

## Practical: - 1

**Aim:** Use of Basic Tags.

➤ Basic HTML

Tag	Description
<u>&lt;!DOCTYPE&gt;</u>	Defines the document type
<u>&lt;html&gt;</u>	Defines an HTML document
<u>&lt;head&gt;</u>	Contains metadata/information for the document
<u>&lt;title&gt;</u>	Defines title for the document
<u>&lt;body&gt;</u>	Defines the document's body
<u>&lt;h1&gt; to &lt;h6&gt;</u>	Defines HTML headings
<u>&lt;p&gt;</u>	Defines paragraph
<u>&lt;br&gt;</u>	Inserts single line break
<u>&lt;hr&gt;</u>	Defines thematic change in the content
<u>&lt;!--...--&gt;</u>	Defines comment

## Practical: - 2

**Aim:** Write HTML program for timetable.

**Code:**

```
<!DOCTYPE html>
<html>
<body>
    <h1>TIME TABLE</h1>
    <table border="5" cellspacing="0" align="center">
        <tr>
            <td align="center" height="50"
                width="100"><br>
                <b>Day/Period</b><br>
            </td>
            <td align="center" height="50"
                width="100">
                <b>I<br>9:30-10:20</b>
            </td>
            <td align="center" height="50"
                width="100">
                <b>II<br>10:20-11:10</b>
            </td>
            <td align="center" height="50"
                width="100">
                <b>III<br>11:10-12:00</b>
            </td>
            <td align="center" height="50"
                width="100">
                <b>12:00-12:40</b>
            </td>
            <td align="center" height="50"
                width="100">
                <b>IV<br>12:40-1:30</b>
            </td>
            <td align="center" height="50"
                width="100">
                <b>V<br>1:30-2:20</b>
            </td>
            <td align="center" height="50"
                width="100">
                <b>VI<br>2:20-3:10</b>
            </td>
```

```
<td align="center" height="50"
    width="100">
    <b>VII<br>3:10-4:00</b>
</td>
</tr>
<tr>
    <td align="center" height="50">
        <b>Monday</b></td>
        <td align="center" height="50">Library</td>
        <td align="center" height="50">Physics</td>
        <td align="center" height="50">Maths</td>
        <td rowspan="6" align="center" height="50">
            <h2>L<br>U<br>N<br>C<br>H</h2>
        </td>
        <td colspan="3" align="center"
            height="50"> BEE LAB</td>
        <td align="center" height="50">Phy</td>
    </tr>

    <tr>
        <td align="center" height="50">
            <b>Tuesday</b>
        </td>
        <td colspan="3" align="center"
            height="50"> PPS LAB
        </td>
        <td align="center" height="50">Library</td>
        <td align="center" height="50">Maths</td>
        <td align="center" height="50">BEE</td>
        <td align="center" height="50">PPS</td>
    </tr>
    <tr>
        <td align="center" height="50">
            <b>Wednesday</b>
        </td>
        <td align="center" height="50">BEE</td>
        <td align="center" height="50">PPS</td>
        <td align="center" height="50">Maths</td>
        <td align="center" height="50">Physics</td>
        <td colspan="3" align="center"
            height="50">
                <h2>P<br>E<br>C<br>S<br>H</h2>
            </td>
    </tr>
```

```
        height="50">Physics LAB
    </td>
</tr>
<tr>
    <td align="center" height="50">
        <b>Thursday</b>
    </td>
    <td align="center" height="50">Maths</td>
    <td align="center" height="50">PPS</td>
    <td align="center" height="50">Presentation</td>
    <td colspan="3" align="center"
        height="50">Workshop (PPS)
    </td>
    <td align="center" height="50">Maths</td>
</tr>
<tr>
    <td align="center" height="50">
        <b>Friday</b>
    </td>
    <td colspan="3" align="center"
        height="50">Workshop (BEE)
    </td>

    <td align="center" height="50">PPS</td>
    <td align="center" height="50">Maths</td>
    <td align="center" height="50">Physics</td>
    <td align="center" height="50">Maths (tutorial)</td>
</tr>
<tr>
    <td align="center" height="50">
        <b>Saturday</b>
    </td>
    <td align="center" height="50">Yoga</td>
    <td align="center" height="50">Sports</td>
    <td align="center" height="50">Sports</td>
    <td colspan="3" align="center"
        height="50">SEMINAR
    </td>
    <td align="center" height="50">Sports</td>
</tr>
</table>
</body>
</html>
```

**Output:**

My Drive - Google Drive WORKSHOP-LAB-MANU timetable.html timetable.html - Workshop

← → C File | E:/time%20table/timetable.html

## TIME TABLE

Day/Period	I 9:30-10:20	II 10:20-11:10	III 11:10-12:00	12:00-12:40	IV 12:40-1:30	V 1:30-2:20	VI 2:20-3:10	VII 3:10-4:00	
Monday	Library	Physics	Maths	L U N C H	BEE LAB			Phy	
Tuesday	PPS LAB				Library	Maths	BEE	PPS	
Wednesday	BEE	PPS	Maths		Physics	Physics LAB			
Thursday	Maths	PPS	Presentation		Workshop(PPS)			Maths	
Friday	Workshop(BEE)				PPS	Maths	Physics	Maths(tutorial)	
Saturday	Yoga	Sports	Sports		SEMINAR			Sports	

## Practical: - 3

**Aim:** Write an HTML program for demonstrating Hyperlinks. a. Navigation from one page to another.

**Code:**

```
<!DOCTYPE html>
<html>

<head>
    <title>
        YASH.COM
    </title>

    <style>
        .gfg {
            font-size: 40px;
            color: #090;
            font-weight: bold;
            text-align: center;
        }

        .nav_tag {
            text-align: center;
            margin: 30px 0;
            font-size: 35px;
        }
    </style>
</head>

<body>
    <div class="gfg">
        Demonstrating Hyperlinks. A. Navigation
    </div>

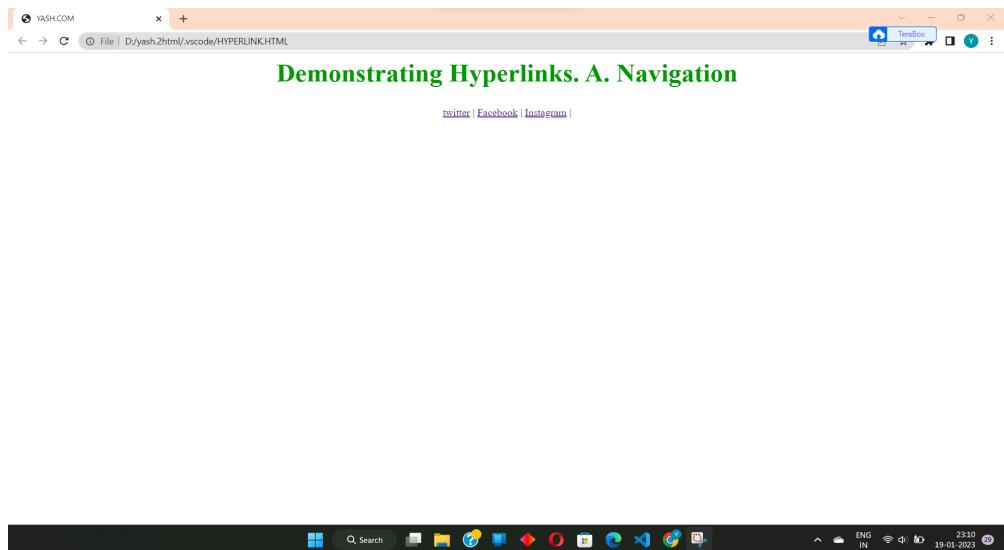
    <div class="nav_tag">
    </div>

    <nav align="center">
```

```
< href="https://twitter.com/yashTiwari893">twitter</a> |
<
Facebook
</a> |
<
Instagram</a> |
</nav>
</body>

</html>
```

## Output:



## Practical: - 4

**Aim:** Write HTML for demonstration of cascading style sheets.

- a. Internal stylesheets.
- b. External stylesheets.
- c. Inline styles.

**Code:**

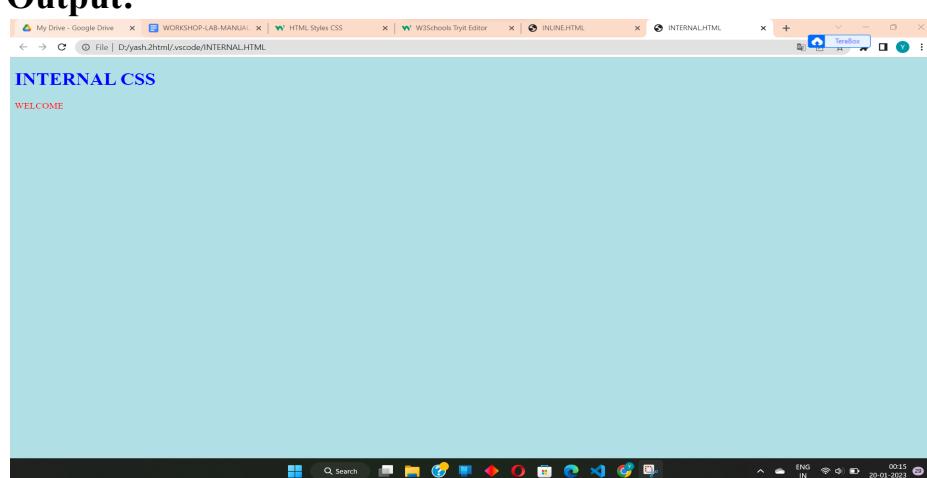
a. Internal stylesheets:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powder blue;}
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>

