**Report on implemented password and captcha features in the System**

**Introduction:**

In our increasingly connected and complicated world, security is critical. Security and cybersecurity have risen to the top of the priority list, gaining widespread international attention as a result of recent assaults and data breaches. A suitable degree of security necessitates a significant amount of effort, commitment, and time. This report will provide an overview of password security and captcha implemented in the system.

**Features implemented in the System:**

The system input fields are all authenticated and it is compulsory for users to fill out all the form fields. It also check whether the entered field match the pattern defined in system or not. The system check the email address entered in register form has been already used if it is in the system database then it will not allow user to register with that email. The system allow user to compare their password by entering in the two input fields so that they can avoid entering wrong password.

To increase the strength of user-chosen passwords, users are typically required to adhere to a set of rules known as password guidelines when creating passwords. Users compose their passwords following the speciﬁc requirements given in the guidelines. For example, the password must contain at least eight characters including at least one number or one upper case letter, lower case letter and special character. Password restriction policies which is implemented in the system check if the content and format of the password meet the requirement of system (M.Yildrim, 2019). To measure the quality of a given password, password strength checker is implemented in the system which usually encourage user to use the strong password. An apparent better approach for the system is to provide better feedback to the password on the quality of their chosen passwords so that the password cannot be easily guessed by others (Xavier de Carné de Carnavalet, 2019).  The strength of the password is determined by the use of numbers, lowercase and uppercase letters, and various characters and its length (Nulab, 2016).

**Use of Cryptographic hashing function (md5) for encryption in the system:**

Use of MD5 Cryptographic encryption in the system make the secure in communication due to its protection on user’s identity or data from being read and protect data from being changed. It also ensure data originates from a particular party. The system employs cryptographic hashes due of their ability to convert data of arbitrary length to a fixed-length byte sequence. Hashes are statistically distinct; no two-byte sequence will hash to the same result. In the system, a secret-key algorithm known as a block cipher is employed to encrypt one block of data at a time. Because it does not utilize an initialization vector to initialize the first plaintext block, the electronic codebook (ECB) block cipher is employed for password hashing. The system primarily uses cryptographic md5 hash functions for information security reasons, such as digital signatures, message authentication codes (MACs), and other types of authentication (ZAIN, 2014). Its principal applications are password verification apps and authentication applications (ZAIN, 2014). In md5 hashing algorithm, the output hash value will vary if the input file undergoes even a little modification where a password is supplied then it computes the password’s hash and compares it with the stored value. This makes it incredibly simple to verify (or refute) the legitimacy of a piece of digital evidence since it makes it difficult to change a file without also changing the accompanying hash value (T. Ebanesar, 2019). The best method for securing passwords is secure hashing. Passwords that had been hashed were impossible to crack, steal, or hack. Anyone could not utilize a stolen hash code because the attacker won't know which user record is updated, hashing the password prevents an attacker with database access from changing application privileges (T. Ebanesar, 2019).

**Implementation of Google Recaptcha V2 in the system:**

System is based on Google Recaptcha V2 which provides more security against attacks. Google Captcha is a program that generate score test that humans can pass easily but not current computer programs to bypass it (Suresh Kumar Krishnamoorthy, 2017). It only allow authorized users to register on the system by providing an images of traffic roads. The algorithm will check the integrity of the user by analyzing the performance of the user. Image captchas require that the user comprehend the images in the provided captcha from a semantic perspective and carry out identification operations (such as choosing semantic images or clicking semantic areas) in accordance with the on-screen instructions. Because they are more user-friendly and seen as being more secure than text captchas, image captchas are becoming more and more prevalent today (Haiqin Weng, 2019). About 1 billion people have completed image captchas, according to a study from Tencent's captcha service (cloud.tencent.com). By adding a Google Recaptcha on the system, system can block automated software from spam and abuse. Recaptcha difference between human and bots and help the system to run smoothly.

System also have the password changing policy every 30 days. When user’s password stored in database are older than 30 days than it automatically redirect user to update page to change the password. The user enter password will be saved for years to prevent users for using the old password, which ensures that the user’s data remains secure and cannot be accessed by unauthorized persons. User can also reset their password using email link, where users will be send with one time temporary password which will expire in 24 hours. The TempPassword database holds the temporary password. In this manner, the user also has the option of using his previous password to log in. When a user signs in, the system examines the database to see if they are using their old or temporary password. The user is led to the Update page to modify the temporary password using the temporary password; once this is done, the temporary password is removed from the database and is no longer usable.

**Conclusion:**

The project on the cyber security implementation in the system was done with establishing full functional related to cyber security. A user –friendly graphical interface web pages where user can register, login, reset their password and update password. User password are stored in well secure manner using md5 cryptographic encryption. User password will expire on 30 days after password update for the security of the system. User also cannot use the last five password to update their password. User feedback on their password are implemented as weak, average, strong and excellent and use of 6 digits password with special character, One capital letter, one numeric and small letter to assist the user for using proper password. The web app is also well secure from the sql injection.

# References

Haiqin Weng, B. Z. S. J. J. C. T. W. Q. H. a. R. B., 2019. Towards Understanding the Security of Modern Image Captchas and Underground Captcha-Solving Services. *BIG DATA MINING AND ANALYTICS,* 2(2), pp. 118-114.

M.Yildrim, I., 2019. Encouraging users to improve password security and memorability. *International Journal of infromation Security,* Issue 18, pp. 741-759.

Nulab, 2016. Five algorithms to measure real password strength. *Five algorithms to measure real password strength.*

Suresh Kumar Krishnamoorthy, S. T., 2017. A novel method to authenticate in website using CAPTCHA-based validation. *A novel method to authenticate in website using CAPTCHA-based validation.*

T. Ebanesar, G. S., 2019. Improving Login Process by Salted Hashing Password Using SHA-256 Algorithm in Web Applications. *International Journal of Computer Sciences and Engineering,* 7(3).

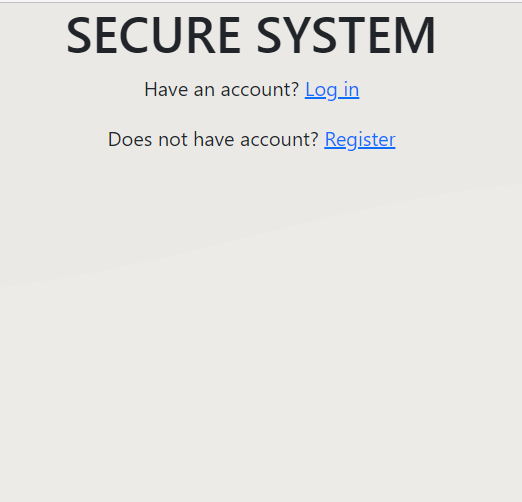
Xavier de Carné de Carnavalet, M. M., 2019. From Very Weak to Very Strong: Analyzing Password-Strength Meters. *Analyzing Password-Strength Meters.*

ZAIN, S. N. B. M., 2014. SECURING PASSWORD AUTHENTICATION USING MD5 ALGORITHM.

