

INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT
KALANKI, KATHMANDU
(AFFILIATED TO TRIBHUVAN UNIVERSITY)



ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT

LAB REPORT

SUBJECT: INTERNET & INTRANET

LAB NO: 02

SUBMITTED BY:

NAME: DIPESH DHUNGANA

ROLL NO: ACE077BCT035

DATE: December 27, 2024

SUBMITTED TO:

DEPARTMENT OF COMPUTER
AND ELECTRONICS

TITLE: DATABASE, JAVASCRIPT AND JQUERY BASICS

OBJECTIVE:

- TO CREATE PAGE WHERE WHEN USER PRESS 'SHOWTIME' BUTTON, THE CURRENT DATE AND TIME OF THE SYSTEM ARE DISPLAYED.
- TO CREATE A DATABASE LOGIN PAGE USING PHP (CREATE TABLE 'USER'), AN INDEX PAGE AND LINK THE PAGE TO DATABASE (INSERT DATA ENTERED IN UI IN DATABASE AND DISPLAY ENTERED DATA IN TABULAR FORM) WITH THE FORM VALIDATION (CHECK FOR USERNAME AND PASSWORD).
*IF USERNAME & PASSWORD FIELD BOTH ARE EMPTY, DISPLAY MESSAGE 'PLEASE ENTER USERNAME & PASSWORD'.

THEORY:

Web development involves the creation of interactive and user-friendly web pages using a combination of technologies like HTML, CSS, and JavaScript. HTML (HyperText Markup Language) is the backbone of any webpage, structuring its content by defining elements such as headers, paragraphs, and buttons. CSS (Cascading Style Sheets) is used to enhance the visual appeal of the page by styling its elements with colors, layouts, fonts, and responsive designs. JavaScript adds interactivity and functionality, enabling dynamic behaviors such as responding to user actions, updating content in real-time, and handling events like button clicks. Together, these technologies provide a robust framework for building web applications that are both visually appealing and functionally rich.

The Objective 1 implementation uses JavaScript, HTML, and CSS to dynamically display the current date and time when a user clicks the "Show Time" button. The JavaScript Date object retrieves the current system date and time, and the toLocaleString() method formats it for user-friendly display. The function showTime() is triggered via an onclick event, which updates the content of the <p> element with id="time-display". This integration demonstrates the use of JavaScript for real-time interactivity on a web page.

Web applications often require user authentication to ensure secure and personalized access. A database-driven login system is a common implementation for this purpose. In this system, user credentials such as username and password are stored in a database. The frontend, designed with HTML and CSS, provides a user interface for inputting credentials, while JavaScript ensures basic validation, such as checking for empty fields. The backend, often built using server-side scripting languages like PHP, processes the login request. It interacts with the database to verify credentials and either grants access or displays appropriate error messages.

The database typically consists of a user table that stores necessary information, such as usernames and hashed passwords. This structure allows secure storage and retrieval of data. Additionally, input validation ensures data integrity by preventing issues such as SQL injection and empty submissions. This system is an essential component of web applications, promoting data security and seamless user experiences.

Regarding Objective 2, A login system is a fundamental component of web applications, ensuring secure access to authorized users. This system typically consists of a frontend interface for user input, backend logic for authentication, and a database for storing credentials. In this project, the frontend (e.g., index.php) provides a user-friendly form for entering a username and password, styled with style.css. The backend, implemented in PHP (e.g., login.php), processes the form data, connects to a MySQL database (db_conn.php), and validates the credentials against the records stored in the user table (db/).

If the username and password are correct, the system redirects the user to a protected area (e.g., home.php). Otherwise, it displays appropriate error messages. The logout functionality (logout.php) ensures secure session termination, preventing unauthorized access. Additionally, input validation and secure password storage (e.g., hashing) are vital to protect against vulnerabilities such as SQL injection and brute force attacks. This implementation demonstrates an end-to-end flow of a secure login system.