<h1>INTERNAL CSS</h1>
<p>WELCOME</p>

</body>
</html>
```

**Output:**

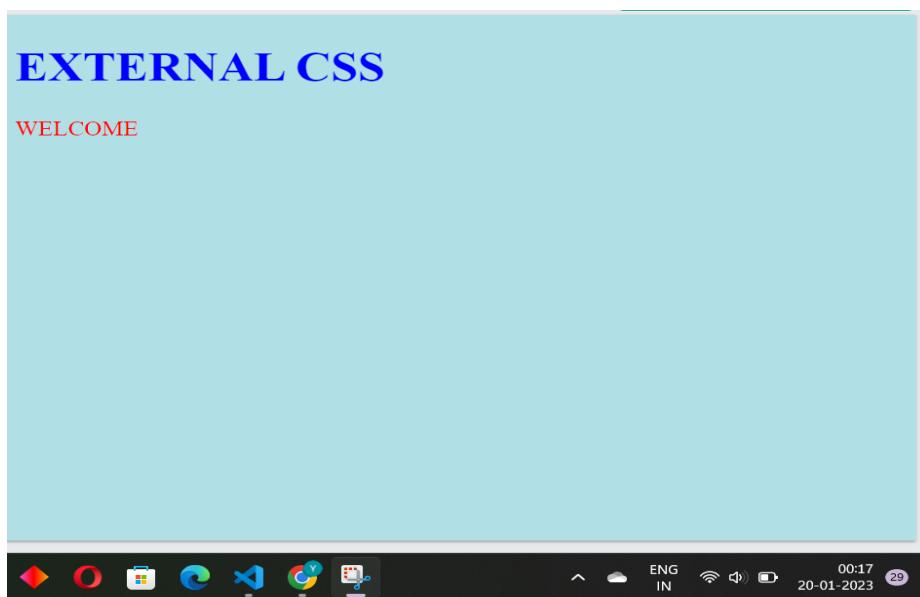


**b. External stylesheets:**

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

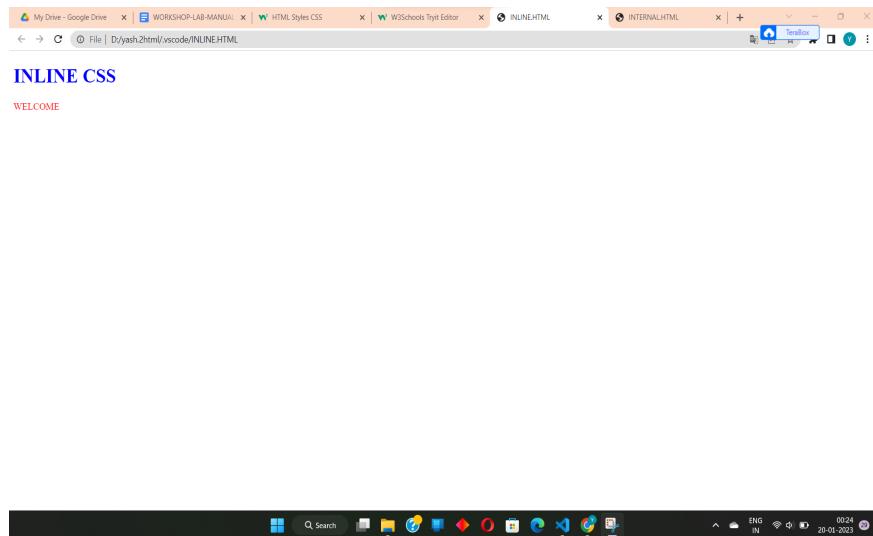
<h1>EXTERNAL CSS</h1>
<p>WELCOME</p>

</body>
</html>
```

**Output:**

**c. Inline styles:**

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="color:blue;">INLINE CSS</h1>
    <p style="color:red;">WELCOME</p>
</body>
</html>
```

**Output:**

## Practical: - 5

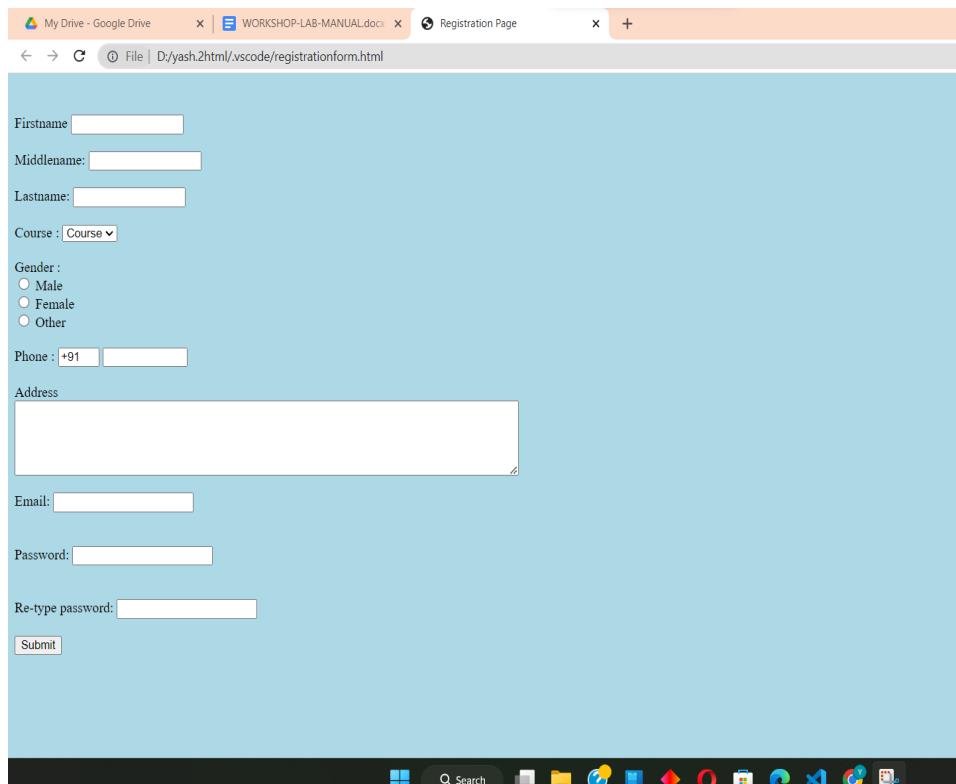
**Aim:** Write a HTML program to develop a static Registration Form.

Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>
        Registration Page
    </title>
</head>
<body bgcolor="Lightblue">
    <br>
    <br>
    <form>
        <label> First Name </label>
        <input type="text" name="firstname" size="15" /> <br> <br>
        <label> Middle Name: </label>
        <input type="text" name="middlename" size="15" /> <br> <br>
        <label> Lastname: </label>
        <input type="text" name="lastname" size="15" /> <br> <br>
        <label> Course : </label>
        <select>
            <option value="Course">Course</option>
            <option value="BCA">BCA</option>
            <option value="BBA">BBA</option>
            <option value="B.Tech">B.Tech</option>
            <option value="MBA">MBA</option>
            <option value="MCA">MCA</option>
            <option value="M.Tech">M.Tech</option>
        </select>
        <br>
        <br>
        <label> Gender : </label><br>
        <input type="radio" name="male" /> Male <br>
        <input type="radio" name="female" /> Female <br>
        <input type="radio" name="other" /> Other
        <br>
        <br>
        <label> Phone : </label>
```

```
<input type="text" name="country code" value="+91" size="2" />
<input type="text" name="phone" size="10" /> <br> <br> Address
<br>
<textarea cols="80" rows="5" value="address"> </textarea>
<br> <br> Email:
<input type="email" id="email" name="email" /> <br>
<br> <br> Password:
<input type="Password" id="pass" name="pass"> <br>
<br> <br> Re-type password:
<input type="Password" id="repass" name="repass"> <br> <br>
<input type="button" value="Submit" />
</form>
</body>
</html>
```

## Output:



The screenshot shows a web browser window with the title "Registration Page". The address bar indicates the file is located at "D:/yash2.html/.vscode/registrationform.html". The page displays a registration form with the following fields:

- Firstname:
- Middlename:
- Lastname:
- Course :
- Gender :  
 Male  
 Female  
 Other
- Phone : +91
- Address:
- Email:
- Password:
- Re-type password:
-

## Practical: - 6

**Aim:** Create your own WebSite using html.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>yash.com</title>
</head>
<body>
    <section id="header">
        <div class="header container">
            <div class="nav-bar">
                <div class="brand">
                    <a href="#">
                        <h1><span>S</span>am <span>m</span>an</h1>
                    </a>
                </div>
                <div class="nav-list">
                    <div class="hamburger">
                        <div class="bar"></div>
                    </div>
                    <ul>
                        <li><a href="#">
data-after="Home">Home</a></li>
                        <li><a href="#">
data-after="Service">Services</a></li>
                        <li><a href="#">
data-after="Projects">Projects</a></li>
                        <li><a href="#">
data-after="About">About</a></li>
                        <li><a href="#">
data-after="Contact">Contact</a></li>
                    </ul>
                </div>
            </div>
        </div>
    </section>
</body>
```

```
</section>
<section id="hero">
  <div class="hero container">
    <div>
      <h1>Welcome to <span></span></h1>
      <h1> New Generation<span></span></h1>
      <h1>Coders <span></span></h1>
      <a href="#projects" type="button" class="cta">CODES</a>
    </div>
  </div>
</section>
<section id="services">
  <div class="services container">
    <div class="service-top">
      <h1 class="section-title">Serv<span>i</span>ces</h1>
      <p></p>
    </div>
    <div class="service-bottom">
      <div class="service-item">
        <div class="icon"></div>
        <h2>Web Design</h2>
        <p> Web design is the creation of websites and
pages to reflect a company's brand and information and ensure a
user-friendly experience. Appearance and design are incorporated as
vital elements whether you're designing a website, mobile
app or maintaining content on a web page.</p>
      </div>
      <div class="service-item">
        <div class="icon"></div>
        <h2>CSS</h2>
        <p>CSS (Cascading Style Sheets) allows you to
create great-looking web pages, but how does it work under the hood?
This article explains what CSS is with a simple syntax example and also
covers some key terms about the language.CSS can
          be used for very basic document text styling.
</p>
        </div>
      <div class="service-item">
        <div class="icon"></div>
```

```
<h2>HTML</h2>


HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as CSS (Cascading Style Sheets) and scripting languages such as JS.