**Appendices:**

**Startup Page (Default.aspx)**

**These startup page has with two option to either login or register.**



**Fig1:** Startup page

**Startup-Backend (Default.aspx.cs)**

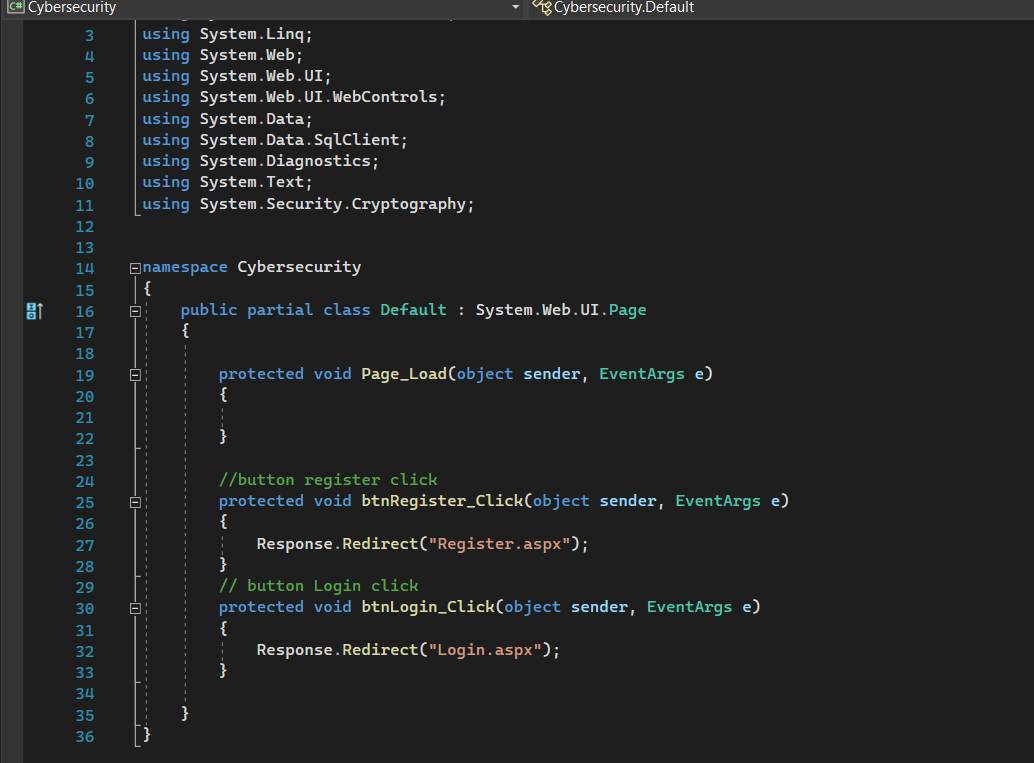
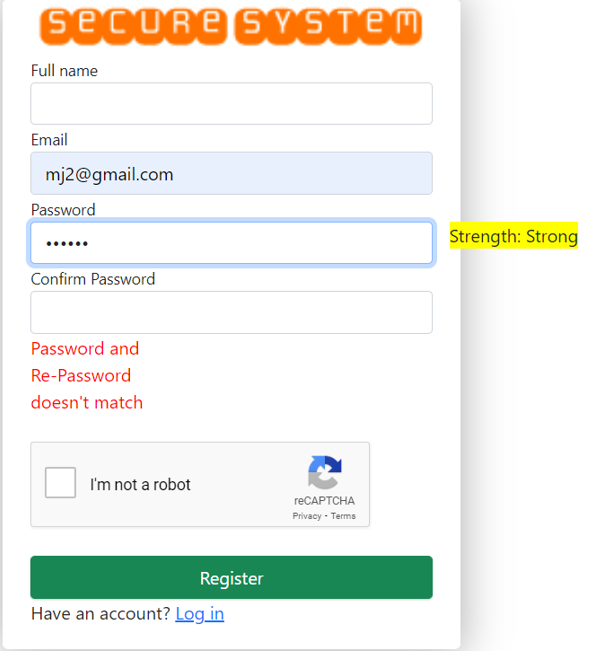
****

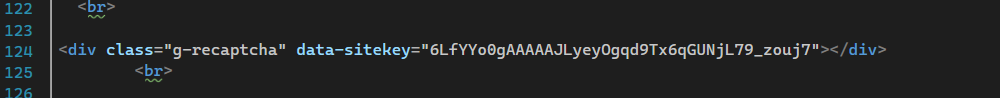
Fig2: Default.aspx.cs

**Register Form with captcha and password validation (register.aspx):**



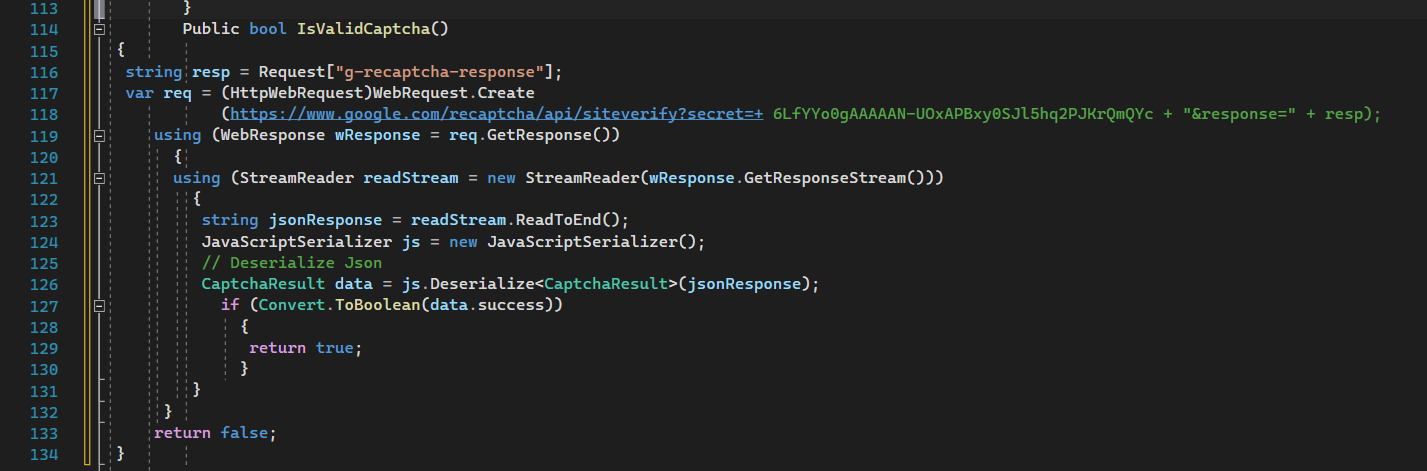
**Fig3:** Server site code of Default.aspx.cs

