Aim: Applying fundamental design principles to create a visually appealing and user-friendly frontend interface using HTML and CSS.

Theory:
HTML:-

HTML Tags Explanation

1. **`<!DOCTYPE html>`**:
   Declares the document as an HTML5 document. It helps the browser correctly interpret and render the page.
2. **`<html>`**:
   The root element of the HTML document. It contains all other HTML elements.
3. **`<head>`**:
   Contains meta-information about the document, such as the title, character encoding, linked stylesheets, and scripts.
4. **`<meta>`**:
   Provides metadata about the document. Common attributes include:
   - `charset`: Specifies character encoding (e.g., UTF-8 for supporting most languages).
   - `viewport`: Ensures proper scaling on mobile devices.
5. **`<title>`**:
   Specifies the title of the web page, displayed on the browser tab.
6. **`<link>`**:
   Links external resources, such as CSS stylesheets. The `rel` attribute specifies the relationship (e.g., "stylesheet").
7. **`<script>`**:
   Includes or references JavaScript files for functionality. The `src` attribute specifies the script's source.
8. **`<body>`**:
   Contains the main content of the webpage that users interact with, including text, images, and forms.
9. **`<div>`**:
   A generic container element used to group and style content.
10. **`<form>`**:
    Defines a form for user input. The `action` attribute specifies where form data should be sent.
11. **`heading`**:
    Heading tags. <h2> represents a second-level heading, while <h5> is a smaller heading. Headings help organize content hierarchically.
12. **`<label>`**:
    Describes an `<input>` element, improving accessibility by associating a label with an input field.

13. **`<input>`**:
    An element for user input. Common attributes:
    - `type`: Defines the input type (e.g., "text").
    - `id`: Identifies the input field uniquely.
    - `placeholder`: Displays placeholder text inside the field.
14. **`<a>`**:
    Defines a hyperlink. The `href` attribute specifies the link's destination.
15. **`<button>`**:
    Represents a clickable button, typically used to submit forms or perform actions.
16. **`<i>`**:
    Represents an icon or styled text, often used with icon libraries like Font Awesome.

CSS:-

➢ **Styling**:
CSS specifies the look of elements, including:
- Colors (`color`, `background-color`)
- Fonts (`font-family`, `font-size`)
- Spacing (`margin`, `padding`)
- Borders (`border`, `border-radius`)

➢ **Layout Control**:
CSS enables the arrangement of elements on a page using:
- **Box Model**: Margins, borders, padding, and content.
- **Positioning**: (`relative`, `absolute`, `fixed`, `sticky`).
- **Flexbox/Grid**: Modern techniques for flexible and responsive designs.

➢ **Responsive Design**:
Media queries and flexible units (`%`, `em`, `rem`) allow designs to adapt to different screen sizes.

➢ **Selectors**:
CSS applies styles using various selectors:
- `*`: Universal selector.
- `.class`: Targets elements with a specific class.
- `#id`: Targets a unique element by ID.
- `element`: Targets HTML tags directly (e.g., `h1`, `div`).

➢ **Pseudo-Classes and Pseudo-Elements**:
Add styles based on the element's state or part of the content:
- Pseudo-class: `:hover`, `:focus`.
- Pseudo-element: `::before`, `::after`.

➢ **Animation and Effects**:
CSS provides transitions, animations, and hover effects (`transition`, `keyframes`) to enhance interactivity

index.html

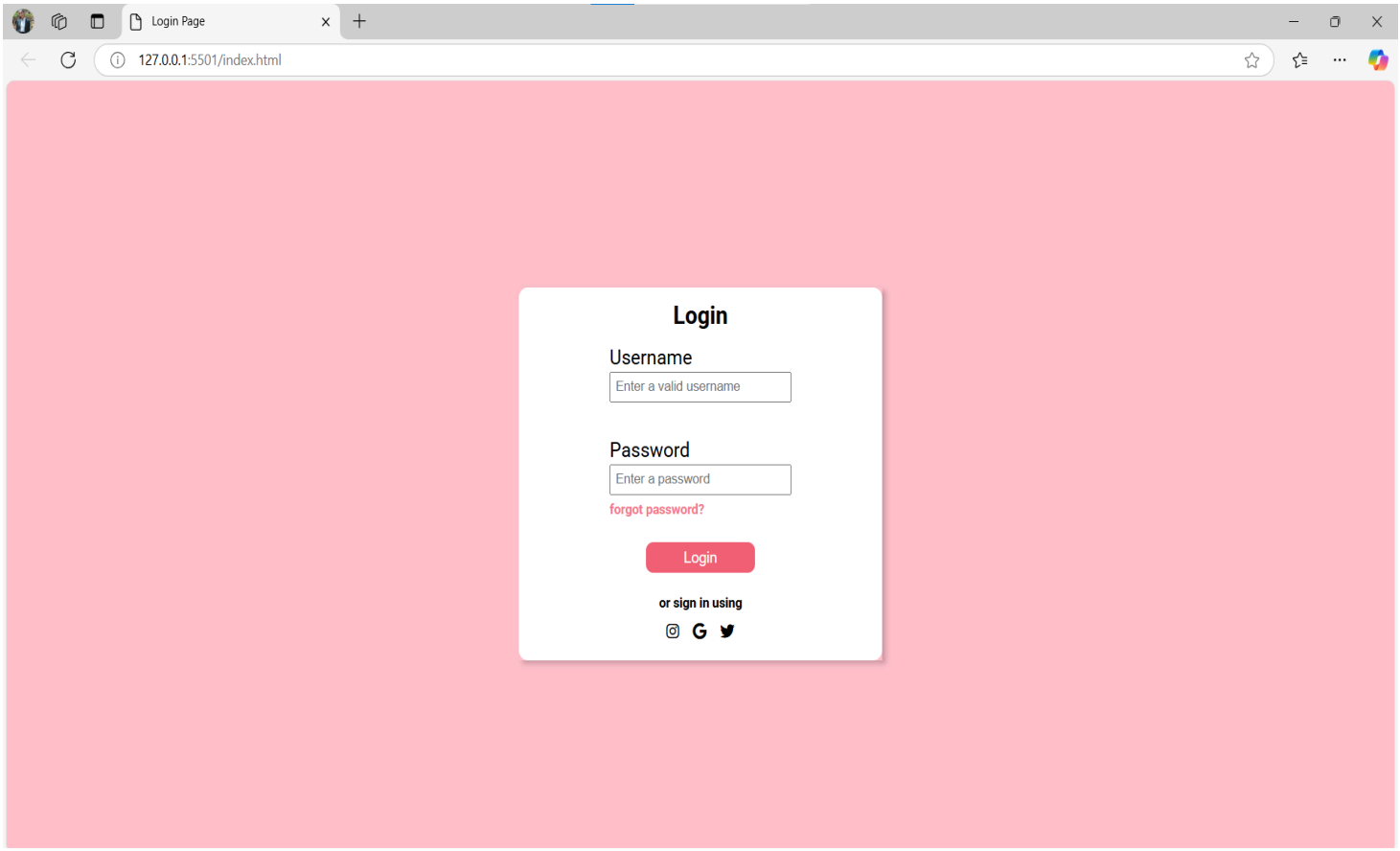
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
content="width=device-width, initial-scale=1.0"
/>
    <title>Login Page</title>
    <link rel="stylesheet" href="style.css" />
    <script