Web browsers receive HTML documents from a web server or from local storage.


</p>
</div>
<div class="service-item">
    <div class="icon"></div>
    <h2>JAVA SCRIPT</h2>
    <p>JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.
</p>
</div>
</div>
</div>
</section>
<section id="projects">
    <div class="projects container">
        <div class="projects-header">
            <h1 class="section-title">Recent

```

```
<div class="project-img">
    
</div>
</div>
<div class="project-item">
    <div class="project-info">
        <h1>Project 2</h1>
        <h2>HTML</h2>
        <p>HTML is an acronym which stands for Hyper  
Text Markup Language which is used for creating web pages and web  
applications. Let's see what is meant by  
Hypertext Markup Language, and Web page. With the help of HTML only,  
we can create static web pages.</p>
    </div>
    <div class="project-img" width="420" height="400" >
        
    </div>
</div>
<div class="project-item">
    <div class="project-info">
        <h1>Project 3</h1>
        <h2>CSS</h2>
        <p>Cascading Style Sheets is a style sheet  
language used for describing the  
presentation of a document written in a  
markup language such as HTML or XML.  
CSS is a cornerstone technology of the  
World Wide Web, alongside HTML and JavaScript.</p>
    </div>
    <div class="project-img">
        
    </div>
</div>
<div class="project-item">
    <div class="project-info">
        <h1>Project 4</h1>
        <h2>JavaScript</h2>
        <p>JavaScript (js) is a light-weight  
object-oriented programming language which is used by several websites
```

for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.</p>

```
</div>
<div class="project-img">
    
        </div>
    </div>
    <div class="project-item">
        <div class="project-info">
            <h1>Project 5</h1>
            <h2>PYTHON</h2>
            <p>Python is a high-level, general-purpose
programming language. Its design philosophy emphasizes code readability
with the use of significant indentation. Python is dynamically typed
and garbage-collected. It supports multiple programming paradigms,
including structured, object-oriented and functional programming. </p>
        </div>
        <div class="project-img">
            
                </div>
            </div>
        </div>
    </div>
</section>
<section id="about">
    <div class="about container">
        <div class="col-left">
            <div class="about-img">
                
            </div>
        </div>
        <div class="col-right">
            <h1 class="section-title">About <span>me</span></h1>
            <h2>SAMMAN TIWARI</h2>
            <p>I am outgoing, dedicated, <BR>and open-minded. I get
across
                <BR>to people and adjust to changes with ease.
                <br> I believe that a person should work on
                <BR> developing their professional skills and<BR>
```

```
        learning new things all the time.  
        <br> Currently, I am a collage student  
        of<BR> <a href="https://bmusurat.ac.in/"  
target="_blank">BHAGWAN MAHAVIR UNIVERSITY</a></p>  
                <a href="#" class="cta">Download Resume</a>  
            </div>  
        </div>  
    </section>  
    <section id="contact">  
        <div class="contact container">  
            <div>  
                <h1 class="section-title">Contact  
<span>info</span></h1>  
            </div>  
            <div class="contact-items">  
                <div class="contact-item">  
                    <div class="icon"></div>  
                    <div class="contact-info">  
                        <h1>Phone</h1>  
                        <h2>+91 9569646803</h2>  
                    </div>  
                </div>  
                <div class="contact-item">  
                    <div class="icon"></div>  
                    <div class="contact-info">  
                        <h1>Email</h1>  
                        <h2>yashtuwari893@gmail.com</h2>  
                    </div>  
                </div>  
                <div class="contact-item">  
                    <div class="icon"></div>  
                    <div class="contact-info">  
                        <h1>Address</h1>  
                        <h2>Surat, Gujrat</h2>  
                    </div>  
                </div>  
            </div>  
        </div>  
    </section>
```

```

<section id="footer">
    <div class="footer container">
        <div class="brand">
            <h1><span>S</span>am <span>m</span>an</h1>
        </div>
        <h2>Your Complete Web Solution</h2>
        <div class="social-icon">
            <div class="social-item">
                <a href="https://www.facebook.com/profile.php?id=100027897226955"></a>
            </div>
            <div class="social-item">
                <a href="https://www.instagram.com/yash.tiwari893/?next=%2F"></a>
            </div>
            <div class="social-item">
                <a href="https://twitter.com/YashTiw36513596"></a>
            </div>
            <p>Copyright © 2023 samman. All rights reserved</p>
        </div>
    </div>
</body>
</html>

```

## C.S.S:

```

@import
'https://fonts.googleapis.com/css?family=Montserrat:300,
400,700&display=swap';
* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
html {
    font-size: 10px;
    font-family: 'Montserrat', sans-serif;

```

```
    scroll-behavior: smooth;
}

a {
    text-decoration: none;
}

.container {
    min-height: 100vh;
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}

img {
    height: 100%;
    width: 100%;
    object-fit: cover;
}

p {
    color: black;
    font-size: 1.4rem;
    margin-top: 5px;
    line-height: 2.5rem;
    font-weight: 300;
    letter-spacing: 0.05rem;
}

.section-title {
    font-size: 4rem;
    font-weight: 300;
    color: black;
    margin-bottom: 10px;
    text-transform: uppercase;
    letter-spacing: 0.2rem;
    text-align: center;
}

.section-title span {
    color: crimson;
}

.cta {
    display: inline-block;
    padding: 10px 30px;
    color: white;
```

```
background-color: transparent;
border: 2px solid crimson;
font-size: 2rem;
text-transform: uppercase;
letter-spacing: 0.1rem;
margin-top: 30px;
transition: 0.3s ease;
transition-property: background-color, color;
}

.cta:hover {
    color: white;
    background-color: crimson;
}

.brand h1 {
    font-size: 3rem;
    text-transform: uppercase;
    color: white;
}

.brand h1 span {
    color: crimson;
}

#header {
    position: fixed;
    z-index: 1000;
    left: 0;
    top: 0;
    width: 100vw;
    height: auto;
}

#header .header {
    min-height: 8vh;
    background-color: rgba(31, 30, 30, 0.24);
    transition: 0.3s ease background-color;
}

#header .nav-bar {
    display: flex;
    align-items: center;
    justify-content: space-between;
    width: 100%;
    height: 100%;
    max-width: 1300px;
    padding: 0 10px;
```

```
}
```

```
#header .nav-list ul {
```

```
    list-style: none;
```

```
    position: absolute;
```

```
    background-color: rgb(31, 30, 30);
```

```
    width: 100vw;
```

```
    height: 100vh;
```

```
    left: 100%;
```

```
    top: 0;
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
    z-index: 1;
```

```
    overflow-x: hidden;
```

```
    transition: 0.5s ease left;
```

```
}
```

```
#header .nav-list ul.active {
```

```
    left: 0%;
```

```
}
```

```
#header .nav-list ul a {
```

```
    font-size: 2.5rem;
```

```
    font-weight: 500;
```

```
    letter-spacing: 0.2rem;
```

```
    text-decoration: none;
```

```
    color: white;
```

```
    text-transform: uppercase;
```

```
    padding: 20px;
```

```
    display: block;
```

```
}
```

```
#header .nav-list ul a::after {
```

```
    content: attr(data-after);
```

```
    position: absolute;
```

```
    top: 50%;
```

```
    left: 50%;
```

```
    transform: translate(-50%, -50%) scale(0);
```

```
    color: rgba(240, 248, 255, 0.021);
```

```
    font-size: 13rem;
```

```
    letter-spacing: 50px;
```

```
    z-index: -1;
```

```
    transition: 0.3s ease letter-spacing;
```

```
}
```

```
#header .nav-list ul li:hover a::after {
    transform: translate(-50%, -50%) scale(1);
    letter-spacing: initial;
}

#header .nav-list ul li:hover a {
    color: crimson;
}

#header .hamburger {
    height: 60px;
    width: 60px;
    display: inline-block;
    border: 3px solid white;
    border-radius: 50%;
    position: relative;
    display: flex;
    align-items: center;
    justify-content: center;
    z-index: 100;
    cursor: pointer;
    transform: scale(0.8);
    margin-right: 20px;
}

#header .hamburger:after {
    position: absolute;
    content: '';
    height: 100%;
    width: 100%;
    border-radius: 50%;
    border: 3px solid white;
    animation: hamburger_puls 1s ease infinite;
}

#header .hamburger .bar {
    height: 2px;
    width: 30px;
    position: relative;
    background-color: white;
    z-index: -1;
}

#header .hamburger .bar::after,
#header .hamburger .bar::before {
    content: '';
    position: absolute;
```

```
height: 100%;  
width: 100%;  
left: 0;  
background-color: white;  
transition: 0.3s ease;  
transition-property: top, bottom;  
}  
  
#header .hamburger .bar::after {  
    top: 8px;  
}  
  
#header .hamburger .bar::before {  
    bottom: 8px;  
}  
  
#header .hamburger.active .bar::before {  
    bottom: 0;  
}  
  
#header .hamburger.active .bar::after {  
    top: 0;  
}  
  
#hero {  
    background-image: url("./img/hero-bg.png");  
    background-size: cover;  
    background-position: top center;  
    position: relative;  
    z-index: 1;  
}  
  