**Google Recaptcha V2 view on front-end(register.aspx):**



**Fig4:** Captcha

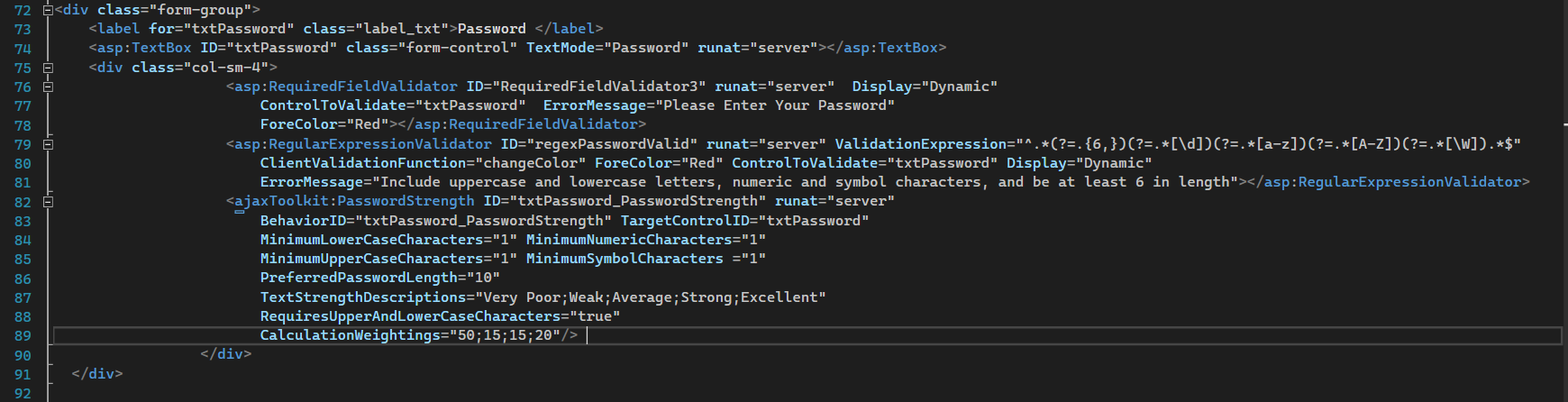
Google Recaptcha V2 Validation on (Register.aspx.cs) from line 114 to 134.



**Fig5:** Captcha Validation on server side

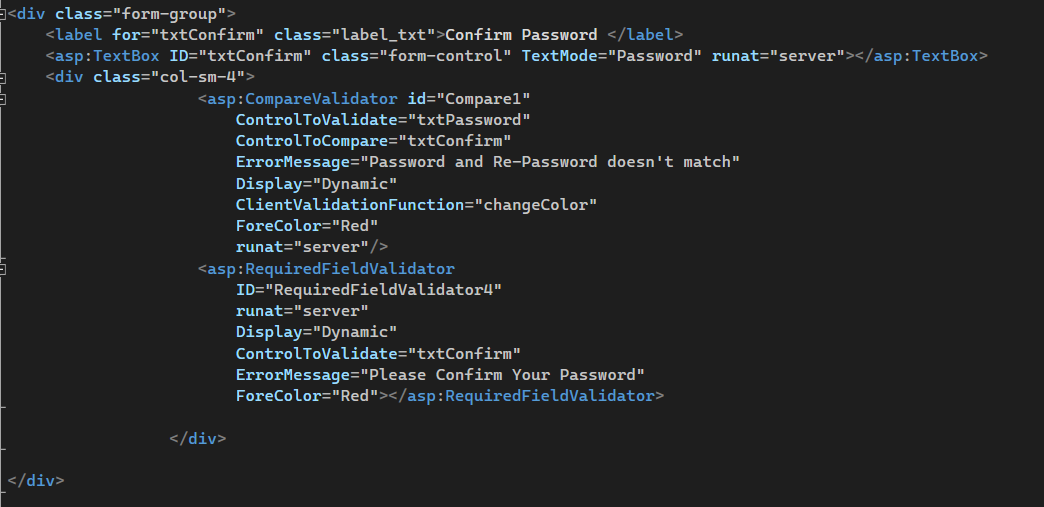
**Feature implementation of Password strength (register.aspx):**

The Password input field had been authenticated (Password Strength in register.aspx from line 72 to 89):



**Fig6:** Password Filed authenticated

**Confirm Password Field authenticated (register.aspx):**



**Fig7:** Confirm password Field

**Email input field has been authenticated:**



**Fig8:** Email input field

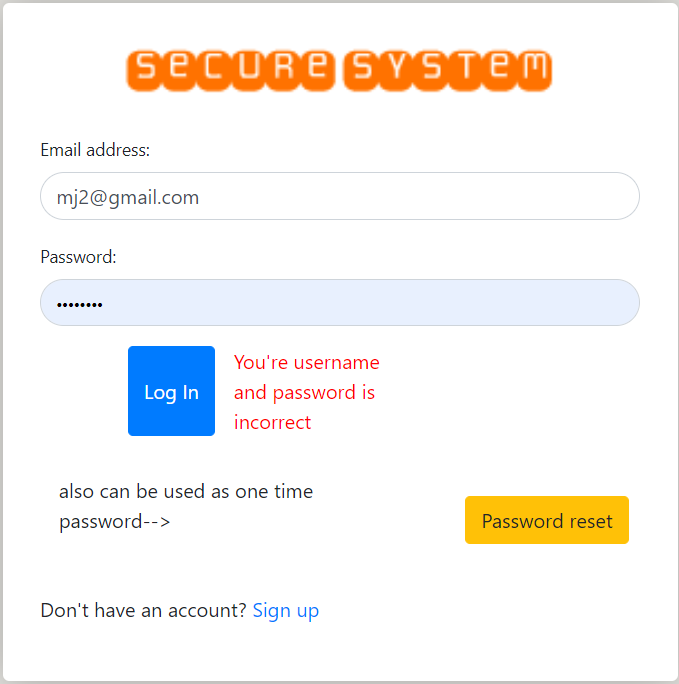
**Password Encryption (register.aspx.cs):**



