

Practicing 4

Date: 11/09/2021

1. Name of Practitioner: Santhi HITHA, W/Engg

2. Procedure: (100)

1. Begins with a standard HTML document
2. Use <h1>, <p>, <hr>, and tags to display personal details and an image.
3. Use tag to describe hobbies and social media links
4. Use headings and paragraphs to format text
5. Use the <table> tag with <tr>, <td>, and <th> elements
6. Use an ordered list () to present technical skills.
7. Use an unordered list () for general items
8. Complete and save the document.

3. Actual Resources Used: (100)

Resource	Tool & Version
Operating System	Windows 11.0 Desktop
Programming Language	HTML (Markup Language)
Software Development Kit	Not Applicable
Libraries	Not Applicable
Application/Software Tool	Not Applicable
Hardware Tool	1. IDE Code (v 1.20.1) 2. Google Chrome (v 110.0.1200.07)
Backend Tool	Not Applicable
Frontend Tool	Not Applicable
Specific Software	Not Applicable

4. Actual Procedure Followed:

Planning & Requirements Gathering

- Identified the purpose of the webpage.
- Learned the required features.
- Chose the theme, layout, color, and video software.

Setting Up the Development Environment

- Installed Visual Studio Code.
- Created a project folder and subfolders for HTML and images.

Determining the Structure

- Wrote the basic HTML skeleton.
- Added headings, paragraphs, images, and hyperlinks as per the design.

Testing & Debugging

- Viewed the webpage in multiple browsers (Chrome, Edge, etc.) to check compatibility.
- Fixed layout issues and corrected broken links or missing images.

Finalizing & Deployment

- Reviewed the webpage for errors and typos.
- Saved and organized all files in the correct folders.
- Uploaded the files to the domain.

5. Observations:

- HTML tags were used to give all levels content and position different tags in final document.
- Class paper source code has been verified by <https://codecademy.com> and found errors of LTR not being used.
- Normal font, different tags rendered on the webpage.
- displayed with dots, while emphasized header content with bold styling.
- Changing the type attribute of the ordered list () affected the numbering style (e.g., numeric, alphabetic, Roman numerals).
- The <hr> element is used for sections, etc., and linking stylescripts, while the <hr/> content style content.
- Semantic tags like <h1>, <h2>, <h3>, and <h4> improve readability and accessibility.
- Link , help in presenting information in an organized manner.
- The main thing of this procedure is adding tag to the webpage and it shows perfectly.

4. Search Code

Effect CSS Basic Page

0. Base HTML

```
<DOCTYPE html>
```

```
<html>
```

```
<title>First CSS Image Filter</title>
```

```
<meta charset="UTF-8" />
```

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

```
<link href="css/style.css" rel="stylesheet" type="text/css" />
```

```
</html>
```

```
<body>
```

```
<div class="Image-1" style="text-align: center;>
```

```
<img alt="Image 1" />
```

```
</div>
```

```
|
```

```
<div class="Image-2" style="text-align: center;>
```

```
<img alt="Image 2" />
```

```
</div>
```

```
|
```

```
<div class="Image-3" style="text-align: center;>
```

```
<img alt="Image 3" />
```

```
</div>
```

```
|
```

```
<div class="Image-4" style="text-align: center;>
```

```
<img alt="Image 4" />
```

```
</div>
```

```
|
```

```
<div class="Image-5" style="text-align: center;>
```

```
<img alt="Image 5" />
```

```
</div>
```

```
|
```

```
<div class="Image-6" style="text-align: center;>
```

```
<img alt="Image 6" />
```

```
</div>
```

```
|
```

```
<div class="Image-7" style="text-align: center;>
```

```
<img alt="Image 7" />
```

```
</div>
```

```
|
```

```
<div class="Image-8" style="text-align: center;>
```

```
<img alt="Image 8" />
```

```
</div>
```

```
|
```

```
<div class="Image-9" style="text-align: center;>
```

```
<img alt="Image 9" />
```

```
</div>
```

```
|
```

```
<div class="Image-10" style="text-align: center;>
```

```
<img alt="Image 10" />
```

```
</div>
```

```
|
```

```
<div class="Image-11" style="text-align: center;>
```

```
<img alt="Image 11" />
```

```
</div>
```

```
</body>
```



```
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  3. 3.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  4. 4.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  5. 5.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  6. 6.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  7. 7.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  8. 8.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  9. 9.1  
    name: 'My  
    type: My  
    namespace:   
    location: 1 2 3 4 5 6 7 8 9  
    value: 10
```

```
  10. 10.1
```

```
  11. 11.1
```

```
  12. 12.1 (My Namespace)
```

```
  13. 13.1 (My Namespace)
```

```
  14. 14.1 (My Namespace)
```

```
  15. 15.1 (My Namespace)
```

```
  16. 16.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  17. 17.1
```

```
  18. 18.1 (My Namespace) (My Namespace) (My Namespace)
```

```
  19. 19.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  20. 20.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  21. 21.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  22. 22.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  23. 23.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  24. 24.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  25. 25.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  26. 26.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  27. 27.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  28. 28.1 (My Namespace) (My Namespace) (My Namespace) (My Namespace)
```

```
  29. 29.1
```

```
  30. 30.1
```