#hero::after {  
    content: '';  
    position: absolute;  
    left: 0;  
    top: 0;  
    height: 100%;  
    width: 100%;  
    background-color: black;  
    opacity: 0.7;  
    z-index: -1;  
}  
  
#hero .hero {  
    max-width: 1200px;  
    margin: 0 auto;  
    padding: 0 50px;  
    justify-content: flex-start;
```

```
}

#hero h1 {
    display: block;
    width: fit-content;
    font-size: 4rem;
    position: relative;
    color: transparent;
    animation: text_reveal 0.5s ease forwards;
    animation-delay: 1s;
}

#hero h1:nth-child(1) {
    animation-delay: 1s;
}

#hero h1:nth-child(2) {
    animation-delay: 2s;
}

#hero h1:nth-child(3) {
    animation: text_reveal_name 0.5s ease forwards;
    animation-delay: 3s;
}

#hero h1 span {
    position: absolute;
    top: 0;
    left: 0;
    height: 100%;
    width: 0;
    background-color: crimson;
    animation: text_reveal_box 1s ease;
    animation-delay: 0.5s;
}

#hero h1:nth-child(1) span {
    animation-delay: 0.5s;
}

#hero h1:nth-child(2) span {
    animation-delay: 1.5s;
}

#hero h1:nth-child(3) span {
    animation-delay: 2.5s;
}

#services .services {
    flex-direction: column;
    text-align: center;
```

```
max-width: 1500px;
margin: 0 auto;
padding: 100px 0;
}

#services .service-top {
    max-width: 500px;
    margin: 0 auto;
}

#services .service-bottom {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-wrap: wrap;
    margin-top: 50px;
}

#services .service-item {
    flex-basis: 80%;
    display: flex;
    align-items: flex-start;
    justify-content: center;
    flex-direction: column;
    padding: 30px;
    border-radius: 10px;
    background-image: url(./img/img-1.png);
    background-size: cover;
    margin: 10px 5%;
    position: relative;
    z-index: 1;
    overflow: hidden;
}

#services .service-item::after {
    content: '';
    position: absolute;
    left: 0;
    top: 0;
    height: 100%;
    width: 100%;
    background-image: linear-gradient(60deg, #29323c 0%, #485563 100%);
    opacity: 0.9;
    z-index: -1;
}

#services .service-bottom .icon {
```

```
height: 80px;
width: 80px;
margin-bottom: 20px;
}

#services .service-item h2 {
    font-size: 2rem;
    color: white;
    margin-bottom: 10px;
    text-transform: uppercase;
}

#services .service-item p {
    color: white;
    text-align: left;
}

#projects .projects {
    flex-direction: column;
    max-width: 1200px;
    margin: 0 auto;
    padding: 100px 0;
}

#projects .projects-header h1 {
    margin-bottom: 50px;
}

#projects .all-projects {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}

#projects .project-item {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    width: 80%;
    margin: 20px auto;
    overflow: hidden;
    border-radius: 10px;
}

#projects .project-info {
    padding: 30px;
    flex-basis: 50%;
```

```
height: 100%;  
display: flex;  
align-items: flex-start;  
justify-content: center;  
flex-direction: column;  
background-image: linear-gradient(60deg, #29323c 0%, #485563 100%);  
color: white;  
}  
  
#projects .project-info h1 {  
    font-size: 4rem;  
    font-weight: 500;  
}  
  
#projects .project-info h2 {  
    font-size: 1.8rem;  
    font-weight: 500;  
    margin-top: 10px;  
}  
  
#projects .project-info p {  
    color: white;  
}  
  
#projects .project-img {  
    flex-basis: 50%;  
    height: 300px;  
    overflow: hidden;  
    position: relative;  
}  
  
#projects .project-img:after {  
    content: '';  
    position: absolute;  
    left: 0;  
    top: 0;  
    height: 100%;  
    width: 100%;  
    background-image: linear-gradient(60deg, #29323c 0%, #485563 100%);  
    opacity: 0.7;  
}  
  
#projects .project-img img {  
    transition: 0.3s ease transform;  
}  
  
#projects .project-item:hover .project-img img {  
    transform: scale(1.1);  
}
```

```
#about .about {
    flex-direction: column-reverse;
    text-align: center;
    max-width: 1200px;
    margin: 0 auto;
    padding: 100px 20px;
}

#about .col-left {
    width: 250px;
    height: 360px;
}

#about .col-right {
    width: 100%;
}

#about .col-right h2 {
    font-size: 1.8rem;
    font-weight: 500;
    letter-spacing: 0.2rem;
    margin-bottom: 10px;
}

#about .col-right p {
    margin-bottom: 20px;
}

#about .col-right .cta {
    color: black;
    margin-bottom: 50px;
    padding: 10px 20px;
    font-size: 2rem;
}

#about .col-left .about-img {
    height: 100%;
    width: 100%;
    position: relative;
    border: 10px solid white;
}

#about .col-left .about-img::after {
    content: '';
    position: absolute;
    left: -33px;
    top: 19px;
    height: 98%;
    width: 98%;
```

```
border: 7px solid crimson;
z-index: -1;
}

#contact .contact {
    flex-direction: column;
    max-width: 1200px;
    margin: 0 auto;
    width: 90%;
}

#contact .contact-items {
    width: 100%;
}

#contact .contact-item {
    width: 80%;
    padding: 20px;
    text-align: center;
    border-radius: 10px;
    padding: 30px;
    margin: 30px;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    box-shadow: 0px 0px 18px 0 #0000002c;
    transition: 0.3s ease box-shadow;
}

#contact .contact-item:hover {
    box-shadow: 0px 0px 5px 0 #0000002c;
}

#contact .icon {
    width: 70px;
    margin: 0 auto;
    margin-bottom: 10px;
}

#contact .contact-info h1 {
    font-size: 2.5rem;
    font-weight: 500;
    margin-bottom: 5px;
}

#contact .contact-info h2 {
    font-size: 1.3rem;
    line-height: 2rem;
```

```
    font-weight: 500;
}

#footer {
    background-image: linear-gradient(60deg, #29323c 0%, #485563 100%);
}

#footer .footer {
    min-height: 200px;
    flex-direction: column;
    padding-top: 50px;
    padding-bottom: 10px;
}

#footer h2 {
    color: white;
    font-weight: 500;
    font-size: 1.8rem;
    letter-spacing: 0.1rem;
    margin-top: 10px;
    margin-bottom: 10px;
}

#footer .social-icon {
    display: flex;
    margin-bottom: 30px;
}

#footer .social-item {
    height: 50px;
    width: 50px;
    margin: 0 5px;
}

#footer .social-item img {
    filter: grayscale(1);
    transition: 0.3s ease filter;
}

#footer .social-item:hover img {
    filter: grayscale(0);
}

#footer p {
    color: white;
    font-size: 1.3rem;
}

@keyframes hamburger_puls {
    0% {
        opacity: 1;
    }
    50% {
        opacity: 0;
    }
    100% {
        opacity: 1;
    }
}
```

```
        transform: scale(1);
    }
    100% {
        opacity: 0;
        transform: scale(1.4);
    }
}

@keyframes text_reveal_box {
    50% {
        width: 100%;
        left: 0;
    }
    100% {
        width: 0;
        left: 100%;
    }
}

@keyframes text_reveal {
    100% {
        color: white;
    }
}

@keyframes text_reveal_name {
    100% {
        color: crimson;
        font-weight: 500;
    }
}

@media only screen and (min-width: 768px) {
    .cta {
        font-size: 2.5rem;
        padding: 20px 60px;
    }
    h1.section-title {
        font-size: 6rem;
    }
    #hero h1 {
        font-size: 7rem;
    }
    #services .service-bottom .service-item {
        flex-basis: 45%;
        margin: 2.5%;
    }
}
```

```
}

#projects .project-item {
    flex-direction: row;
}

#projects .project-item:nth-child(even) {
    flex-direction: row-reverse;
}

#projects .project-item {
    height: 400px;
    margin: 0;
    width: 100%;
    border-radius: 0;
}

#projects .all-projects .project-info {
    height: 100%;
}

#projects .all-projects .project-img {
    height: 100%;
}

#about .about {
    flex-direction: row;
}

#about .col-left {
    width: 600px;
    height: 400px;
    padding-left: 60px;
}

#about .about .col-left .about-img::after {
    left: -45px;
    top: 34px;
    height: 98%;
    width: 98%;
    border: 10px solid crimson;
}

#about .col-right {
    text-align: left;
    padding: 30px;
}

#about .col-right h1 {
    text-align: left;
}

#contact .contact {
```

```
flex-direction: column;
padding: 100px 0;
align-items: center;
justify-content: center;
min-width: 20vh;
}

#contact .contact-items {
    width: 100%;
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
    margin: 0;
}

#contact .contact-item {
    width: 30%;
    margin: 0;
    flex-direction: row;
}

#contact .contact-item .icon {
    height: 100px;
    width: 100px;
}

#contact .contact-item .icon img {
    object-fit: contain;
}

#contact .contact-item .contact-info {
    width: 100%;
    text-align: left;
    padding-left: 20px;
}

}

@media only screen and (min-width: 1200px)
{
    #header .hamburger {
        display: none;
    }