SOURCE CODE (OBJECTIVE 1)

1. Showtime.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Show Time</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<header>
  Time Checkpoint
</header>

<main>
  <h1>Press the Button to Display Current Date and Time</h1>
  <div id="time-container">
    <!-- This will display the time -->
    <p id="time-display">Time will appear here...</p>
  </div>
  <button onclick="showTime()">Show Time</button>
</main>

<footer>
  &copy; 2024 Dipesh Dhungana. All Rights Reserved.
</footer>

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

</body>
</html>
```

2. Styles.css

```
/* General Body Styles */
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  background: linear-gradient(to bottom, #f4f4f4, #e8e8e8);
  color: #333;
  text-align: center;
}

/* Header Styling */
header {
  background-color: #4CAF50;
```

```

        color: white;
        padding: 20px 0;
        font-size: 2rem;
        font-weight: bold;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
    }

    /* Main Content Styling */
    main {
        padding: 40px 20px;
    }

    #time-container {
        margin: 20px auto;
        padding: 20px;
        background: white;
        border-radius: 10px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        max-width: 400px;
        font-size: 1.5rem;
    }

    button {
        background-color: #4CAF50;
        color: white;
        border: none;
        padding: 10px 20px;
        font-size: 1rem;
        border-radius: 5px;
        cursor: pointer;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
    }

    button:hover {
        background-color: #45a049;
    }

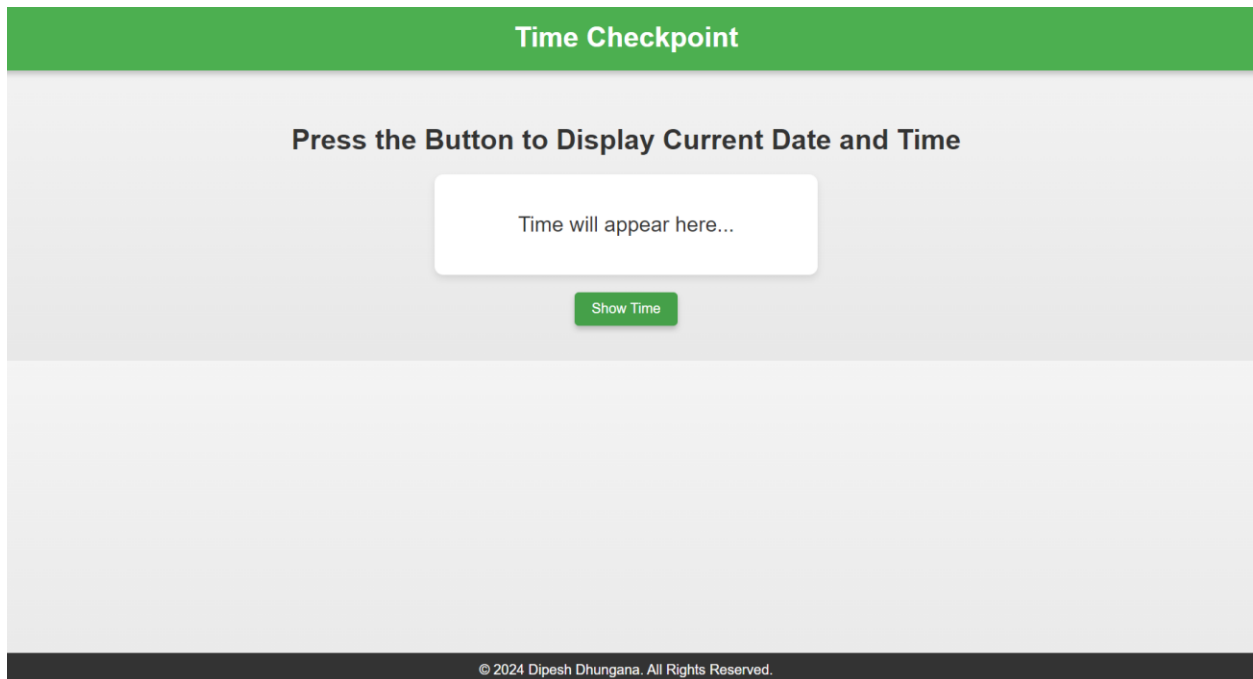
    /* Footer Styling */
    footer {
        background-color: #333;
        color: white;
        text-align: center;
        padding: 10px 0;
        position: fixed;
        bottom: 0;
        width: 100%;
    }

```

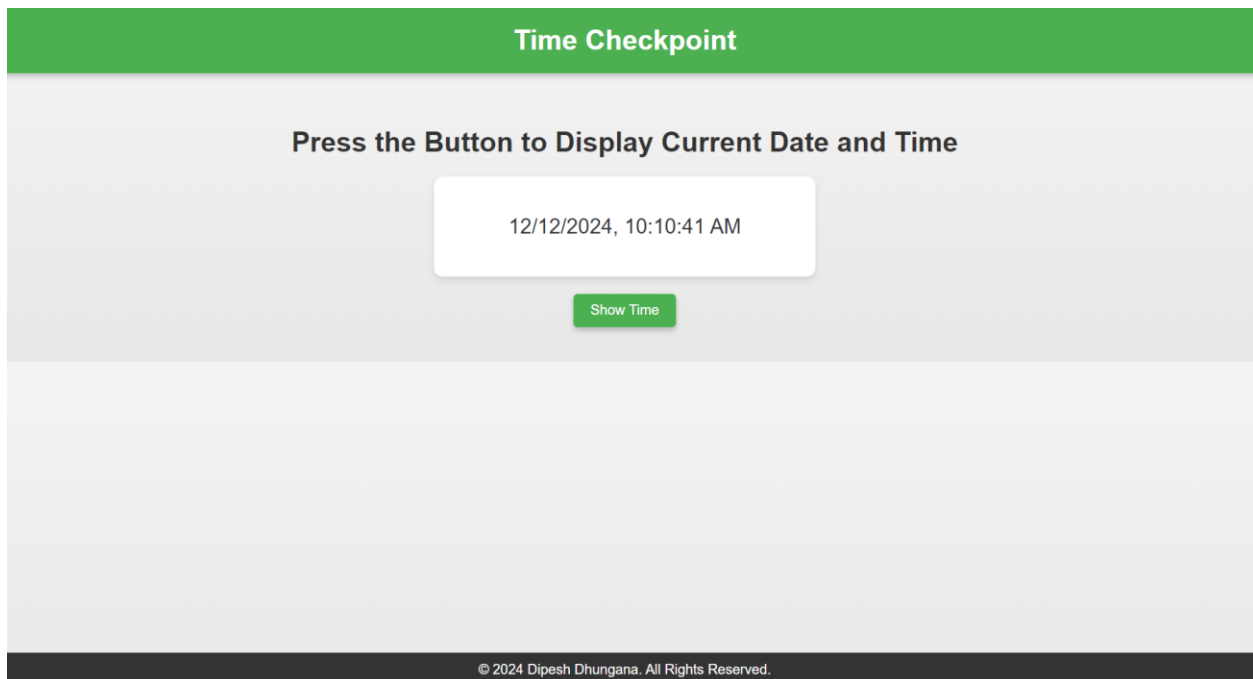
3. Script.js