146

15.1 Example 10

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { apply(v, g); }

-using T = Eigen::T; using U =

U;

15.1 class "vector example"

-using T = Eigen::T; using U =

U; void f() { for (int i = 0; i < v.size();

i++) v[i] = v[i] * 2; }

15.1

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++) v[i] = 2 * v[i]; }

15.1

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++) v[i] = 2 * v[i]; }

15.1

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++)

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++)

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++)

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++)

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++)

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++) v[i] = 2 * v[i]; }

15.1

15.1 class "vector example"

-using Eigen::Vector = Eigen::Vec<

int, 3, Eigen::Dense>;

-void f() { for (int i = 0; i < v.size();

i++) v[i] = 2 * v[i]; }


```

    Test case 1: empty list with null values
    Test case 2: list with null values
    Test case 3: list with non-null values

```

```

    Test case 4: list with null values
    Test case 5: list with non-null values

```

```

    Test case 6:

```

```

    Test case 7:

```

```

    Test case 8:

```

```

    Test case 9:

```

```

Test case 10:

```

```

Test case 11:

```

```

Test case 12:

```

```

Test case 13:

```

```

Test case 14:

```

```

    Test case 15: list with null values

```

```

    Test case 16: list with null values

```

```

    Test case 17: list with null values and non-null values

```

```

    Test case 18: list with null values and non-null values

```

```

    Test case 19:

```

```

    Test case 20:

```

```

    Test case 21: list with null values

```

```

    Test case 22: list with null values

```

```

    Test case 23:

```

```

    Test case 24:

```

```

    Test case 25:

```

```

    Test case 26:

```

```

    Test case 27:

```

```

    Test case 28:

```

```

    Test case 29:

```

```

    Test case 30: list with null values

```

```

    Test case 31:

```

```

    Test case 32:

```

```

    Test case 33:

```

```

    Test case 34:

```

```

    Test case 35:

```

```

    Test case 36:

```

```

    Test case 37:

```

```

    Test case 38:

```

```

    Test case 39:

```

```

    Test case 40:

```

```

    Test case 41:

```

```

    Test case 42:

```

```

    Test case 43:

```

```

    Test case 44:

```

```

    Test case 45:

```

```

    Test case 46:

```

```

    Test case 47:

```

```

    Test case 48:

```

```

    Test case 49:

```

```

    Test case 50:

```

```

    Test case 51:

```

```

    Test case 52:

```

```

    Test case 53:

```

```

    Test case 54:

```

```

    Test case 55:

```

```

    Test case 56:

```

```

    Test case 57:

```

```

    Test case 58:

```

```

    Test case 59:

```

```

    Test case 60:

```



```

    @Override
    public void start() {
        // Start the server
        try {
            ServerSocket serverSocket = new ServerSocket(8080);
            while (true) {
                Socket clientSocket = serverSocket.accept();
                // Create a new thread to handle the client
                new Thread(new ClientHandler(clientSocket)).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    // ClientHandler class
    class ClientHandler implements Runnable {
        private Socket clientSocket;

        ClientHandler(Socket clientSocket) {
            this.clientSocket = clientSocket;
        }

        @Override
        public void run() {
            try {
                // Read the request
                String request = clientSocket.getInputStream().readLine();
                // Parse the request
                String[] requestParts = request.split(" ");
                String method = requestParts[0];
                String uri = requestParts[1];
                String version = requestParts[2];

                // Handle the request
                String response = "HTTP/1.1 200 OK\n";
                response += "Content-Type: text/html\n";
                response += "\n";
                response += "<h1>Hello, World!</h1>";

                // Write the response
                clientSocket.getOutputStream().write(response.getBytes());
                clientSocket.getOutputStream().flush();

                // Close the client socket
                clientSocket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```


Three word

1247

12) class "Two example table"

→ class table Weight → class table
→ class table weight table → class table
→ class table

1248

13) class "Two example table"

→ class table Weight → class table
→ class table weight table → class table
→ class table

1249

14) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1250

15) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1251

16) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1252

17) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1253

18) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1254

19) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1255

20) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1256

21) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1257

22) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

1258

23) class "Two example table"

→ class table table → class table
→ class table table → class table
→ class table

149

14. **Chickadee** (1)

- 42-100-0000-1 (Blue 1) (Blue 1) (Blue 1)
- 42-100-0000-2 (Blue 2) (Blue 2) (Blue 2)
- 42-100-0000-3 (Blue 3) (Blue 3) (Blue 3)

150

- 42-100-0000-4 (Blue 4) (Blue 4) (Blue 4)
- 42-100-0000-5 (Blue 5) (Blue 5) (Blue 5)
- 42-100-0000-6 (Blue 6) (Blue 6) (Blue 6)
- 42-100-0000-7 (Blue 7) (Blue 7) (Blue 7)
- 42-100-0000-8 (Blue 8) (Blue 8) (Blue 8)

151

152

- 42-100-0000-9 (Blue 9) (Blue 9) (Blue 9)
- 42-100-0000-10 (Blue 10) (Blue 10) (Blue 10)
- 42-100-0000-11 (Blue 11) (Blue 11) (Blue 11)

153

154

- 42-100-0000-12 (Blue 12) (Blue 12) (Blue 12)
- 42-100-0000-13 (Blue 13) (Blue 13) (Blue 13)
- 42-100-0000-14 (Blue 14) (Blue 14) (Blue 14)

155

156

- 42-100-0000-15 (Blue 15) (Blue 15) (Blue 15)
- 42-100-0000-16 (Blue 16) (Blue 16) (Blue 16)
- 42-100-0000-17 (Blue 17) (Blue 17) (Blue 17)

157

158

- 42-100-0000-18 (Blue 18) (Blue 18) (Blue 18)
- 42-100-0000-19 (Blue 19) (Blue 19) (Blue 19)
- 42-100-0000-20 (Blue 20) (Blue 20) (Blue 20)

159

160

- 42-100-0000-21 (Blue 21) (Blue 21) (Blue 21)
- 42-100-0000-22 (Blue 22) (Blue 22) (Blue 22)
- 42-100-0000-23 (Blue 23) (Blue 23) (Blue 23)

161

162

- 42-100-0000-24 (Blue 24) (Blue 24) (Blue 24)
- 42-100-0000-25 (Blue 25) (Blue 25) (Blue 25)
- 42-100-0000-26 (Blue 26) (Blue 26) (Blue 26)

163

164

- 42-100-0000-27 (Blue 27) (Blue 27) (Blue 27)
