngOnChanges() method of the OnChanges lifecycle hook interface detects and acts upon changes to input property values. You may prefer this approach to the property setter when watching multiple, interacting input properties.  Let's consider an example where we have MinmaxChildComponent which detects changes to the minimum and maximum input properties and composes a log message reporting these changes. Passing Data from Child to Parent with Output Binding An Output is an observable property annotated with an @Output decorator, the property always returns an Angular EventEmitter. Values flow out of the component as events bound with an event binding.  In Angular, a component can emit an event using @Output an EventEmitter. Both are parts of the @angular/core.  Let's consider an example where we are emitting the sum value from the component ExampleChildComponent.   |  | | --- | | import{Component,EventEmitter,Output}from'@angular/core';  @Component({  selector:'app-example-child',  template:`<button class='btn btn-primary' (click)="changeValue()">Click me</button> `  })  exportclassExampleChildComponent{  @Output() valueChange =newEventEmitter();  sum =0;  changeValue(){  // You can give any function name  this.sum=this.sum+10;  this.valueChange.emit(this.sum);  }  } |   Let's consider an example where we are going to emit an event and pass a parameter to the event. In the below example, we are emitting a value from ExampleChildComponent to ExampleComponent. Displaying the sum value from ExampleChildComponent.   |  | | --- | | import{Component,OnInit}from'@angular/core';  @Component({  selector:'app-example',  template:`<app-example-child (changeValue)='displaySum($event)'></app-example-child>`  })  exportclassExampleComponentimplementsOnInit{  ngOnInit(){  }  displaySum(sum){  console.log(sum);  }  } | |
| **Question Bank:**   1. *In Angular, how can you interact between Parent and Child components?* 2. *What is the main difference between constructor and ngOnInit?* 3. *What modules should you import in Angular to use [(ngModel)] and reactive forms?* 4. *How many Change Detectors can there be in the whole application?* 5. *What change detection strategies do you know?* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 13**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Implement the service injection module in Angular framework. |
| **Outcome:**  A date and time component to be displayed in footer section of every components’ HTML template. |
| **Problem Statement:**  Definition of Done  DoD1: A service module need to be define to show the date and time  DoD2: This component will provide their service to every component of the applications  DoD3: Every components will show the date and time in the footer section of the component.  DoD4: Service module should be updated as per the life cycle of the angular project |
| **Background Study:** Introduction We might come across a situation where we need some code to be used everywhere on the page. It can be for data connection that needs to be shared across components, etc. Services help us achieve that. With services, we can access methods and properties across other components in the entire project.  To create a service, we need to make use of the command line. The command for the same is –  C:\projectA4\Angular 4-app>ng g service myservice  installing service  create src\app\myservice.service.spec.ts  create src\app\myservice.service.ts  WARNING Service is generated but not provided, it must be provided to be used  C:\projectA4\Angular 4-app>  The default service component file would be like:  import { Injectable } from '@angular/core';  @Injectable()  export class MyserviceService {  constructor() { }  }  Here, the Injectable module is imported from the @angular/core. It contains the @Injectable method and a class called MyserviceService. We will create our service function in this class.We have imported the Service with the class name and the same class is used in the providers. Let us now switch back to the service class and create a service function.  In the service class, we will create a function which will display today’s date. We can use the same function in the main parent component app.component.ts and also in the new component new-cmp.component.ts that we created in the previous chapter.  Let us now see how the function looks in the service and how to use it in components.  import { Injectable } from '@angular/core';  @Injectable()  export class MyserviceService {  constructor() { }  showTodayDate() {  let ndate = new Date();  return ndate;  }  }  In the above service file, we have created a function **showTodayDate**. Now we will return the new Date () created. Let us see how we can access this function in the component class.  The ngOnInit function gets called by default in any component created. The date is fetched from the service as shown above. To fetch more details of the service, we need to first include the service in the component ts file. |
| **Question Bank:**   1. *How is Dependency Hierarchy formed?* 2. *What is the purpose of async pipe?* 3. *What is a service worker and its role in Angular?* 4. *What are the design goals of service workers?* 5. *How do you provide a singleton service?* 6. *Is it mandatory to use injectable on every service class?* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 14**