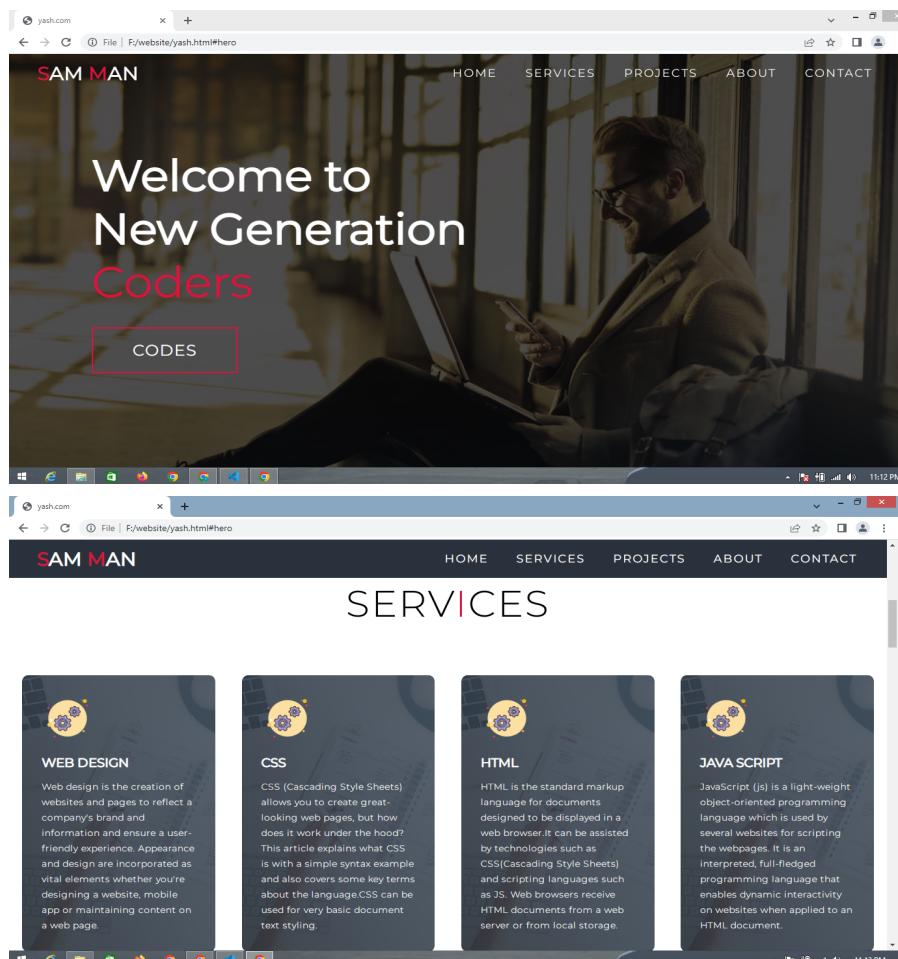
    #header .nav-list ul {
        position: initial;
        display: block;
        height: auto;
        width: fit-content;
        background-color: transparent;
    }
}
```

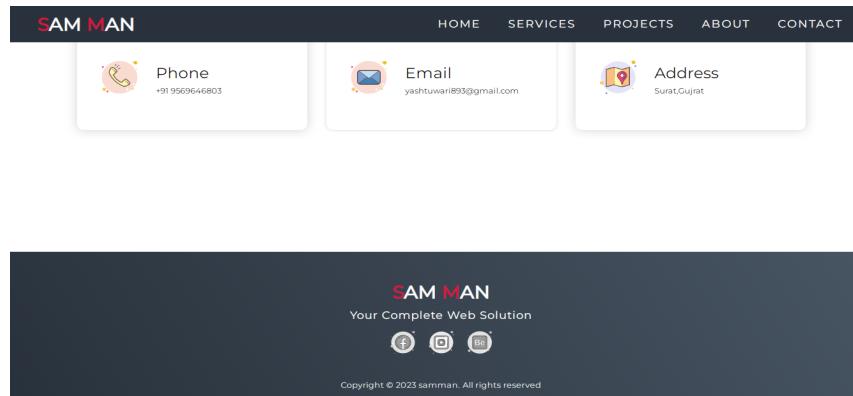
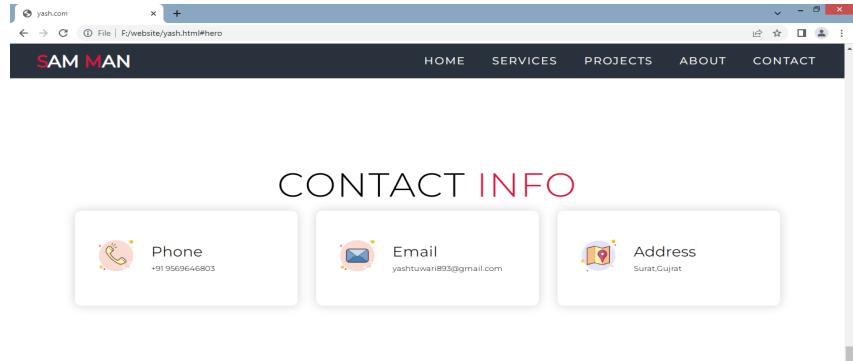
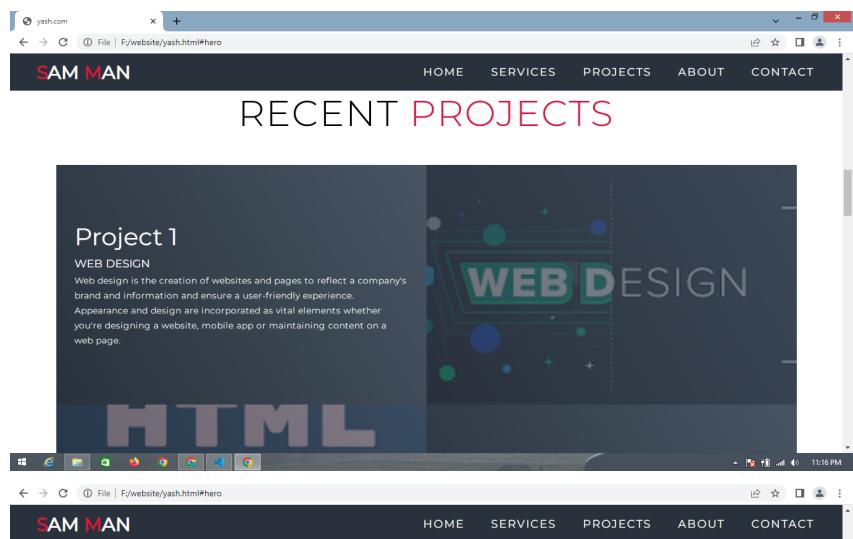
```

}
#header .nav-list ul li {
    display: inline-block;
}
#header .nav-list ul li a {
    font-size: 1.8rem;
}
#header .nav-list ul a::after {
    display: none;
}
#services .service-bottom .service-item {
    flex-basis: 22%;
    margin: 1.5%;
}
}

```

## Output:





## Practical: - 7

### Aim: How to install Windows operating system

#### ► Step 1 - Format the drive and set the primary partition as active

1. Connect the USB flash drive to your technician PC.
2. Open Disk Management: Right-click on Start and choose Disk Management.
3. Format the partition: Right-click the USB drive partition and choose Format. Select the FAT32 file system to be able to boot either BIOS-based or UEFI-based PCs.
4. Set the partition as active: Right-click the USB drive partition and click Mark Partition as Active.

#### ► Step 2 - Copy Windows Setup to the USB flash drive

1. Use File Explorer to copy and paste the entire contents of the Windows product DVD or ISO to the USB flash drive.
2. Optional: add an unattend file to automate the installation process. For more information, see Automate Windows Setup.

#### ► Step 3 - Install Windows to the new PC

1. Connect the USB flash drive to the new PC.
2. Turn on the PC and press the key that opens the boot-device selection menu for the computer, such as the Esc/F10/F12 keys. Select the option that boots the PC from the USB flash drive.  
Windows Setup starts. Follow the instructions to install Windows.
3. Remove the USB flash drive.

#### ► If your Windows image is larger than 4GB

Windows USB install drives are formatted as FAT32, which has a 4GB file size limit. If your image is larger than the file size limit:

1. Copy everything except the Windows image file (sources\install.wim) to the USB drive (either drag and drop, or use this command, where D: is the mounted ISO and E: is the USB flash drive.)

**> Command**

```
robocopy D: E: /s /max:3800000000
```

2. Split the Windows image file into smaller files, and put the smaller files onto the USB drive:

**> Command**

```
Dism /Split-Image /ImageFile:D:\sources\install.wim /SWMFile:E:\sources\install.swm /FileSize:3800
```

## Practical: - 8

### Aim: Study of Basic commands of Linux/UNIX

#### ➤ Basic Linux Commands

**1. pwd** " When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the `pwd` command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by forward slash ( / ). The user directory is usually something like "/home/username".

```
nayso@Alok-Aspire:~$ pwd
/home/nayso
```

**2. ls** " Use the "ls" command to know what files are in the directory you are in. You can see all the hidden files by using the command `ls -a` .

```
nayso@Alok-Aspire:~$ ls
Desktop      itsuserguide.desktop  reset-settings   VCD_Copy
Documents    Music                School_Resources Videos
Downloads    Pictures             Students_Works_10
examples.desktop Public              Templates
GplatesProject Qgis Projects       TuxPaint-Pictures
```

**3. cd** " Use the "cd" command to go to the directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in `cd Downloads` . Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is. But there is a problem with these commands. Imagine you have a folder named `Raspberry Pi` . In this case, when you type in `cd Raspberry Pi` , the shell will take the second argument of the command as a different one, so you will get an error saying that the directory does not exist. Here, you can use backward slash. That is, you can use `cd Raspberry\Pi` in this case. Spaces are denoted like this: If you just type `cd` and press enter, it takes you to the home directory. To go back from folder to folder before that, you can type `cd ..` . The two dots represent the back.