- 42-100-0000-28 (Blue 28) (Blue 28) (Blue 28)

165

- 42-100-0000-29 (Blue 29) (Blue 29) (Blue 29)
- 42-100-0000-30 (Blue 30) (Blue 30) (Blue 30)
- 42-100-0000-31 (Blue 31) (Blue 31) (Blue 31)
- 42-100-0000-32 (Blue 32) (Blue 32) (Blue 32)
- 42-100-0000-33 (Blue 33) (Blue 33) (Blue 33)
- 42-100-0000-34 (Blue 34) (Blue 34) (Blue 34)
- 42-100-0000-35 (Blue 35) (Blue 35) (Blue 35)
- 42-100-0000-36 (Blue 36) (Blue 36) (Blue 36)
- 42-100-0000-37 (Blue 37) (Blue 37) (Blue 37)
- 42-100-0000-38 (Blue 38) (Blue 38) (Blue 38)
- 42-100-0000-39 (Blue 39) (Blue 39) (Blue 39)
- 42-100-0000-40 (Blue 40) (Blue 40) (Blue 40)

166

167. **Blue Jay** (1)

- 42-100-0000-41 (Blue 41) (Blue 41) (Blue 41)

191

192

193-194

195-200 (continued) (page 195)

201

202-203

204-210 (continued) (page 204)

211

212-213

214-220 (This page is blank at home)

221

222-223

224-230 (continued)

231

232-233

234-240 (continued)

241

242-243

244-250 (continued)

251

252-260 (continued)

261

262-270 (This page is blank at home)

271

272-280 (This page is blank at home)

281

282-290 (This page is blank at home)

291

292-300 (This page is blank at home)

301

302-310 (This page is blank at home)

311

312-320 (This page is blank at home)

321

322-330 (This page is blank at home)

331

332-333

334

335

336

337

338

339

340

2. Results
1. Data Collection
1. Results



1. Results



8.2.1.1.1.1



8.2.1.1.1.2





EXERCISE 2

A screenshot of a web form titled 'User Profile' with a light blue background. The form is divided into three sections with light blue headers. The first section, 'Profile Information', includes fields for name, email, and a dropdown menu. The second section, 'Account Settings', has fields for password and a checkbox. The third section, 'Security Settings', includes a field for security questions and a checkbox. The form is centered on a white background.A screenshot of a web form titled 'Registration Form' with a light blue background. The form is divided into three sections with light blue headers. The first section, 'Personal Details', includes fields for name, date of birth, and gender. The second section, 'Contact Information', has fields for email and phone number. The third section, 'Account Information', includes fields for password, a confirmation password, and a checkbox. The form is centered on a white background.



Y1 Shopping List





11. MSF, Windows 6/200







7.2 Solution Output:

ii) Pure CSS Demo:

librews.html



librews.html



11.10.2017



11.10.2017



VideoBeitrag



III. MSN's Kibitz mit USG UMSC, Kibitz



8. Sitzung



9. Sitzung



Test Case Report: Proctoren I

Librarian On COS

Test Case Name:	Using the Home page of LIB
Module Name:	Form
Test Priority:	Importance of Student
Test Designed By:	Jack Julius Chris Vidal
Test Reviewed By:	DAI Ilin
Test Designed Date:	24-09-2020
Test Executed By:	Jack Julius Chris Vidal
Test Executed Date:	4-10-2020
Test Summary:	Page displays HTML icons (login with user buttons (Home, Logout, New Menu, Login). Navigation works with correct effects, and your name appears in the bottom right.
Pre-Requisites:	Chrome - Version (89.1) VSCode - Version (1.51.0)
Expected Results:	Not applicable

Test Case ID	Module / Feature	Test Scenario / Description	Input Data	Expected Output	Actual Output	Status (Pass/Fail)	Remarks / Notes
TC001	Logging in Feature	Check if the user can log in to the system	Valid Username & Password	Successful login with user ID and name	Yes	Pass	
TC002	Link generation and display	Checking if the links generated are clickable and working	Clicking the links in the system	All links are clickable and working as expected	Yes	Pass	
TC003	Form validation	Check the form validation and error messages	Click the form to submit the required data	All the form validation and error messages are displayed	Yes	Pass	