**Fig9:** Password Hashing

**Login page (login.aspx):**

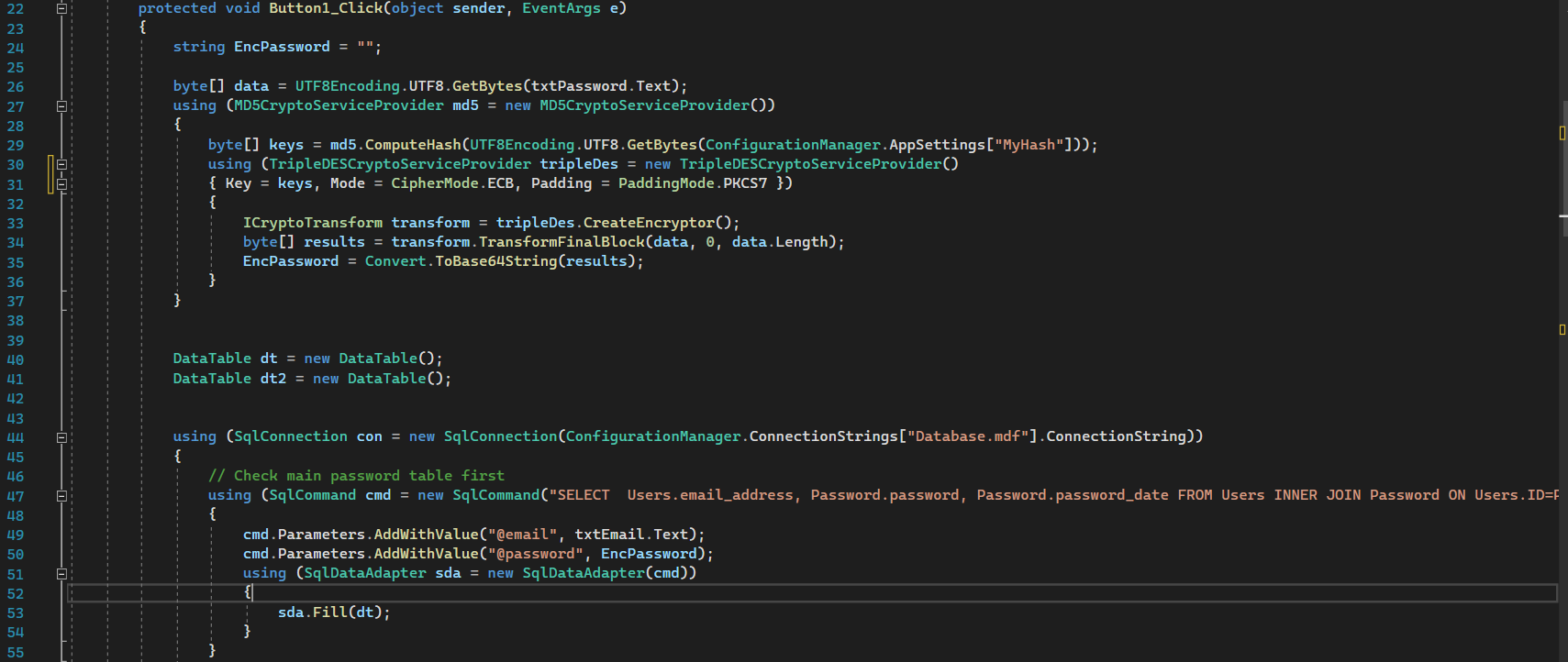
User can access the page through their registered email and password store in database. If user enter incorrect password or email then the system will deny them to login by showing error. In addition user can also reset their password using password reset button in case they forget their password.



**Fig10:** Login Form

**Server-Side of Login page (login.aspx.cs):**

User email and password will be check from the database which are store in encryption.



**Fig10:** Login form code (server-side)

**Login Form (Login.aspx.cs) Shows:** First, check the main password database, then the temp password table for a reset. The primary password match, Check for passwords that are less than a month old. Show a dialog and redirect to the update page, Change the user status to logged in session. Password substitution, Check that the password is not more than 24 hours old. Set the user status in the session to logged out.

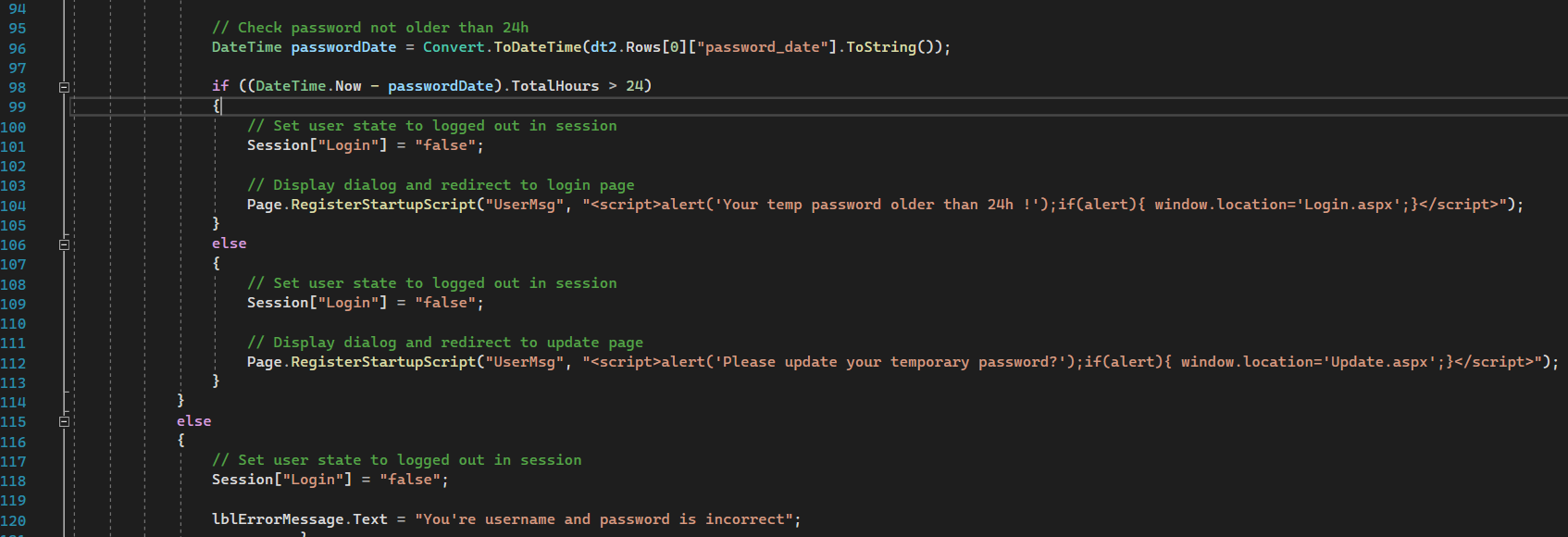


Fig11: Check Password (login)