```
nayso@Alok-Aspire:~$ cd Downloads
nayso@Alok-Aspire:~/Downloads$ cd
nayso@Alok-Aspire:~$ cd Raspberry\ Pi
nayso@Alok-Aspire:~/Raspberry Pi$ cd ..
nayso@Alok-Aspire:~$ █
```

**4. mkdir & rmdir** " Use the `mkdir` command when you need to create a folder or directory. For example, if you want to make a directory called `DIY` , then you can type `mkdir DIY` . Remember, as told before, if you want to create a directory named `DIY Hacking` , then you can type `mkdir DIY\ Hacking` . Use `rmdir` to delete the directory. But `rmdir` can only be used to delete an empty directory. To delete a directory containing files, use `rm`.

```
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ mkdir DIY
nayso@Alok-Aspire:~/Desktop$ ls
DIY
nayso@Alok-Aspire:~/Desktop$ rmdir DIY
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ █
```

**5. rm** - Use the rm command to delete files and directories. Use "rm -r" to delete just the directory. It deletes both the folder and the files it contains when using only the rm command.

```
nayso@Alok-Aspire:~/Desktop$ ls
newer.py New Folder
nayso@Alok-Aspire:~/Desktop$ rm newer.py
nayso@Alok-Aspire:~/Desktop$ ls
New Folder
nayso@Alok-Aspire:~/Desktop$ rm -r New\ Folder
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ █
```

**6. touch** ” The touch command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, `touch new.txt`

```
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ touch new.txt
nayso@Alok-Aspire:~/Desktop$ ls
new.txt
```

**7. man & --help** ” To know more about command and how to use it, use the man command. It shows the manual pages of the command. For example, `man cd` shows the manual pages of the cd command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., `cd --help`).

TOUCH(1)	User Commands	TOUCH(1)
<b>NAME</b>	<code>touch</code> - change file timestamps	
<b>SYNOPSIS</b>	<code>touch [OPTION]... FILE...</code>	
<b>DESCRIPTION</b>	Update the access and modification times of each FILE to the current time.	
A FILE argument that does not exist is created empty, unless <code>-c</code> or <code>-h</code> is supplied.		
A FILE argument string of <code>-</code> is handled specially and causes <code>touch</code> to change the times of the file associated with standard output.		
Mandatory arguments to long options are mandatory for short options too.		
<code>-a</code> change only the access time		

**Manual page touch(1) line 1 (press h for help or q to quit)**

**8. cp** ” Use the cp command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.

```
nayso@Alok-Aspire:~/Desktop$ ls /home/nayso/Music/
nayso@Alok-Aspire:~/Desktop$ cp new.txt /home/nayso/Music/
nayso@Alok-Aspire:~/Desktop$ ls /home/nayso/Music/
new.txt
```

**9. mv** ” Use the mv command to move files through the command line. We can also use the mv command to rename files. For example, if we want to rename the file `text` to `new` , we can use `mv text new` . It takes the two arguments, just like the cp command.

```
nayso@Alok-Aspire:~/Desktop$ ls
new.txt
nayso@Alok-Aspire:~/Desktop$ mv new.txt newer.txt
nayso@Alok-Aspire:~/Desktop$ ls
newer.txt
```

**10. locate** ” The locate command is used to locate files in the Linux system, just like the search command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file. Using the `-i` argument with the command helps to ignore the case (it doesn't matter if it is uppercase or lowercase). So, if you want a file that has the word `hello` , it gives the list of all the files in your Linux system containing the word "hello" when you type in `locate -i hello` . If you remember two words, you can separate them using an asterisk (\*). For example, to locate file containing the words "hello" and "this", you can use the command `locate -i *hello*this` .

```
nayso@Alok-Aspire:~$ locate newer.txt
/home/nayso/Desktop/newer.txt
nayso@Alok-Aspire:~$ locate *DIY*Hacking*
/home/nayso/DIY Hacking
```

## ➤ Intermediate Commands

**1. echo** ” The "echo" command helps us move some data, usually text into file. For example, if you want to create a new text file or add to an already made text file, you just need to type in, `echo hello, my name is alok >> new.txt` . You do not need to separate the spaces by using the backward slash here, because we put in two triangular brackets when we finish what we need to write.

**2. cat** ” Use the cat command to display the contents of the file. It is usually used to easily view programs.

```
nayso@Alok-Aspire:~/Desktop$ echo hello, my name is alok >> new.txt
nayso@Alok-Aspire:~/Desktop$ cat new.txt
hello, my name is alok
nayso@Alok-Aspire:~/Desktop$ echo this is another line >> new.txt
nayso@Alok-Aspire:~/Desktop$ cat new.txt
hello, my name is alok
this is another line
```

**3. nano, vi, jed** ” nano and vi are already installed as text editors in the Linux command line. The nano command is a good text editor that denotes keywords with color and can recognize most languages. And vi is simpler than nano. You can create new files or modify files using this editor. For example, if you need to make a new file named "check.txt", you can create it by using the command `nano check.txt` . You can save your files after editing by using the sequence Ctrl+X, then Y (or N for no). In my experience, using nano for HTML editing

doesn't seem as good, because of its color, so I recommend jed text editor. We will come to install packages soon.

```
GNU nano 2.2.6          File: check.txt          Modified

This is a file named check.txt edited in Nano Text Editor!!

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes      ^C Cancel
```

**4. sudo** ” widely used command in the Linux command line, sudo stands for "SuperUser Do". So, if you want any command to be done with administrative or root privileges, you can use the sudo command. For example, if you want to edit files like viz. alsa-base.conf, which needs root permissions, you can use the command “ sudo nano alsa-base.conf. You can enter the root command line using the command `sudo bash`, then type in your user password. You can also use the command `su` to do this, but you need to set the root password before that. For that, you can use the command `sudo passwd` (not misspelled, it is `passwd`). Then type in the new root password.

```
nayso@Alok-Aspire:~/Desktop$ sudo passwd
[sudo] password for nayso:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
nayso@Alok-Aspire:~/Desktop$ su
Password:
root@Alok-Aspire:/home/nayso/Desktop#
```

**5. df** ” Use the df command to see the available disk space in each of the partitions in your system. You can just type in df in the command line and you can see each mounted partition and their used/available space in % and in KBs. If you want it shown in megabytes, you can use the command `df -m` .

```
root@Alok-Aspire:/home/nayso/Desktop# df -m
Filesystem 1M-blocks Used Available Use% Mounted on
udev         940     1      940   1% /dev
tmpfs        191     2      189   1% /run
/dev/sda5  96398 23466    68013  26% /
none         1     0       1   0% /sys/fs/cgroup
none         5     0       5   0% /run/lock
none        951     1      950   1% /run/shm
none        100     1      100   1% /run/user
```

**6. du** ” Use du to know the disk usage of files in your system. If you want to know the disk usage for a particular folder or file in Linux, you can type in the command df and the name of

the folder or file. For example, if you want to know the disk space used by the documents folder in Linux, you can use the command `du Documents`. You can also use the command `ls -lah` to view the file sizes of all the files in the folder.

```
nayso@Alok-Aspire:~$ du Documents
516    Documents/DIYHacking
548    Documents
```

**7. tar** ” Use tar to work with tarballs (or files compressed in tarball archive) in the Linux command line. It has a long list of uses. It can be used to compress and uncompress different types of tar archives like .tar, .tar.gz, .tar.bz2,etc. It works on the basis of the arguments given to it. For example, "tar -cvf" for creating .tar archive, -xvf to untar tar archive, -tvf to list the contents of the archive, etc. Since it is a wide topic, here are some examples of tar commands.

**8. zip, unzip** Use zip to compress files into zip archive, and unzip to extract files from zip archive.

**9. uname** ” Use uname to show the information about the system your Linux distro is running. Using the command `uname -a` prints most of the information about the system. This prints the kernel release date, version, processor type, etc.

```
nayso@Alok-Aspire:~$ uname -a
Linux Alok-Aspire 4.4.0-22-generic #40~14.04.1-Ubuntu SMP Fri May 13 17:27:18 UT
C 2016 i686 i686 i686 GNU/Linux
```

**10. apt-get** ” Use apt to work with packages in the Linux command line. Use apt-get to install packages. This requires root privileges, so use the sudo command with it. For example, if you want to install the text editor jed (as I mentioned earlier), we can type in the command `sudo apt-get install jed`. Similarly, any packages can be installed like this. It is good to update your repository each time you try to install a new package. You can do that by typing `sudo apt-get update`. You can upgrade the system by typing `sudo apt-get upgrade`. We can also upgrade the distro by typing `sudo apt-get dist-upgrade`. The command `apt-cache search` is used to search for packages. If you want to search for one, you can type in `apt-cache search jed` (this doesn't require root).