TC004	Broken images and links	Checking for broken images and links	Open any one of the images	Images and links are properly displayed	Yes	Pass	

TC08 8	CR1	Check whether CR1 is used properly or not!	Check CR1 status in the vehicle	CR1 is used in a well good manner	Yes	Pass	
TC09 8	Vehicle Status	Check the Vehicle status	Check vehicle status	Check the vehicle status properly	Yes	Pass	
TC10 7		Check the body frame type like -metal -plastic -composite	Verify whether it is -metal -plastic -composite	It is done properly	Yes	Pass	
TC11 8	CR1	Check the CR1 & used properly or not	Checking through a CR1	Make sure that the CR1 is used properly	Yes	Pass	
TC12 8	Vehicle on	Check whether the vehicle is on or not	Vehicle is ON	Vehicle is on	Yes	Pass	

Lilly's Kitchen

Est Title Name	Young Lilly's Kitchen
Website Name	Lilly's Kitchen
Est Priority	Important level Medium
Est Designed By	Jack Antonia Chris Vimal
Est Reviewed By	Est Title
Est Designed Date	21-05-2025
Est Examined By	Jack Antonia Chris Vimal
Est Examined Date	21-05-2025
Est Summary	Recipe webpage displaying Chicken Fry, Gulab Jamun, and Shrimp Lassi with ingredients, procedures, video, images and links. Customizes for use of different HTML tags and includes custom links and social media tags.
Pre-Scriptives	Chrome Version: 1.04.1 Visio Version: 1.04.1.2
Dependencies	Not applicable

Test Case ID	Module / Features	Test Scenario / Description	Input Data	Expected Output	Actual Output	Status (Pass/Fail)	Remark or Notes
TC001	Logging in Module	Check if the page loads successfully	Open: URL: http://www.example.com	Website loads with the login button	Yes	Pass	
TC002	Forgot Login	Checking if the forgot login link is working or not	Page number: 1234567890 Clicked on forgot	Page number jumps to 1234567890	Yes	Pass	
TC003	Search for User	Check the search for user functionality if the search is successful	Click "Search for User"	All users listed with their names	Yes	Pass	
TC004	User Display	Check whether the users are displayed	View user profile e.g. User Name: John Doe	Users listed correctly with their details	Yes	Pass	

T00 05	Video Editing 1	Check whether the Video is Accepted.	Check the Video is visible 1	Video duration is correct and good	Yes	Pass	
T00 06	Video File Format	Check the Video file format	Check video format is correct	Links from the YouTube are correctly placed	Yes	Pass	
T00 07	File Structure 1	Check the file structure highlights	Verify structure and file names	Highlight or add links	Yes	Pass	
T00 08	Content Links	Check the content links and add links or unlink	Check "Add to" "Add to" link	Check link is not broken or is	Yes	Pass	
T00 09	Content is	Check whether the link and text are correct	Verify link is working	No content errors	Yes	Pass	

8. Results:

- The original webpage
- 1. Home page for CSU
- 2. About/Welcome

- The original file was chosen to present various skills in a structured, ordered format
- Successfully created a functional webpage using appropriate HTML tags
- Correctly applied tags for headings, paragraphs, images, links, tables, and lists
- Utilized proper HTML structure with <DOCTYPE>, <html>, <head>, and <body> tags
- Used CSS to give the website webpage
- Validated the webpage using W3C HTML Validation with no major errors
- Successfully added Title and Meta tag to the code

9. INTERPRETATION OF RESULTS

-CSU Home

This website contains the guidelines of basic CSS elements.

-Mall's Kitchen

The webpage consists of 7 images of my favorite which my Momma makes for me with links for Guidance with CSS address making food amounts.

RECONCLUSION

- Successfully created a functional webpage using HTML
- Utilized the code structure of different HTML tags
- Ensured the webpage was W3C standard-compliant after validation
- Learned the importance of class and property structural code
- Gained hands-on experience in basic web development practice
- Used CSS to give the website webpage