📌 1. Introduction to the Login System

A login system allows users to:

✅ Create an account (Register)

✅ Log in to an existing account

✅ Reset their password if forgotten

✅ View all registered users (for admin purposes)

This system stores user credentials in a file (users.txt), ensuring persistence across multiple sessions.

==========

📌 2. System Components and Workflow

🔹 1️⃣ User Registration (register_user())

🔍 Purpose: Allows new users to sign up by providing an email and a strong password.

🔧 How it Works:

The system asks for an email and validates it using is_valid_email().

The system asks for a password and checks its strength using is_valid_password().

If both are valid, the credentials are stored in users.txt.

✅ The user receives a "Registration successful!" message.

🚨 Error Handling:

If the email is invalid → ❌ "Invalid email format!"

If the password is weak → ❌ "Password must contain at least one uppercase letter, one special character, and be at least 6 characters long."

==========

🔹 2️⃣ User Login (login_user())

🔍 Purpose: Allows registered users to log in by verifying their credentials.

🔧 How it Works:

The system asks for an email and password.

It checks if the email exists in users.txt.

If the email exists:

✅ If the password matches → "Login successful!"

❌ If the password is incorrect → "Incorrect password!"

If the email does not exist → ❌ "User not found!" → Redirects to registration.

🚨 Error Handling:

If the email is not found → The user is redirected to register_user().

If the password is incorrect → Displays a message but does not allow access.

==========

🔹 3️⃣ Forgot Password (forgot_password())

🎯 Purpose: Helps users reset their password if they forget it.

🔧 How it Works:

The system asks for the user's email.

It checks if the email exists in users.txt.

If found, the user is prompted to enter a new password.

The system validates the new password and updates it in users.txt.

✅ "Password reset successful!" message is shown.

🚨 Error Handling:

If the email is not found → ❌ "User not found!" → Redirects to registration.

If the new password is weak → ❌ "Password must meet the security criteria."

==========

🔹 4️⃣ Retrieve Registered Users (retrieve_users())

🔍 Purpose: Displays a list of all registered users.

🔧 How it Works:

The system reads users.txt.

Extracts and displays only the registered email addresses (not passwords).

If no users are found → ❌ "No users found in the system."

🚨 Error Handling:

If users.txt is empty, an appropriate message is displayed.

==========

🔹 5️⃣ Redirect to Registration (redirect_to_registration())

🔍 Purpose: If a user is not found during login or password reset, they are automatically redirected to registration.

🔧 How it Works:

Calls the register_user() function, ensuring that unregistered users can create an account immediately.

==========

🔹 6️⃣ Validation Functions

These functions ensure input correctness and increase security.

✅ is_valid_email(email)

Uses regex to check if the email follows a proper format.

✅ is_valid_password(password)

Ensures passwords are secure by checking:

✅ At least 6 characters

✅ At least one uppercase letter

✅ At least one special character

==========

📌 3. Complete System Flow

💡 Welcome to the System!

1️⃣ Register → Calls register_user()

2️⃣ Login → Calls login_user()

3️⃣ Forgot Password → Calls forgot_password()

4️⃣ Retrieve Users → Calls retrieve_users()

5️⃣ Exit → Terminates the program

🔹 If the user selects **Login**:

- If credentials are correct → ✅ Login successful

- If email exists but password is wrong → ❌ Incorrect password!

- If email does not exist → ❌ "User not found!" → Redirects to Register

🔹 If the user selects **Forgot Password**:

- If email exists → ✅ Allows password reset

- If email does not exist → ❌ Redirects to Register

🔹 If the user selects **Retrieve Users**:

- Displays registered emails

🔹 If the user selects **Exit**:

- Ends the program

==========

📌 4. Why is This System Important?

A secure and user-friendly login system is essential for any online platform. This system:

- ✅ Ensures user authentication
- ✅ Protects against unauthorized access
- ✅ Provides a smooth user experience
- ✅ Stores data securely in a file

By implementing error handling, validations, and redirections, this system ensures a seamless and secure login process.