```
function showTime() {  
    const now = new Date();  
    const formattedTime = now.toLocaleString();  
    document.getElementById('time-display').textContent = formattedTime;  
}
```

OUTPUT 1 (WEBPAGE SCREENSHOT)



(Page Initially)



(Page After Clicking Show Time)

SOURCE CODE (OBJECTIVE 2)

Index.html (Sourcecode)
































```
<!DOCTYPE html>
<html>
<head>
    <title>LOGIN</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <form action="login.php" method="post">
        <h2>LOGIN</h2>
        <?php if (isset($_GET['error'])) { ?>
            <p class="error"><?php echo $_GET['error']; ?></p>
        <?php } ?>
        <label>User Name</label>
        <input type="text" name="uname" placeholder="User Name"><br>

        <label>User Name</label>
        <input type="password" name="password" placeholder="Password"><br>

        <button type="submit">Login</button>
    </form>
</body>
</html>
```

****THE OTHER SOURCECODES OF OBJECTIVE 2 IS ATTACHED AS A FILE AS THE CODE IS LONG****

OUTPUT 2 (LOGIN SYSTEM SNAPSHOTS)

					id	user_name	password	name
<input type="checkbox"/>	 Edit	 Copy	 Delete		1	biplav	biplav1	ACE077BCT030
<input type="checkbox"/>	 Edit	 Copy	 Delete		2	dipesh	dipesh1	ACE077BCT035
<input type="checkbox"/>	 Edit	 Copy	 Delete		3	bimarsha	bimarsha1	ACE077BCT027
<input type="checkbox"/>	 Edit	 Copy	 Delete		4	aayush	aayush1	ACE077BCT005
<input type="checkbox"/>	 Edit	 Copy	 Delete		5	hem	hem1	ACE077BCT041
<input type="checkbox"/>	 Edit	 Copy	 Delete		6	nawraj	nawraj1	ACE077BCT047
<input type="checkbox"/>	 Edit	 Copy	 Delete		7	divyansh	divyansh1	ACE077BCT037
<input type="checkbox"/>	 Edit	 Copy	 Delete		8	bipinale	bipinale1	ACE077BCT028
<input type="checkbox"/>	 Edit	 Copy	 Delete		9	abhishek	abhishek1	ACE077BCT008
<input type="checkbox"/>	 Edit	 Copy	 Delete		10	anish	anish1	ACE077BCT011