|  |
| --- |
| **Student Name and Roll Number: Avtar Singh 20csu41** |
| **Semester /Section: 5th / FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Demonstrate the importance of routing in page access and show the features available in Angular framework. |
| **Outcome:**  An Angular application with linked components to demonstrate the routing. |
| **Problem Statement:**  Definition of Done  DoD1: A root component with hyperlinked child components  DoD2: A child components to be loaded with respective URL  DoD3: A linked component to be loaded along with respective URL |
| **Background Study:** Introduction The Angular router is an essential element of the Angular platform. It allows developers to build Single Page Applications with multiple states and views using routes and components and allows client side navigation and routing between the various components. It’s built and maintained by the core team behind Angular development and it’s contained in the @angular/router package.  You can use the browser's URL to navigate between Angular components in the same way you can use the usual server side navigation. Routes and Paths In Angular, a **route** is an object (instance of Route) that provides information about which component maps to a specific path. A **path** is the fragment of a URL that determines where exactly is located the resource (or page) you want to access. You can get the path by taking off the domain name from the URL.  In Angular you can define a route using route configurations or instances of the Route interface.  A collection of routes defines the router configuration which is an instance of Routes.  Each route can have the following properties:  pathis a string that specifies the path of the route.  pathMatchis a string that specifies the matching strategy. It can take prefix (default) or full.  componentis a component type that specifies the component that should be mapped to the route.  [redirectTo](https://angular.io/api/router/Route#redirectTo)is the URL fragment which you will be redirected to if a route is matched.  These are the commonly used properties of routes but there are many others.  For example, this is the definition of a route that maps the /my/path/ path to the MyComponent component:  { path: 'my/path/', component: MyComponent }  The path can be the empty string which usually refers to the main URL of your application or can be also a **wildcard** string (\*\*) which will be matched by the router if the visited URL doesn’t match any paths in the router configuration. This is usually used to display a **page doesn’t exist** message or redirect the users to an existing path. Route Matching Strategies The Angular router has a powerful matching algorithm with various built-in and custom matching strategies.  The built-in matching strategies are **prefix** (the default) and **full.**  When the matching strategy of a route is **prefix**, the router will simply check if the start of the browser’s URL is prefixed with the route’s path. If that’s the case, it will render the related component.  This is not always the wanted behavior. In some scenarios, you want the router to match the full path before rendering a component. You can set the full strategy using the pathMatch property of a route. For example:  { path: 'products', pathMatch: 'full', component: ProductListComponent}  A full strategy ensures that the path segment of browser’s URL equals exactly the route’s path.  A special case of using the full property is when you want to match the empty path. Because using the prefix strategy will match all paths since the empty path prefixes all paths.  For example, we want to redirect the user to the /products route when they visit our application. This is how our route configuration should look like:  { path: '', redirectTo: '/products', pathMatch: 'full' }  You can also use custom matcher if the combination of the path property and matching strategy doesn’t help you match your component to a specific URL.  You can provide a custom matcher using the matcher property of a route definition. For an example, see [UrlMatcher](https://angular.io/api/router/UrlMatcher#description). Route Parameters Dynamic routes are often used in web applications to pass data (parameters) or state to the application or between various components and pages. The Angular router has support for dynamic paths and provides an easy to use API to access route parameters.  You can define a route parameter using the colon syntax followed by the name of the parameter. For example:  {path: 'product/:id' , component: ProductDetailComponent}  In the example, id is the route parameter. /product/1, /product/2, /product/p1 … are examples of URLs that will be matched via this route definition.  The last segment of these URLs are the values of the id parameter that will be passed to ProductDetailComponent.  In your matched components, you can access the route parameters using various APIs:   * Using the ActivatedRoute service, * Using the ParamMap Observable available starting with Angular 4.  Angular Route Guards Route guards enables you to allow or disallow access to your specific application routes based on some criteria (for example if the user is logged in or not).You can also use a guard to prevent users from leaving a component depending on some conditions (for example if a form is not submitted yet and data can be lost).You can use Angular guards to protect components or complete modules.To protect a route, you first need to create a guard by sub-classing the CanActivate interface and overriding the canActivate() method which needs to return a Boolean value (true means access is allowed) and add it to route definition via the canActivate attribute. For example:  { path: 'product/:id, canActivate:[ExampleGuard], component:ProductDetailComponent}  This is an example implementation of ExampleGuard:  class MyGuard implements CanActivate {  canActivate() {  return true;  }  }  Since the canActivate() method will always return true, this guard will always allow access to ProductDetailComponent. The Router Outlet The [Router-Outlet](https://angular.io/api/router/RouterOutlet) is a directive exported by RouterModule and acts as a placeholder that indicates to the router where it needs to insert the matched component(s). The component that has the router outlet is referred to as the **application shell:**  <router-outlet></router-outlet>  The Angular router supports more than one outlet in the same application. The main (or top-level) outlet is called the **primary outlet.** Other outlets are called **secondary outlets**.  You can specify a target outlet for a route definition using the outlet attribute. The Navigation Directives Angular Router provides two directives for navigation: The routerLink directive which replaces the href attribute in the <a> tags to create links and routerLinkActive for marking the active link. For example:  <a [routerLink]="'/products'">Products</a> |
| **Question Bank:**   1. *How would you add a hyperlink in a view?* 2. *If a route parameter needed to be defined for an application where would you put it?* 3. *How would you use a router outlet to display a view?* 4. *Which angular package would you use to route to URLs?* 5. *Where would you put the value for a route parameter?* 6. *Where would you define child URL routes?* |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**Annexure 2**

**Web Framework**

**CSL253**

Project Report



Faculty name: Student name:

Roll No.:

Semester:

Group:

**Department of Computer Science and Engineering**

**The NorthCap University, Gurugram- 122001, India**

**Session 2022-23**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** |  | **Page No.** |
| **1.** | **Project Description** |  |
| **2.** | **Problem Statement** |  |
| **3.** | **Analysis**  **3.1 Hardware Requirements**  **3.2 Software Requirements** |  |
| **4.** | **Design**  **4.1 Data/Input Output Description:**  **4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps** |  |
| **5.** | **Implementation and Testing (stage/module wise)** |  |
| **6.** | **Output (Screenshots)** |  |
| **7.** | **Conclusion and Future Scope** |  |