**Password Expire in 30 days (Login.aspx.cs)**

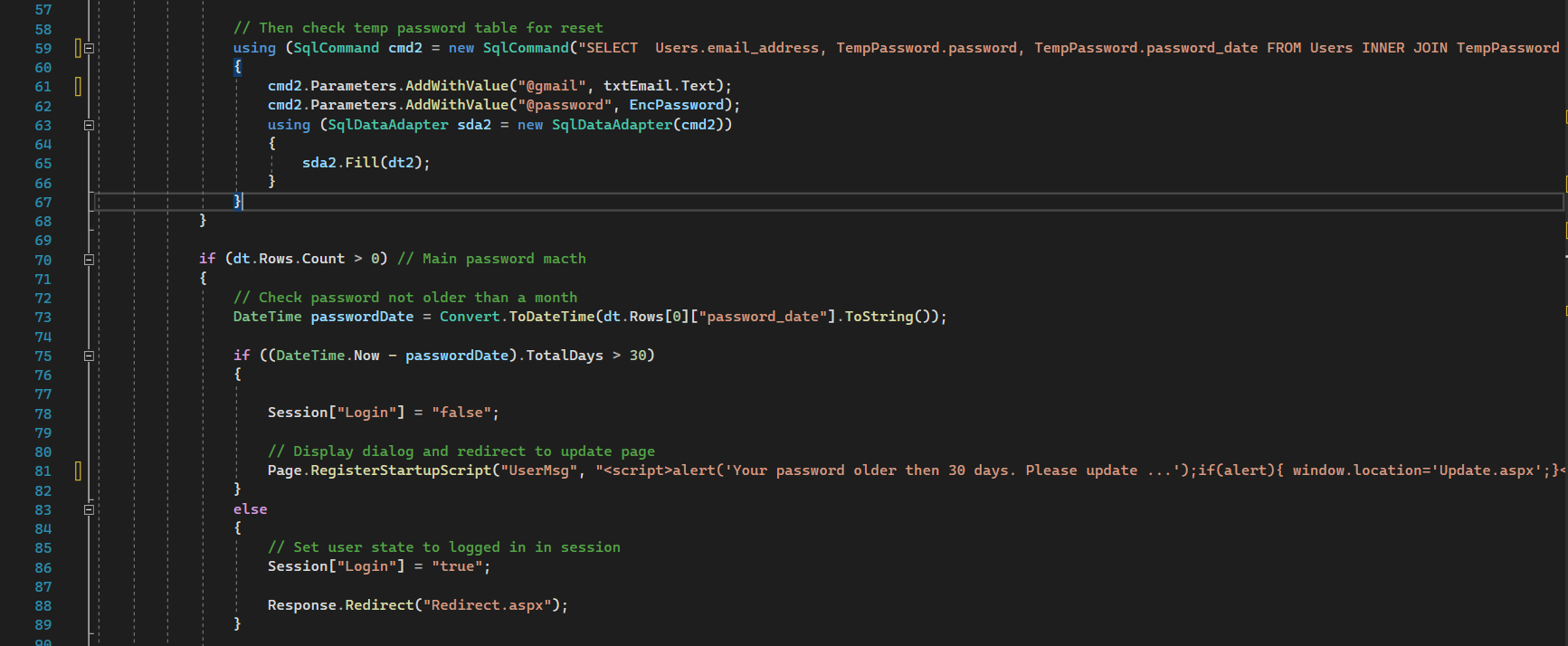


Fig12: Password Expire

**Password Update Page (update.aspx):**

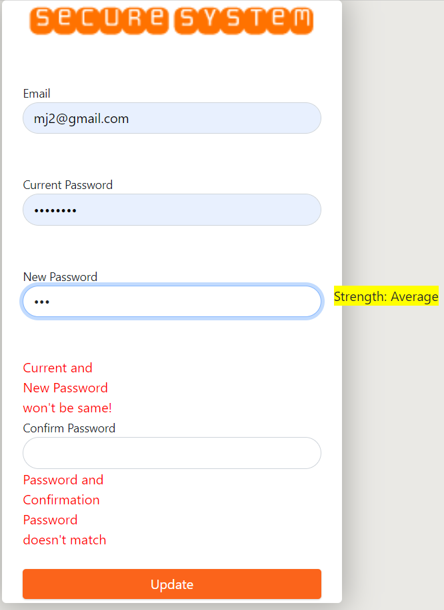


Fig13: Login Form

**Repetition of password (Update.aspx.cs):**

User cannot use the password which has been used for last 1 year. The old password are store in old password table then the system will check if the user has enter the old password or not.

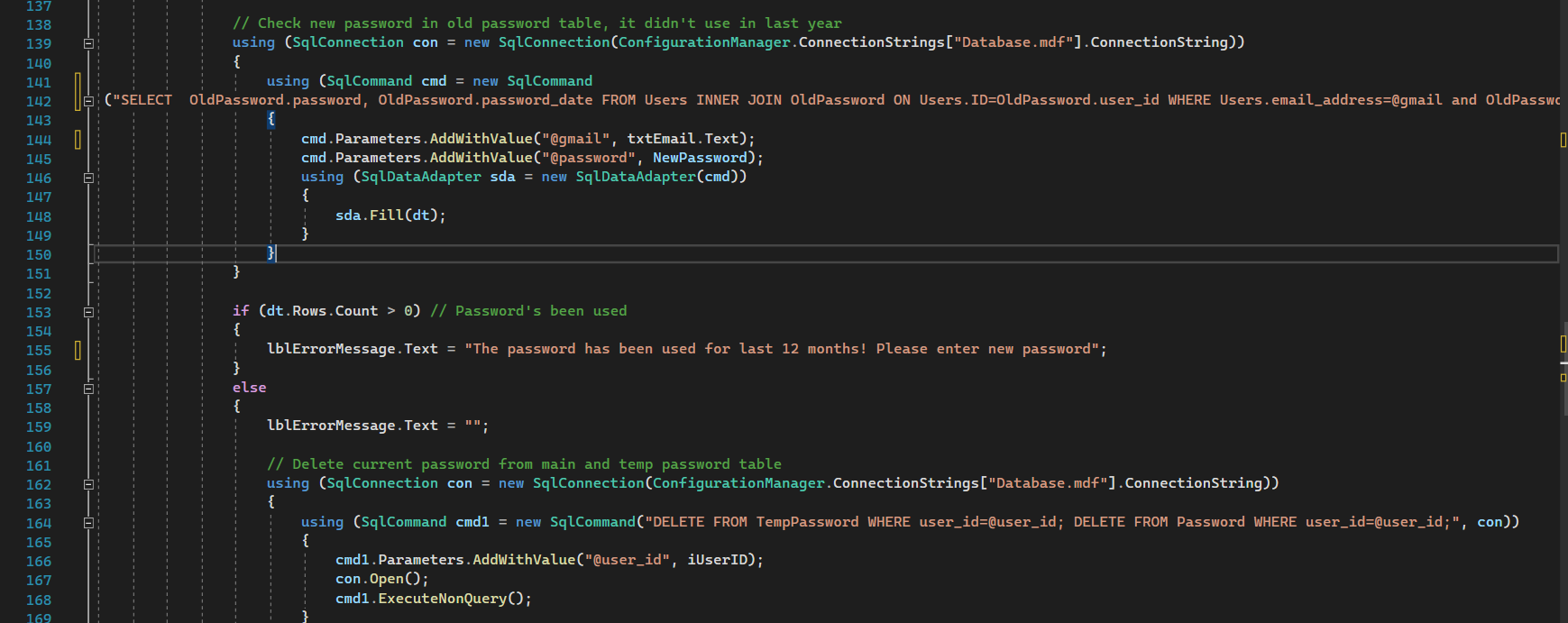


Fig14: Check Old password

**Password Update Form (Update.aspx.cs) Shows:** Check email and password matching, Encrypt old password, Check main password table first.