(Database Table)

LOGIN

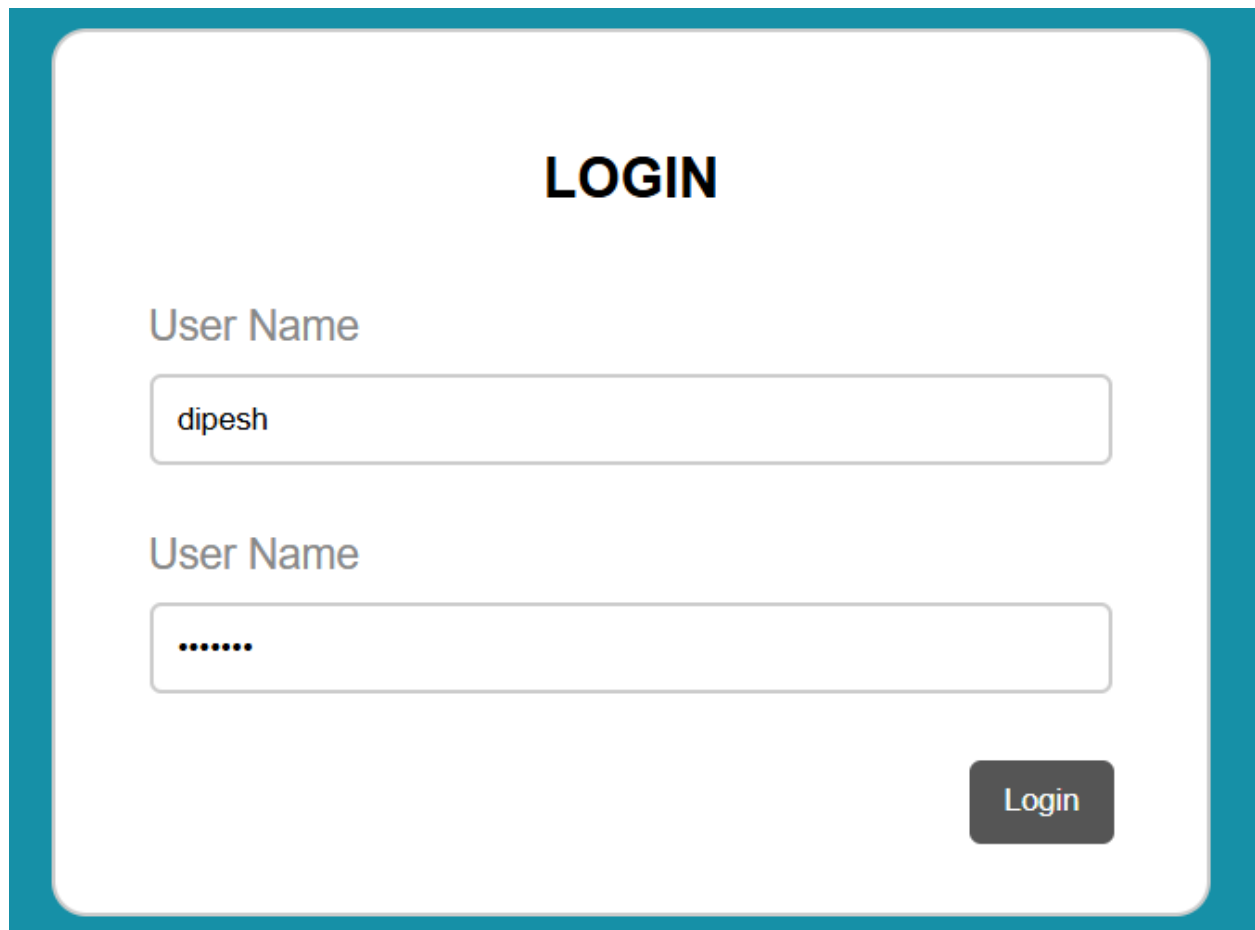
User Name is required

User Name

User Name

Login

(Page Asking for the Credentials)

A login form with a teal border. At the top, the word "LOGIN" is centered in bold black text. Below it, there are two input fields. The first is labeled "User Name" and contains the text "dipesh". The second is also labeled "User Name" and contains seven dots, indicating a password. To the right of the second input field is a dark grey button with the text "Login" in white.

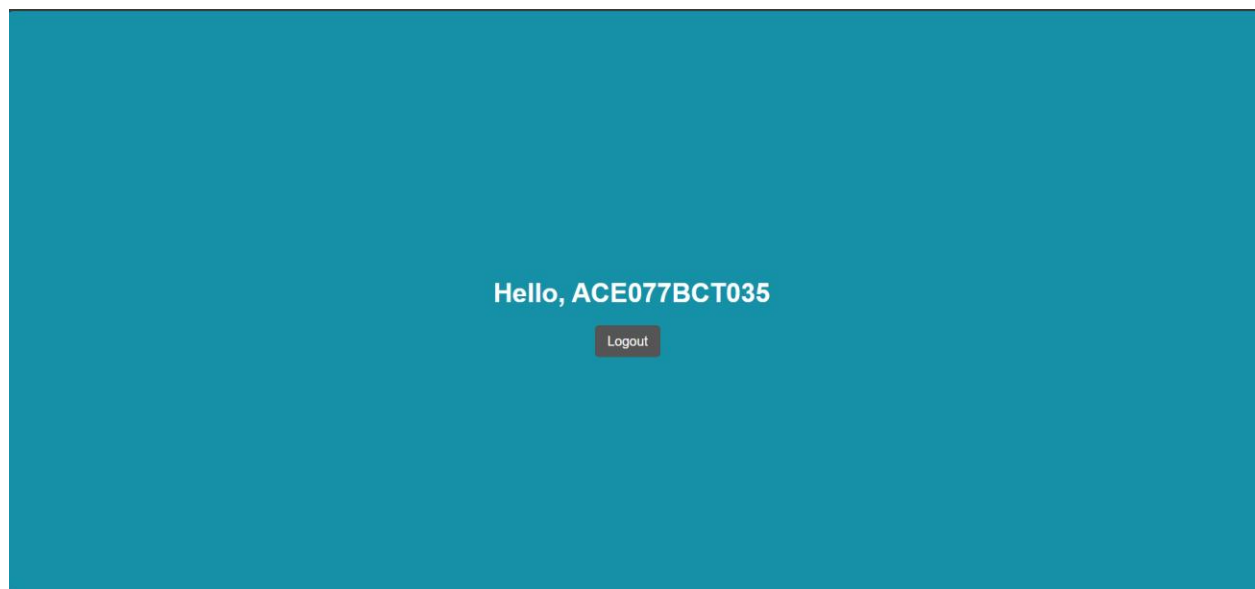
LOGIN

User Name

User Name

Login

(User Details Input)



(Index Page Output after Logging In)

DISCUSSION AND CONCLUSION:

Creating an interactive webpage to display the current date and time showcases the effective use of JavaScript to dynamically update content in response to user actions. The seamless combination of HTML for structure, CSS for styling, and JavaScript for functionality highlights how these technologies work together to enhance user engagement and deliver a polished web experience.

Developing a secure login system demonstrates the importance of integrating frontend interfaces, backend processing, and database management. By validating user inputs, securely storing credentials, and managing sessions, the system ensures reliable authentication and controlled access to protected pages. This emphasizes the critical role of security and database interaction in modern web applications.

Dynamic content updates, as demonstrated in the first implementation, highlight the value of JavaScript in creating responsive and engaging web pages. Similarly, the login system exemplifies a robust approach to user authentication by combining multiple technologies. Together, these implementations provide essential insights into building functional and secure web applications.