Flow Chart Of Handling User Authentication

Start

↓

User chooses 'Login' or 'Register'

↓

If 'Register' is chosen:

Collect user registration information (e.g., username, email, password)

Validate input data (e.g., check for valid email format, strong password)

Hash the password securely

Store user data in the database

Display 'Registration Successful' message

↳ End

If 'Login' is chosen:

Collect user login credentials (e.g., username/email, password)

Retrieve hashed password from the database based on the provided username/email

Hash the entered password and compare with the stored hashed password

If the hashes match:

Generate a session token/cookie

Create a session on the server and associate it with the user

Display 'Login Successful' message

↳ End

Else:

Check for maximum login attempts exceeded

If yes, lock the account

Display 'Login Failed' message

↳ End

While the user is logged in:

Handle user interactions and session management

↳ End

User chooses 'Logout':

Invalidate the session token/cookie

Destroy the session on the server

Display 'Logout Successful' message

↳ End

User forgets password:

Provide a 'Forgot Password' option

Collect the user's email to send a password reset link

Generate a secure password reset token and store it with an expiration timestamp

Send the reset link to the user's email

↳ End

User clicks on the password reset link:

Verify the token and its expiration time

If valid:

Allow the user to reset the password

Update the password in the database after securely hashing it

Display 'Password Reset Successful' message

↳ End

Else:

Display 'Password Reset Link Expired' message

↳ End

End