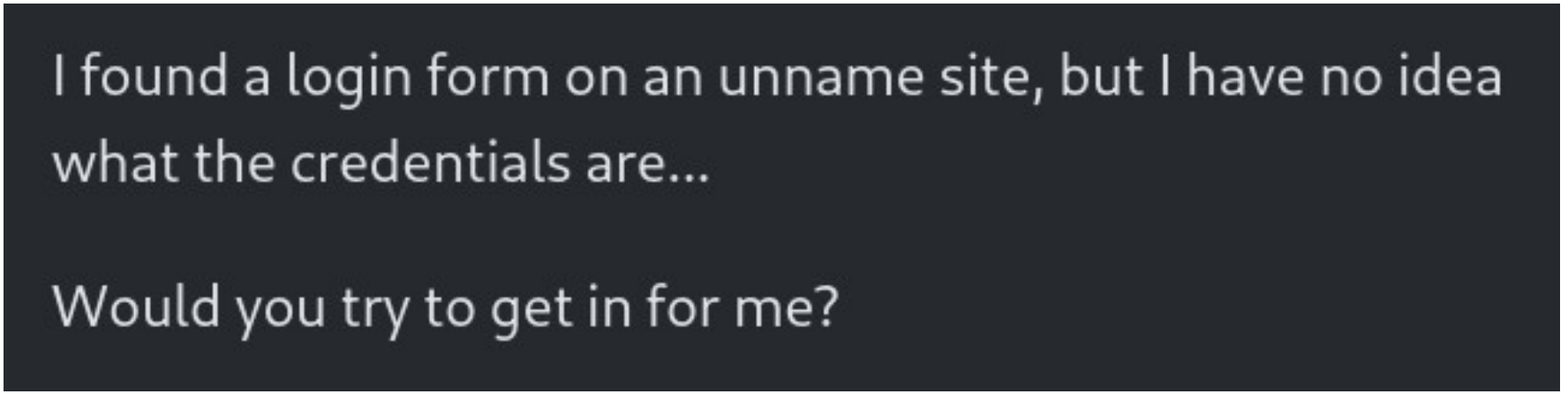
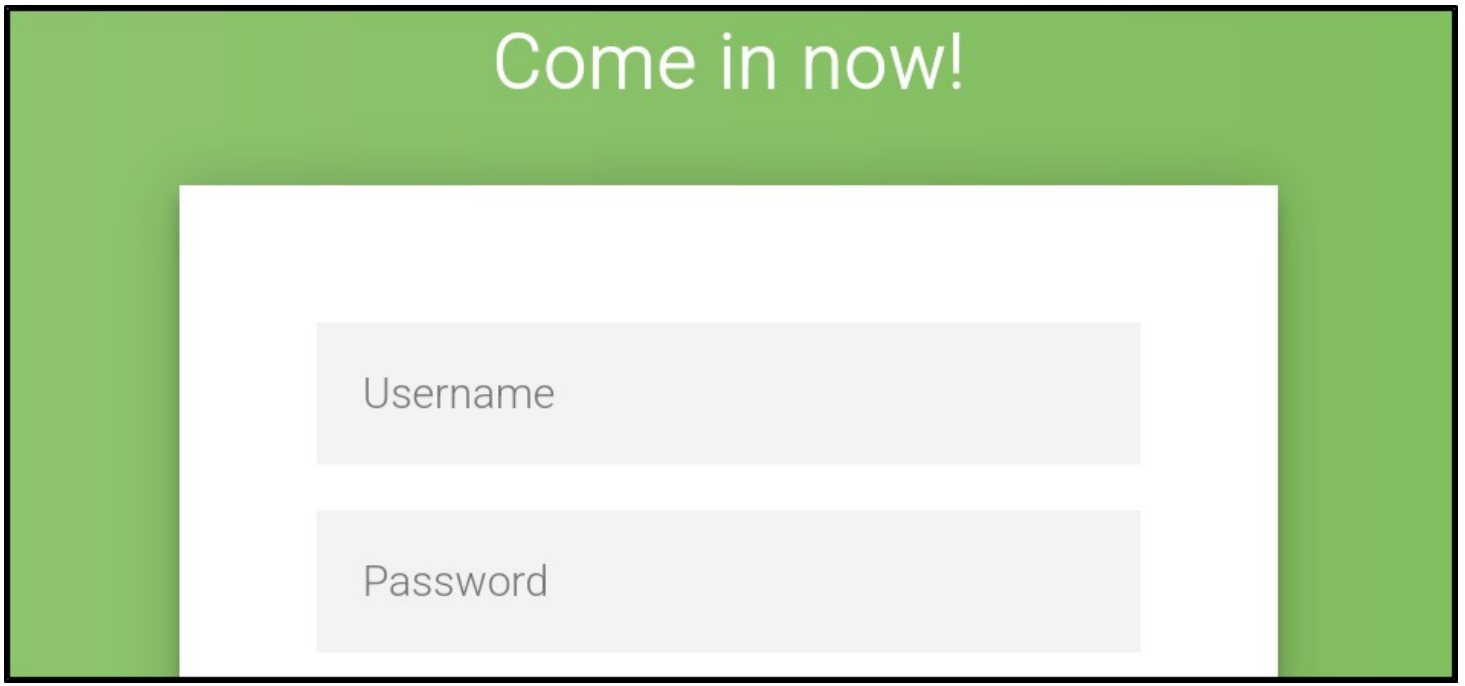
OLIcyber – Just a Reminder



The description for this challenge asks us to login to a website

OLIcyber – Just a Reminder



The website doesn't give us any indication to the correct username and password at first glance..

Server-side Authentication Code

```
<script src="default.js"></script>
<body>
  <div class="login-page">
    <p class="welcome">Entra ora!</p>
    <div class="form">
      <form class="login-form">
        <input type="text" placeholder="Usern
        <input type="password" placeholder="P
        <button onclick="login_check()">login
```

The source code for the page indicates a couple of JavaScript files. If we look at the default.js file that holds the code for the login_check function..

Server-side Authentication Code

```
<script src="default.js"></script>
<body>
  <div class="login-page">
    <p class="welcome">Entra ora!</p>
    <div class="form">
      <form class="login-form">
        <input type="text" placeholder="Usern
        <input type="password" placeholder="P
        <button onclick="login_check()">login
```

This challenge is a reminder that authentication code should never be included in client-accessible files, otherwise malicious users could reverse engineer credentials, which we can do here..

Server-side Authentication Code

```
function login_check() {
  if (
    username_field.value === 'admin' &&
    AES_decrypt('U2FsdGVkX1/JEKDXgPl2Rqt
      password_field.value
```

The contents of the JS script with the login_check function defines a username 'admin', and a password compared against a AES decryption of a defined value with the s3cr37 key...

Server-side Authentication Code

```
var s3cr37 = 'ML4czctKUzigEeuR';
```

```
console.log(AES_decrypt
('U2FsdGVkX1/JEKDXgPl2RqtEgj0LMdp8/
Q1FQelH7whIP49sq+WvNOeNjjXwmdrl', s3cr37))
```

Which is defined in another part of the code. We can run the code through the web-browser JavaScript console to get the password...