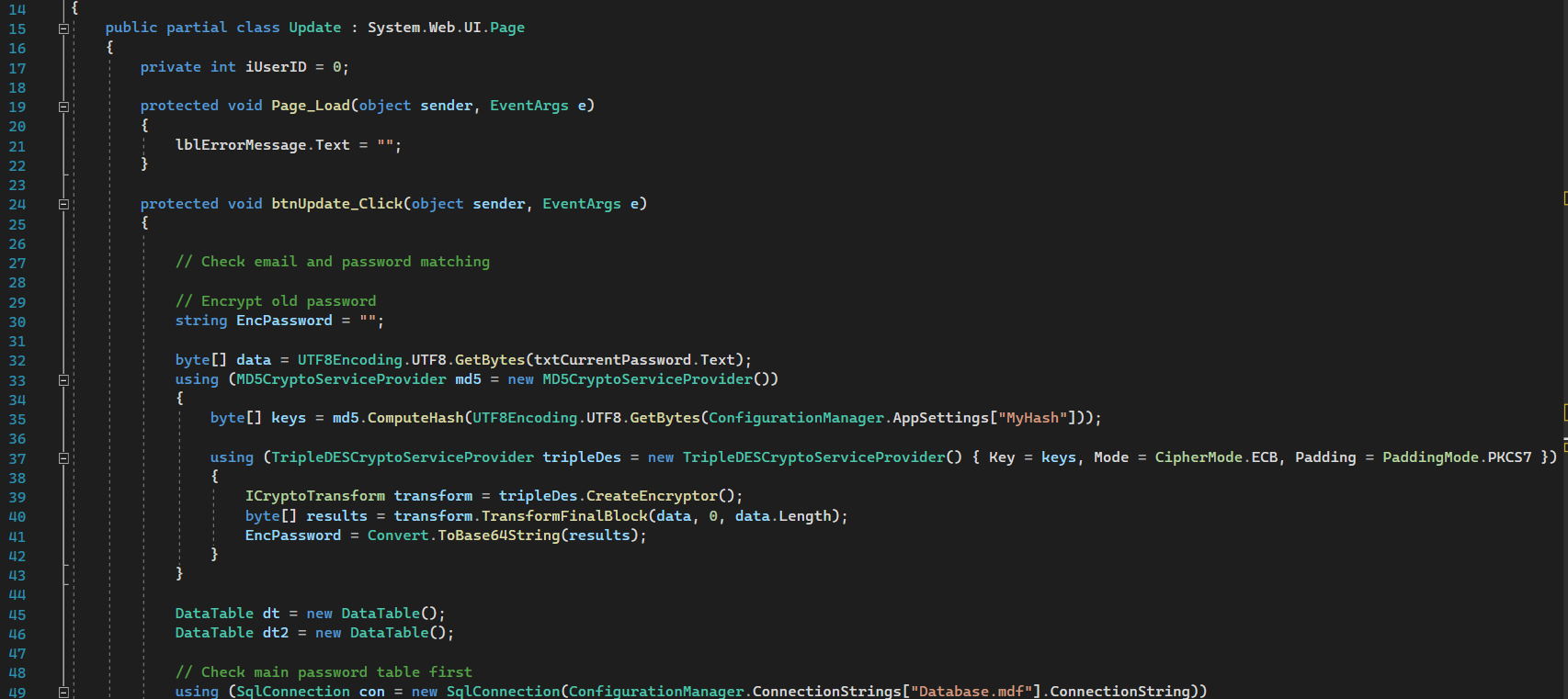


Fig15: Check email and password Matching

**Password Update Form (Update.aspx.cs) Shows:** Then check temp password table for reset, Main password match, Temp password match, Check password not older than 24h.

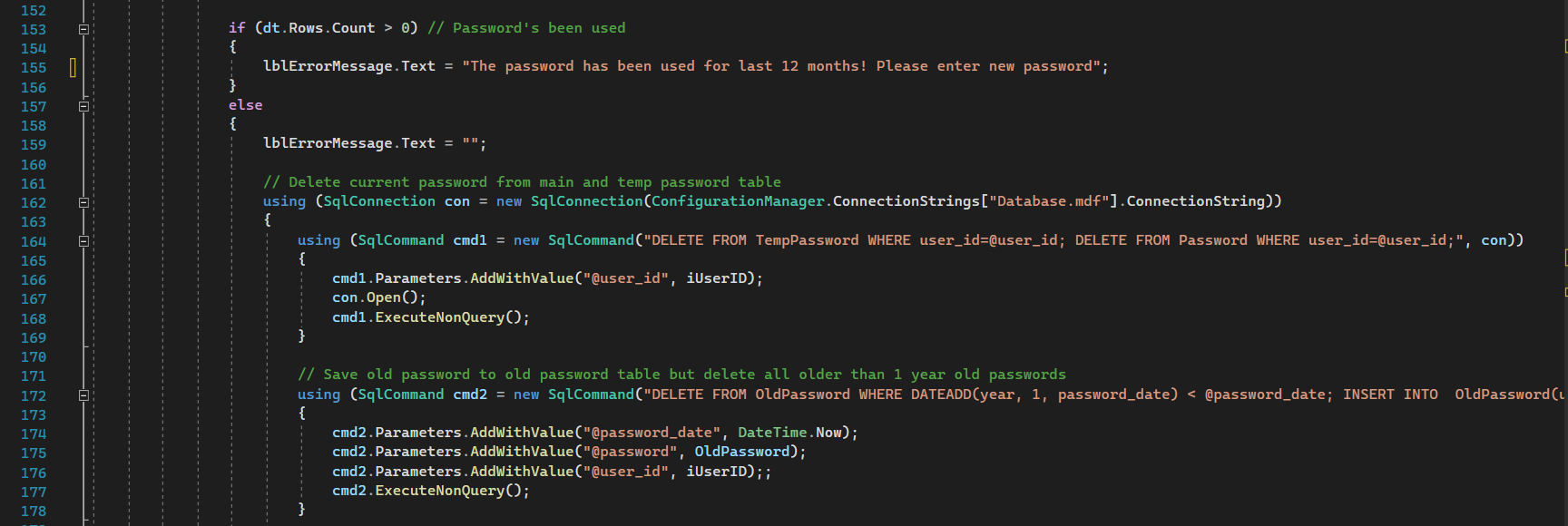


Fig16: Check password not older than 24 hours

**Reset Form (Reset.aspx)**

On the Reset form page, you will receive a generated new password in an email by entering the email address. Once this is done, you can update your password and to access the Redirect page.