src="https://kit.fontawesome.com/176107b5a0.
js"
      crossorigin="anonymous"
    ></script>
  </head>
  <body>
    <div class="container">
      <form action="form">
        <div class="form-container">
          <div class="apply-box">
            <div class="login-page">
              <h2>Login</h2>

              <div class="form-control">
                <label for="username">
Username</label>
                <input
                  type="text"
                  class="username"
                  id="username"
                  placeholder="Enter a valid
username"
                />
              </div>
              <br />
              <div class="form-control">
                <label for="password">
Password</label>
                <input
                  type="text"
                  class="password"
                  id="password"
                  placeholder="Enter a password"
                />
              </div>
              <div class="forget-password">
                <h5><a
href="forgot-password">forgot
password?</a></h5>
              </div>

              <br />

              <div class="button-container">
                <button
type="submit">Login</button>
              </div>

              <div class="option">
                <div class="social_login">
                  <h5>or sign in using</h5>
                </div>
                <div class="icons">
                  <i class="fa fa-instagram"></i>
                  <i class="fa fa-google"></i>
                  <i class="fa fa-twitter"></i>
                </div>
              </div>
            </div>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

style.css

```
@import
url('https://fonts.googleapis.com/css2?family=R
oboto:ital,wght@0,100;0,300;0,400;0,500;0,70
0;0,900;1,100;1,300;1,400;1,500;1,700;1,900&
display=swap');


*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
html,body{
    background-color: pink;
    line-height: 1.4;
    font-family: "Roboto",sans-serif;
}
.form-control{
    display: flex;
    flex-direction: column;
```

```
    margin-top: 10px;
}
.container{
    max-width: 400px;
    margin: 0 auto;

    background: rgb(255, 255, 255);
    border-radius: 10px;
}
.apply-box{
    display: flex;
    justify-content: center;
    align-items: center;

    box-shadow: 4px 3px 5px rgba(1,1,1,0.2);

}
.login-page h2{
    margin-top: 10px;
    text-align: center;
}
label{
    font-size: 20px;

}
.form-control input{
    width: 200px;
    padding: 5px;
}
.forget-password{
    margin-top: 5px;
}
.forget-password h5 a{
    text-decoration: none;
    font-style: none;
    color: rgb(248, 122, 143);

}
.form-container{

    margin-top: 200px;
    display: grid;
    grid-template-columns: repeat(auto-fit,
minmax(200px, 1fr));
    gap: 20px;

}
.button-container{
    display: flex;
    justify-content: center;
    align-items: center;
}
button{
    width: 120px;
    height: 30px;
     background :rgb(243, 95, 119);
     border: transparent solid 2px;
     padding: 5px 10px;
     color: white;
     border-radius: 8px;
     transition:0.3 ease-in ;
     font-size: 15px;
     text-align: centers;

    }
button:hover{
    background: green;
}
.option{
    margin-top: 20px;

}
.social_login h5{
 text-align: center;
}
.icons{
    display: flex;
    flex-direction: row;
    justify-content: center;
    gap: 15px;
    margin-top: 10px;
    margin-bottom: 20px;
}
```

OUTPUT:



Conclusion

HTML and CSS work together as the foundation of web development. While **HTML** provides the structure and content of a webpage, **CSS** enhances its presentation and style, creating visually appealing, user-friendly, and responsive designs. By separating structure from design, developers can maintain consistency, improve code reusability, and simplify updates.

Key takeaways:

- HTML structures the content using elements like headings, forms, buttons, and links.
- CSS customizes the layout, typography, colors, and spacing, offering control over the webpage's aesthetics and responsiveness.
- Together, they enable developers to create professional, accessible, and functional web pages, forming the basis for interactive and dynamic web applications.

Mastering these two technologies is essential for building modern, user-centered websites.