```
nayso@Alok-Aspire:~$ sudo apt-get install jed
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  jed-common libslang2-modules slsh
Suggested packages:
  gpm
The following NEW packages will be installed:
  jed jed-common libslang2-modules slsh
0 upgraded, 4 newly installed, 0 to remove and 419 not upgraded.
Need to get 810 kB of archives.
After this operation, 2,992 kB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

**11. chmod** ” Use chmod to make the file executable and to change the permissions granted to it in Linux. Imagine you have a python code named numbers.py in your computer. You'll need to run `python numbers.py` every time you need to run it. Instead of that, when you

make it executable, you'll just need to run `numbers.py` in the terminal to run the file. To make the file executable, you can use the command `chmod +x numbers.py` in this case. You can use `chmod 755 numbers.py` to give it root permissions or `sudo chmod +x numbers.py` for root executable. Here is some more information about the `chmod` command.

```
nayso@Alok-Aspire:~/Desktop$ ls
numbers.py
nayso@Alok-Aspire:~/Desktop$ chmod +x numbers.py
nayso@Alok-Aspire:~/Desktop$ ls
numbers.py
```

**12. hostname** " Use `hostname` to know your name in your host or network. Basically, it displays your hostname and IP address. Just typing `hostname` gives the output. Typing in `hostname -I` gives you your IP address in your network.

```
nayso@Alok-Aspire:~/Desktop$ hostname
Alok-Aspire
nayso@Alok-Aspire:~/Desktop$ hostname -I
192.168.1.36
```

**13. ping** " Use `ping` to check your connection to the server. Wikipedia says, "Ping is computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network". Simply, when you type in, for example, `ping google.com`, it checks if it can connect to the server and come back. It measures this round-trip time and gives you the details about it. The use of this command for simple users like us is to check your internet connection. If it pings the Google server (in this case), you can confirm that your internet connection is active!

```
nayso@Alok-Aspire:~/Desktop$ ping google.com
PING google.com (172.217.26.206) 56(84) bytes of data.
64 bytes from google.com (172.217.26.206): icmp_seq=1 ttl=56 time=51.2 ms
64 bytes from google.com (172.217.26.206): icmp_seq=2 ttl=56 time=47.9 ms
64 bytes from google.com (172.217.26.206): icmp_seq=3 ttl=56 time=48.9 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 47.959/49.388/51.299/1.417 ms
```

## ➤ Basic UNIX Commands to Work on Mac Terminal

### 1. pwd

In the console, you are always working in a directory, or folder, on your computer. We call this your working directory. You can see where you are using this command



### 2. ls

This command will show you all the files in that directory. and you can also view all the hidden files by this command "ls -a"

```
Hari:~ admin$ pwd
/Users/admin
Hari:~ admin$ ls
Applications      Library      Pictures
Desktop          MacKeeper Backups  Public
Documents        Movies       VirtualBox VMs
Downloads        Music
```

### 3. cd

This command will take you to directory ,suppose right now we are in the home folder and you want to go to Desktop, so just type

" cd Desktop/ "

```
Hari:~ admin$ cd Desktop/
Hari:Desktop admin$
```

here we use "\ " for space , suppose you want to open file named "Macbook" from Desktop we should type

"cd Desktop/Mac\ book/"

```
Hari:~ admin$ cd Desktop/Mac\ book/
Hari:Mac book admin$
```

Now you are in the folder named Macbook , but you want to come back to the desktop , here we use " cd .. " , This command will bring you back one directory and typing " cd " will take you back to Desktop directly

```
Hari:~ admin$ cd Desktop/Mac\ book/
Hari:Mac book admin$ cd ..
Hari:Desktop admin$
```

### 4. mkdir

This command is used for making a directory . to make directory you should type the directory name also for example "mkdir Mac", this will create new directory with name Mac

```
Hari:Desktop admin$ ls
my
Hari:Desktop admin$ mkdir Mac
Hari:Desktop admin$ ls
Mac   my
Hari:Desktop admin$
```

## 5. rmdir

This command is used to remove an empty directory. to remove directory name Mac we use this command "rmdir Mac"

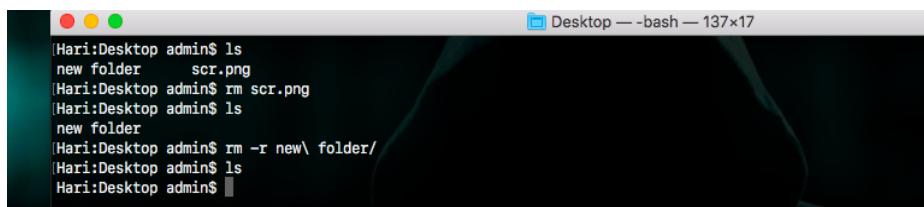


```
Hari:Desktop admin$ ls
Mac      my
Hari:Desktop admin$ rmdir Mac
Hari:Desktop admin$ ls
my
Hari:Desktop admin$
```

## Working With Files

### 1. rm

"rm" command is used to remove the directory and "rm -r" is used to remove the directory with files. If you want to confirm the file to be deleted, use " -i " as in " rm -i scr.png " (works only with files )

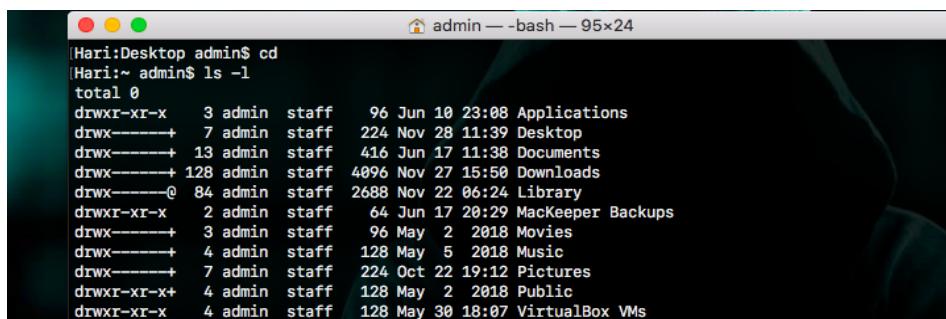


```
Hari:Desktop admin$ ls
new folder      scr.png
Hari:Desktop admin$ rm scr.png
Hari:Desktop admin$ ls
new folder
Hari:Desktop admin$ rm -r new\ folder/
Hari:Desktop admin$ ls
Hari:Desktop admin$
```

### 2. chmod

Changes permissions for access to file. Now let's see how to modify permissions with chmod. For total control over permissions, you can use two Unix commands "- ls" and "chmod -" to display permissions and modify them. Assume you want to find folder TM's current permissions and then change them to 755. This would give you as the owner read, write and execute permissions, and everyone else read and execute permissions.

Type ls "l, and then press enter. The symbolic permissions of the files and folders in your home directory are displayed



```
Hari:Desktop admin$ cd
Hari:~ admin$ ls -l
total 0
drwxr-xr-x  3 admin  staff   96 Jun 10 23:08 Applications
drwxr-----  7 admin  staff  224 Nov 28 11:39 Desktop
drwxr----- 13 admin  staff  416 Jun 17 11:38 Documents
drwxr----- 128 admin staff 4096 Nov 27 15:50 Downloads
drwxr-----@ 84 admin staff 2688 Nov 22 06:24 Library
drwxr-xr-x  2 admin  staff  64 Jun 17 20:29 MacKeeper Backups
drwxr-----  3 admin  staff  96 May  2 2018 Movies
drwxr-----  4 admin  staff 128 May  5 2018 Music
drwxr-----  7 admin  staff 224 Oct 22 19:12 Pictures
drwxr-xr-x+  4 admin  staff 128 May  2 2018 Public
drwxr-xr-x  4 admin  staff 128 May 30 18:07 VirtualBox VMs
```

Type "chmod 755 folder name", and then press enter. This changes the permissions of the folder to rwxr-xr-x .

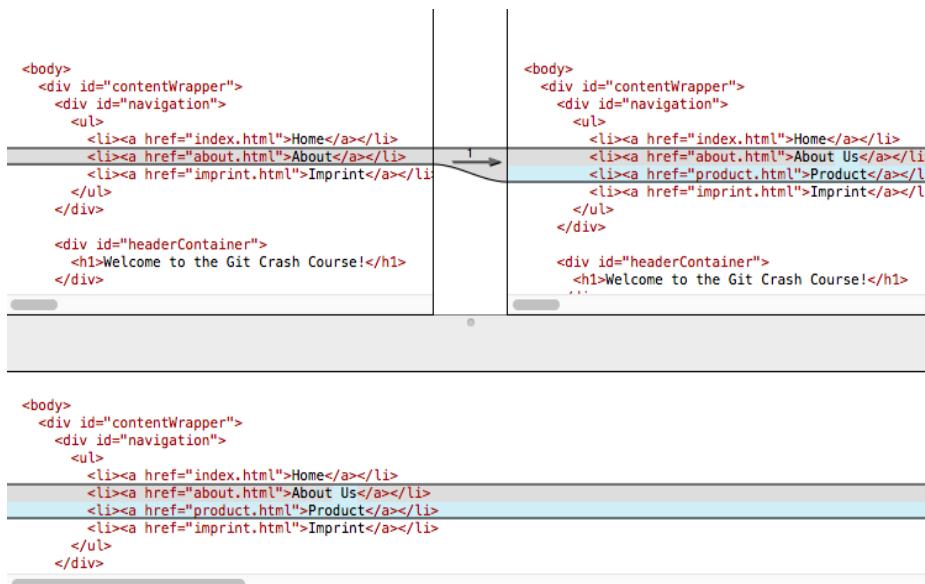
```

[Hari:Desktop admin$ cd
[Hari:~ admin$ ls -l
total 0
drwxr-xr-x  3 admin  staff   96 Jun 10 23:08 Applications
drwxr-xr-x+ 7 admin  staff  224 Nov 28 11:39 Desktop
drwxr-xr-x 13 admin  staff  416 Jun 17 11:38 Documents
drwxr-xr-x+ 128 admin  staff 4096 Nov 27 15:50 Downloads
drwxr-xr-x@ 84 admin  staff 2688 Nov 22 06:24 Library
drwxr-xr-x  2 admin  staff   64 Jun 17 20:29 MacKeeper Backups
drwxr-xr-x+ 3 admin  staff   96 May  2 2018 Movies
drwxr-xr-x+ 4 admin  staff  128 May  5 2018 Music
drwxr-xr-x+ 7 admin  staff  224 Oct 22 19:12 Pictures
drwxr-xr-x+ 4 admin  staff  128 May  2 2018 Public
drwxr-xr-x  4 admin  staff 128 May 30 18:07 VirtualBox VMs
[Hari:~ admin$ chmod 755 Documents/]

