

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Welcome Back</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      display: flex;

      flex-direction: column;

      align-items: center;

      justify-content: center;

      height: 100vh;

      margin: 0;

      background-color: #f0f0f0;

    }

    .container {

      text-align: center;

      background: white;

      padding: 20px;

      border-radius: 10px;

      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    }

    input {

      display: block;

      width: 100%;

      padding: 10px;

      margin: 10px 0;

      border: 1px solid #ccc;

      border-radius: 5px;
```

```
}  
button {  
    padding: 10px 20px;  
    background-color: #007bff;  
    color: white;  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
}  
button:hover {  
    background-color: #0056b3;  
}  
video {  
    display: block;  
    margin: 20px auto;  
    border: 2px solid #ccc;  
    border-radius: 10px;  
}  
.hidden {  
    display: none;  
}  
</style>  
</head>  
<body>  
    <div class="container">  
        <h1>Welcome Back</h1>  
        <form id="loginForm">  
            <input type="text" id="loginId" placeholder="Login ID">  
            <input type="password" id="password" placeholder="Password">  
            <button type="submit">Login</button>  
        </form>
```

```

<video id="video" width="320" height="240" autoplay class="hidden"></video>

<button id="faceLoginButton">Login with Face</button>
</div>

<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/blazeface"></script>
<script>

  const faceLoginButton = document.getElementById('faceLoginButton');
  const video = document.getElementById('video');
  const loginForm = document.getElementById('loginForm');

  faceLoginButton.addEventListener('click', async () => {
    video.classList.remove('hidden');
    await detectFace();
  });

  loginForm.addEventListener('submit', (e) => {
    e.preventDefault();
    alert('Logging in with password...');
    // Add your actual login logic here
  });

  async function setupCamera() {
    const stream = await navigator.mediaDevices.getUserMedia({
      video: { facingMode: 'user' }
    });
    video.srcObject = stream;
    return new Promise((resolve) => {
      video.onloadedmetadata = () => {
        resolve(video);
      };
    });
  }

```

```
}
```

```
async function detectFace() {
```

```
  const videoElement = await setupCamera();
```

```
  const model = await blazeface.load();
```

```
  videoElement.play();
```

```
  const detect = async () => {
```

```
    const predictions = await model.estimateFaces(videoElement, false);
```

```
    if (predictions.length > 0) {
```

```
      alert('Face detected! Logging in...');
```

```
      videoElement.pause();
```

```
      videoElement.srcObject.getTracks().forEach(track => track.stop());
```

```
      // Add your actual login logic here
```

```
      return;
```

```
    }
```

```
    requestAnimationFrame(detect);
```

```
  };
```

```
  detect();
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```