***ML  
Module-1***BATCH-12

Aiml-E

***P. Man mohan***

Department of ACSE(AI&ML) *231FA18167*  
 VFSTR(Deemed to be)University [231fa18167@gmail.com](mailto:231fa18167@gmail.com)

**K. Sai Kumar**

Department of ACSE(AI&ML)  
 *231FA18429* VFSTR(Deemed to be)University  
 [231fa18429@gmail.com](mailto:231fa18429@gmail.com)

R. Yamini   
Department of ACSE(AI&ML)  
*231fa18139*VFSTR(Deemed to be)University  
[231fa18139@gmail.com](mailto:231fa18139@gmail.com)

**P.Gajendra Monish**

Department of ACSE(AI&ML)  
 231FA18442 VFSTR(Deemed to be)University  
 [231fa18442@gmail.com](mailto:231fa18442@gmail.com)

### Abstract

User authentication is a critical aspect of information security in modern digital systems. This paper presents the design and implementation of a simple login page based on ****Username and Date of Birth (DOB) verification****. The proposed system validates inputs, checks credentials against stored data, and provides clear feedback to the user. A sample implementation using ****HTML, CSS, and JavaScript**** is discussed along with decision-making logic and flow diagrams. The system demonstrates how lightweight web-based authentication can be implemented for academic or small-scale applications.

INTRODUCTION

### Authentication ensures that only authorized users gain access to restricted resources. Traditional login systems rely on **username and password**, but alternative approaches like **username with Date of Birth (DOB)** are also used in academic and institutional environments for simplicity. This paper focuses on the design of a **login page** that validates both username and DOB, checks correctness, and grants access upon successful validation.

## II. Related Work

Many authentication techniques exist, ranging from basic credential checks to advanced mechanisms like **two-factor authentication** and **biometric systems**. While these provide stronger security, they may be unnecessary for low-risk applications. The presented system lies in the category of lightweight authentication, where the focus is on **simplicity, clarity, and usability**.

## III. Methodology

The proposed login system follows three main steps:

1. **Input Validation** – Checks whether both fields (username and DOB) are provided.
2. **Format Checking** – Ensures DOB follows the DD/MM/YYYY format.
3. **Credential Verification** – Compares entered details with stored data (e.g., admin, 01/01/2000).

## IV. Implementation

### A. Logic Required

* function validateInput(username, dob) {
* if (!username || !dob) {
* return "Both fields are required.";
* }
* const dobPattern = /^\d{2}\/\d{2}\/\d{4}$/;
* if (!dobPattern.test(dob)) {
* return "Invalid DOB format. Use DD/MM/YYYY.";
* }
* return "Input looks valid.";
* }

***B. Decision Making Process:***

*const storedUser = "admin";*

*const storedDOB = "01/01/2000";*

*function checkCredentials(username, dob) {*

*if (username !== storedUser) {*

*return "Invalid Username";*

*} else if (dob !== storedDOB) {*

*return "Incorrect Date of Birth";*

*} else {*

*return "Access Granted";*

*}*

*}*

### C. Login Page Design (HTML, CSS, JavaScript)

* Front-end designed with HTML & CSS.
* Logic handled using JavaScript.
* Sample user messages:
  + ✅ "Login Successful, Welcome!"
  + ❌ "Invalid Username or DOB, Try Again."

Code: <!DOCTYPE html>

<html>

<head>

<title>Login Page</title>

<style>

body {

font-family: Arial, sans-serif;

background: #f0f2f5;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.login-box {

background: white;

padding: 20px;

border-radius: 10px;

box-shadow: 0px 4px 10px rgba(0,0,0,0.2);

width: 320px;

text-align: center;

}

.login-box h2 {

margin-bottom: 15px;

}

input {

width: 90%;

padding: 10px;

margin: 8px 0;

border-radius: 5px;

border: 1px solid #ccc;

}

button {

background: #007bff;

color: white;

border: none;

padding: 10px 15px;

border-radius: 5px;

cursor: pointer;

width: 95%;

}

button:hover {

background: #0056b3;

}

.message {

margin-top: 12px;

font-weight: bold;

color: #333;

}

.success {

color: green;

}

.error {

color: red;

}

</style>

</head>

<body>

<div class="login-box">

<h2>Login</h2>

<input type="text" id="username" placeholder="Enter Username" />

<input type="text" id="dob" placeholder="DD/MM/YYYY" />

<button onclick="login()">Login</button>

<div id="message" class="message"></div>

</div>

<script>

// Predefined stored credentials

const storedUser = "admin";

const storedDOB = "01/01/2000";

// Function to validate input fields

function validateInput(username, dob) {

if (!username || !dob) {

return "Please fill in all fields.";

}

const dobPattern = /^\d{2}\/\d{2}\/\d{4}$/;

if (!dobPattern.test(dob)) {

return "Invalid DOB format. Use DD/MM/YYYY.";

}

return "valid";

}

// Function to check login credentials

function login() {

let username = document.getElementById("username").value.trim();

let dob = document.getElementById("dob").value.trim();

let messageBox = document.getElementById("message");

// Validate input

let validation = validateInput(username, dob);

if (validation !== "valid") {

messageBox.innerText = "❌ " + validation;

messageBox.className = "message error";

return;

}

// Check credentials

if (username === storedUser && dob === storedDOB) {

messageBox.innerText = "✅ Login Successful! Welcome.";

messageBox.className = "message success";

// Redirect or allow access

// window.location.href = "home.html"; // optional redirect

} else {

messageBox.innerText = "❌ Invalid Username or DOB. Try Again.";

messageBox.className = "message error";

}

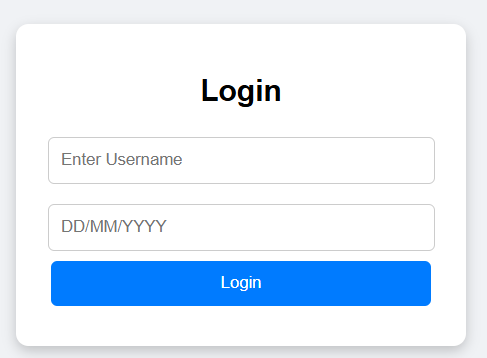
}

</script>

</body>

</html>

### Output:



## V. Decision Flow Diagram

**Start → Enter Username & DOB → Validate Inputs → Check Stored Data → Access Granted / Access Denied**

## VI. Results

* If correct credentials are entered → **Access Granted**.
* If incorrect username or DOB is entered → **Error message with retry option**.
* The system demonstrates **real-time feedback** and ensures proper **input validation**.

## VII. Conclusion

This paper demonstrates the design and development of a basic authentication system using **Username and Date of Birth verification**. The approach provides a simple yet effective solution for applications requiring lightweight authentication, such as **academic projects, small websites, or controlled environments**. Future enhancements may include integrating **passwords, OTPs, or biometric checks** to improve security.

## References

1. W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson, 2017.
2. IEEE Computer Society, “IEEE Standard for Information Technology—Security Techniques,” IEEE Std 27001, 2020.
3. Mozilla Developer Network (MDN), “Web Authentication & JavaScript Documentation,” [Online].

**\*\*\*\***