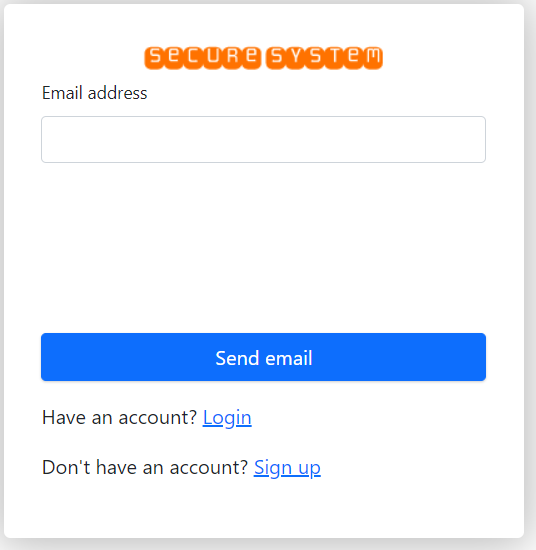


Fig17: Reset Form

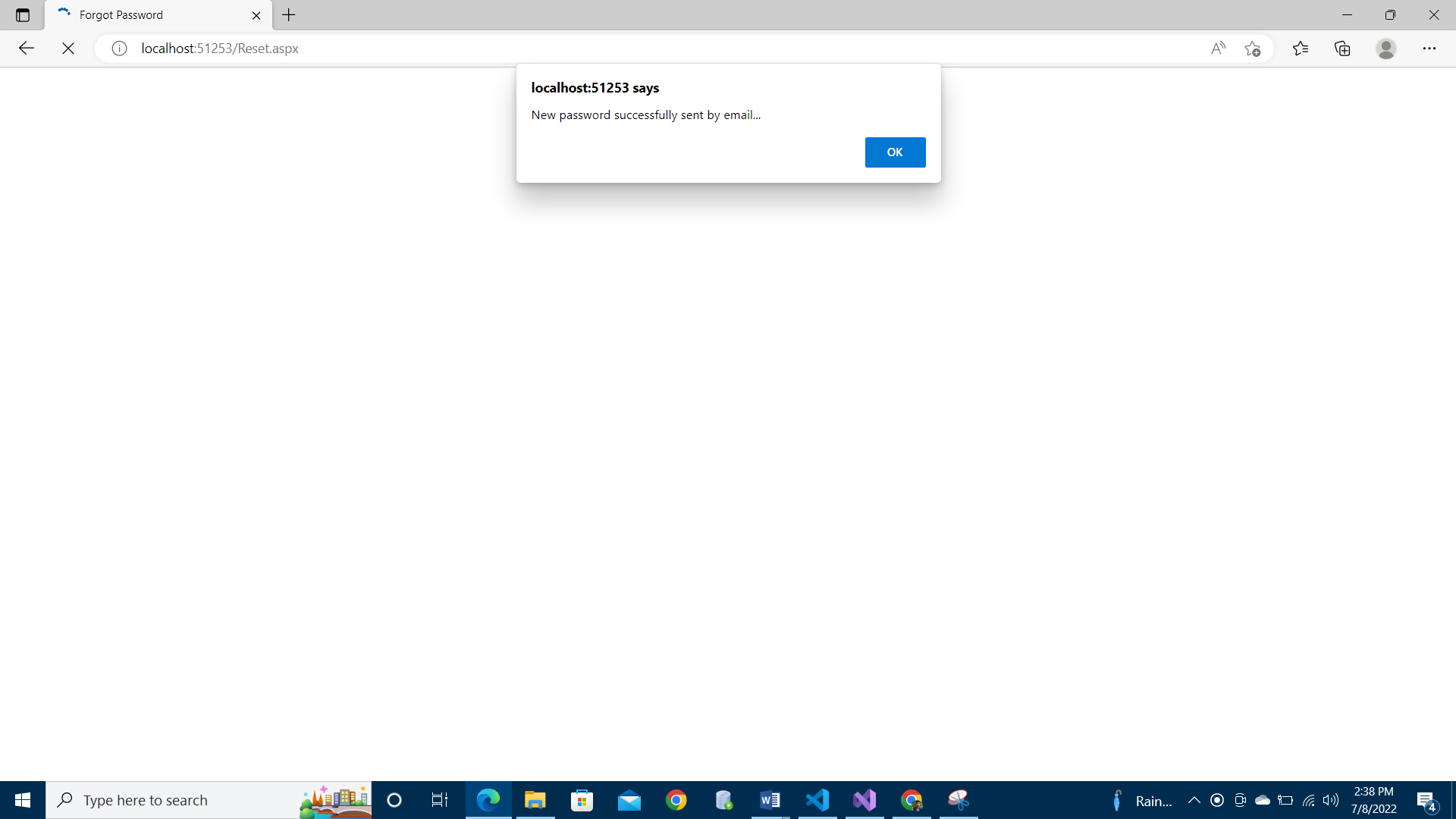


Fig18: Password Reset Email (reset.aspx)

**Reset Form (Reset.aspx.cs) Shows:** Check your email for validity and obtain your user ID. The email address is correct. Create a new password and encrypt it. Get Id if your email address is legitimate. Save the new password to the database. Delete the user's old password and replace it in the temporary password database. User should be emailed a new password. Display the sending dialog and the error dialog.