```

### 3. diff

Compare two files line by line. It might not be the most elegant tool, but it's definitely a solid one that does the job of comparing & merging text. the command is like "diff [firstfile] [secondfile]"



### 4. more

is command to view the contents of text file one screen.Press the spacebar to see the next page and press Q to quit. The command is like "more [filename]"

```

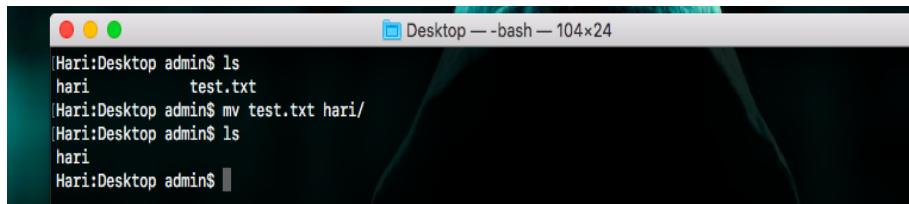
[Hari:Desktop admin$ more test.txt
hello
this is a trial

```

## 5. mv

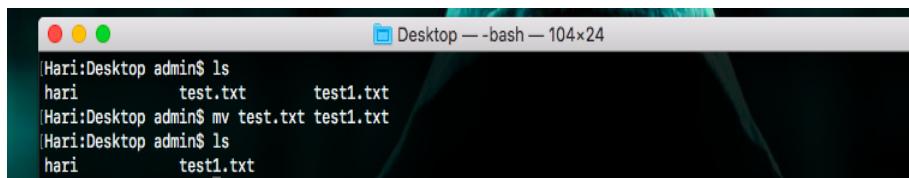
This command can move one file into directory by the command "mv [filename] [folder name]" and it can also move the contents in text file to another text file by the command

"mv [1st filename] [2nd filename]"



```
Hari:Desktop admin$ ls
hari      test.txt
Hari:Desktop admin$ mv test.txt hari/
Hari:Desktop admin$ ls
hari
Hari:Desktop admin$
```

Here the text file has been moved into the folder named hari



```
Hari:Desktop admin$ ls
hari      test.txt      test1.txt
Hari:Desktop admin$ mv test.txt test1.txt
Hari:Desktop admin$ ls
hari      test1.txt
```

Here the content in the first file has been moved to the second file and the content in the second file will be removed

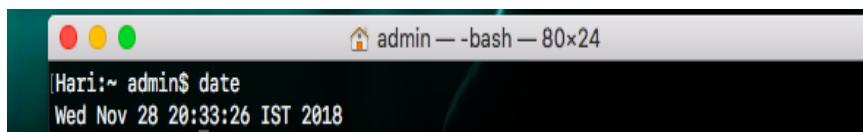
## Miscellaneous Commands

### 1. ctrl+C & ctrl+z

ctrl+c is used to terminate most of the operations and ctrl+z is for force stop. We use this command mostly when we run python , jav or other programs.

### 2. date

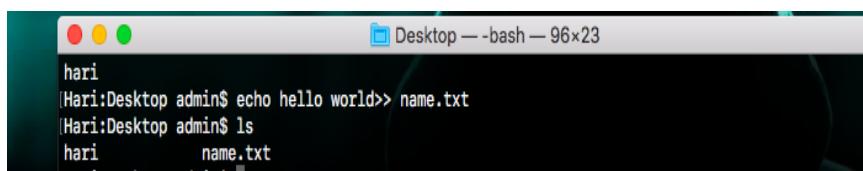
It displays the date and time



```
admin — bash — 80x24
Hari:~ admin$ date
Wed Nov 28 20:33:26 IST 2018
```

### 3. echo

Its command is used to output the text to file. suppose you want to type " hello world " and save it in text file, just type this command " echo hello world >> name.txt "



```
hari
Hari:Desktop admin$ echo hello world>> name.txt
Hari:Desktop admin$ ls
hari      name.txt
Hari:Desktop admin$
```

#### **4. help**

Displays partial list of bash commands. for this simply type "help" in the terminal

#### **5. history**

Displays the last commands you typed. You can redo the command by typing an exclamation point (!) followed immediately (no space) by the number of that command in the history list. To repeat the last command, type " !! ". To repeat the last filename, type " !\* "

#### **6. pico**

simple UNIX text editor. simply type the command "pico" and hit enter, the text editor will appear

```
Hari:~ admin$ ps
 PID TTY      TIME CMD
 47016 ttys000  0:00.02 -bash
```

#### **8. zip and unzip**

These commands are used to compress the zip archive and to extract from the zip archive.

#### **9. sudo**

To execute the sudo command , you must be logged in with an administrator account that has a password.

you can use the sudo command in the Terminal to execute commands as different users, such as the root user. After you enter the command, Terminal asks you to enter your account password. If you forgot your password or your account doesn't have a password, add or change your password in Users & Groups preferences. You can then execute sudo commands in Terminal.

#### **10. ping**

Using this command yo can check the connection to service , for example "ping maker.pro "

```
Hari:Desktop admin$ ping maker.pro
PING maker.pro (104.20.19.96): 56 data bytes
64 bytes from 104.20.19.96: icmp_seq=0 ttl=53 time=81.279 ms
64 bytes from 104.20.19.96: icmp_seq=1 ttl=53 time=81.100 ms
64 bytes from 104.20.19.96: icmp_seq=2 ttl=53 time=81.084 ms
64 bytes from 104.20.19.96: icmp_seq=3 ttl=53 time=81.343 ms
64 bytes from 104.20.19.96: icmp_seq=4 ttl=53 time=81.351 ms
```

#### **3. hostname**

This command will display your hostname. you can also change your hostname by this command "sudo scutil --set HostName [your host name]"

```
Hari:Desktop admin$ sudo scutil --set HostName Harilrishnan.localhost
```

## Practical: - 9

### **Aim: Study of UNIX Shell and Environment Variables.**

What is an environment variable?

Environment variables or ENVs basically define behavior of the environment. They can affect the processes ongoing or the programs that are executed in the environment.

#### **Scope of an environment variable**

Scope of any variable is the region from which it can be accessed or over which it is defined. An environment variable in Linux can have global or local scope.

#### **Global**

globally scoped ENV that is defined in terminal can be accessed from anywhere in that particular environment which exists in the terminal. That means it can be used in all kinds of scripts, programs or processes running in the environment bound by that terminal.

#### **Local**

locally scoped ENV that is defined in the terminal cannot be accessed by any program or process running in the terminal. It can only be accessed by the terminal( in which it was defined) itself.

Variable	Description
<b>USER</b>	The username
<b>HOME</b>	Default path to the user's home directory
<b>EDITOR</b>	Path to the program which edits the content of files
<b>UID</b>	User's unique ID
<b>TERM</b>	Default terminal emulator
<b>SHELL</b>	Shell being used by the user

## Practical: - 10

### **Aim: Study of Advance commands and filters of Linux/UNIX.**

Linux Filter commands accept input data from stdin (standard input) and produce output on stdout (standard output). It transforms plain-text data in a meaningful way and can be used with pipes to perform higher operations.

#### **Linux Filter Commands**

##### **1. cat**

###### **Syntax:**

cat <fileName> | cat or tac | cat or tac | . . .

##### **2. cut**

###### **Syntax:**

The 'cut' command is useful in selecting a specific column of file. After (-d), delimiter (from where you want to separate the columns) comes. Delimiters can be space (' '), hyphen (-), slash (/) or anything else. After (-f), column number is mentioned.

###### **Syntax:**

cut -d(delimiter) -f(columnNumber) <fileName>

##### **3. grep**

The 'grep' command is generally used with pipe ()|.

###### **Syntax:**

command | grep <searchWord>

##### **4. comm**

The 'comm' command compares two files or streams. By default, 'comm' will always display three

columns. First column indicates non-matching items of first file, second column indicates non-matching items of the second file, and the third column indicates matching items of both the files. Both the

files have to be in sorted order for 'comm' command to be executed.

###### **Syntax:**

comm <file1> <file2>

##### **5. sed**

Command 'sed' stands for stream editor. You can use this command to edit streams (files) using regular expressions. But this editing is not permanent. It remains only in display, but in actual, file content remains the same.

###### **Syntax:**

command | sed 's/<oldWord>/<newWord>/'

**6. tee**

The 'tee' command is similar to the 'cat' command with only one difference. It puts stdin on stdout and also put them into file.

Syntax:

cat or tac <fileName> | tee <newFile> | cat or tac |.....

**7. tr**

The command 'tr' stands for 'translate'. It is used to translate, like from lowercase to uppercase and vice versa or newlines into spaces.

Syntax:

command | tr <'old'> <'new'>

**8. uniq**

With the help of uniq command you can form sorted list in which every word will occur only once.

Syntax:

command <fileName> | uniq

**9. wc**

The 'wc' command helps in counting the lines, words and characters in the file.

Syntax:

1. wc <fileName> (Counts words, lines and characters)
2. wc -l <fileName> (Counts only lines)
3. wc -w <fileName> (Counts only words)
4. wc -c <fileName> (Counts only characters)

**10. od**

The 'od' term stands for octal dump. It displays content of file in different human-readable formats like hexadecimal, octal and ASCII characters.

Syntax:

1. od -b <fileName> (display files in octal format)
2. od -t x1 <fileName> (display files in hexadecimal bytes format)
3. od -c <fileName> (display files in ASCII (backslash) character format)