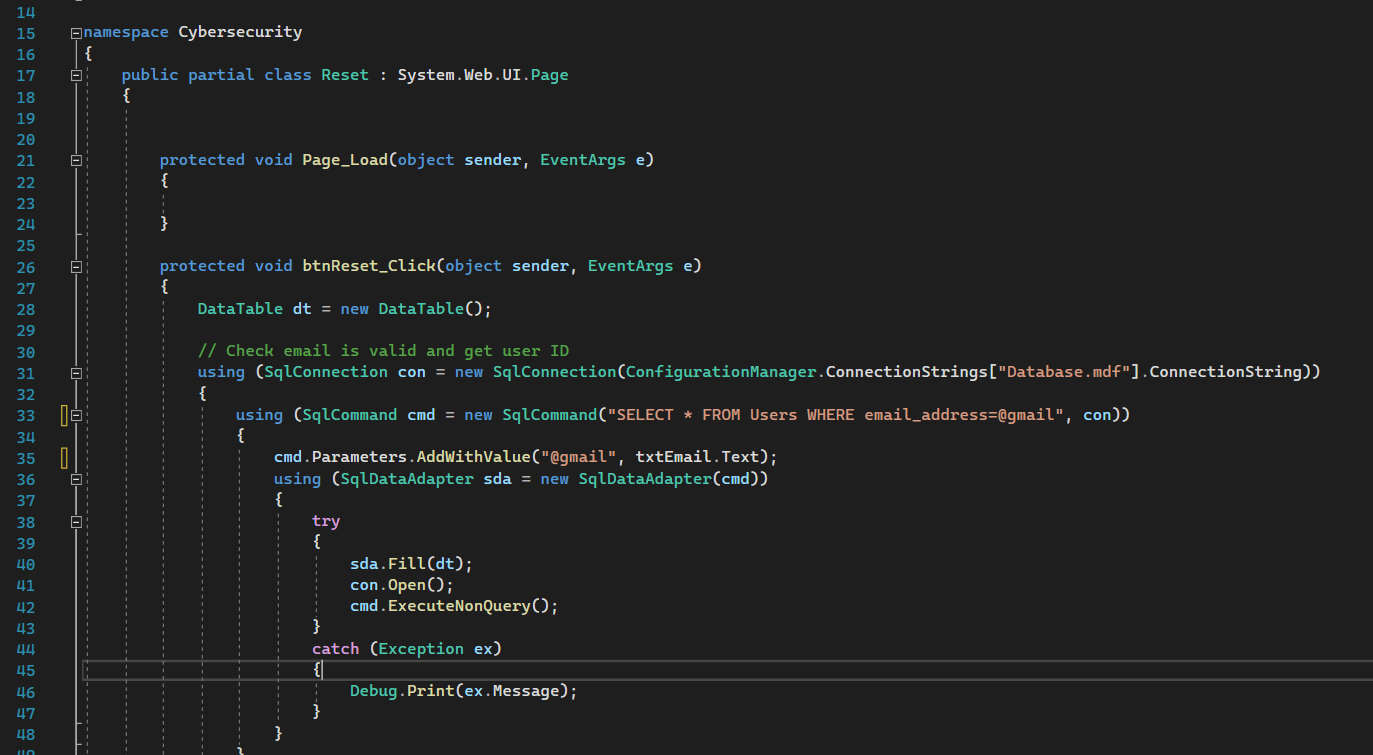


Fig19: Validate email and send mail

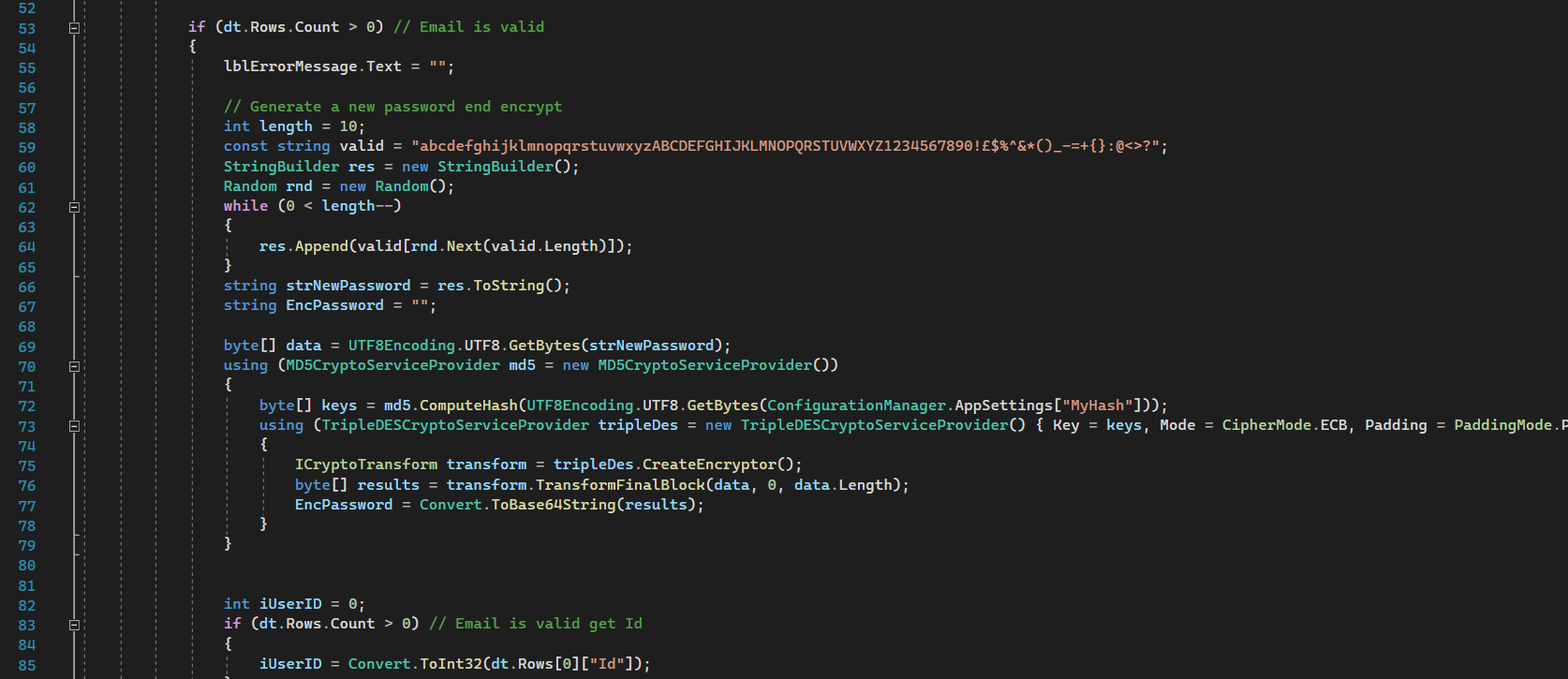


Fig20: Login.aspx.cs(server side code)

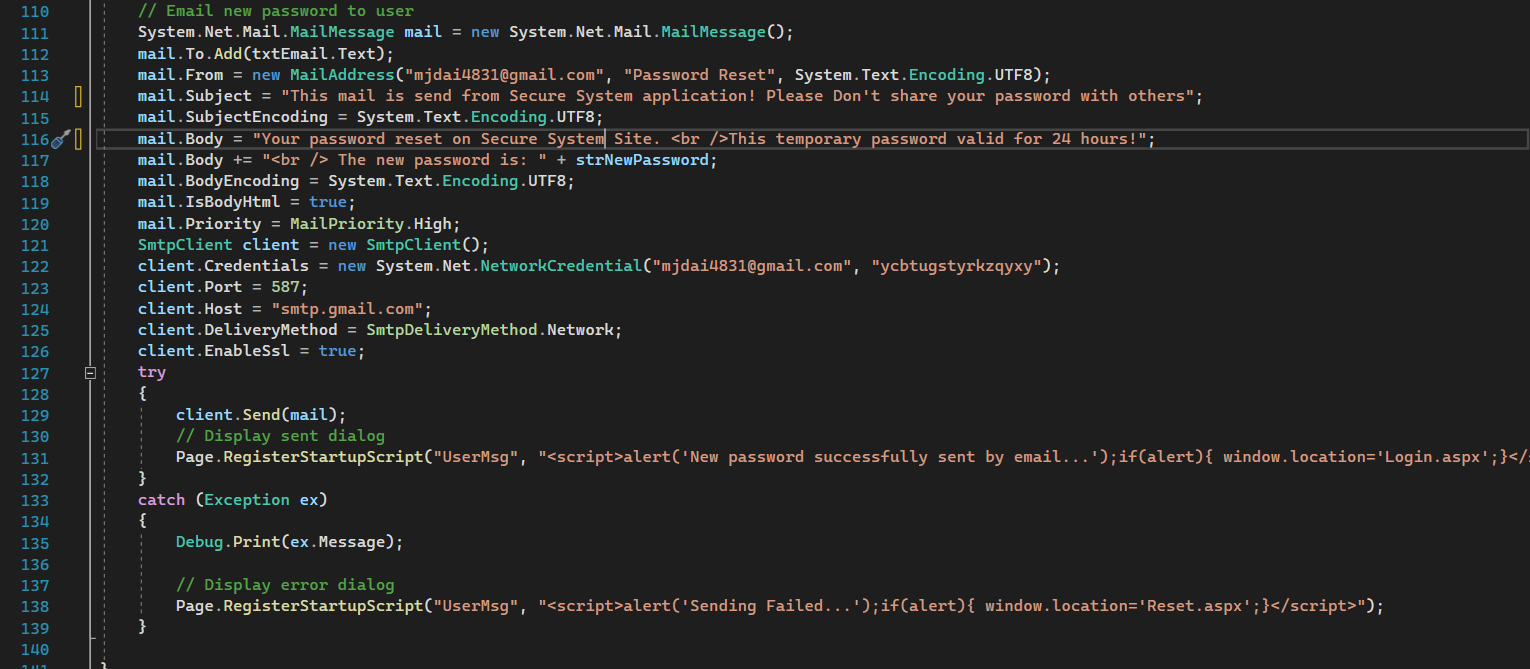


Fig21: Login.aspx.cs(server side code)

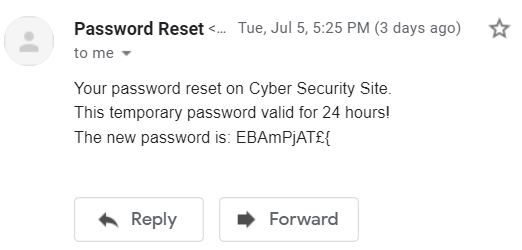


Fig22: Mail sent from system to reset password