Project 2

Python Banking Application Project

## Objective

Your task is to develop a simple Python application that functions as a banking system for a single user. This application will manage account and balance information, featuring a user-friendly interface for various banking operations. Key elements include password encryption for security, basic account transactions, and a straightforward navigation menu.

## Detailed Requirements

1. **Password Encryption:**

* **Functionality**: Implement a function named encrypt\_password that applies a Caesar cipher encryption to the user's password, shifting each letter (uppercase and lowercase) and digit by three positions forward. Characters that wrap around (e.g., 'z' to 'c', '9' to '2') should be correctly handled, while non-alphanumeric characters remain unchanged.
* **Technical Details**: Utilize ASCII values for encryption, adapting the formula (m + 3 - base) % range + base for different character sets (uppercase, lowercase, digits), where m is the ASCII value of the character, base is the starting ASCII value of the character set, and range is the size of the character set (26 for letters, 10 for digits).

1. **Account Transactions:**

* **Deposits**: Develop a function deposit\_amount that increases the account's balance by a specified amount. It should accept a floating-point number as input and print the updated balance.
* **Withdrawals**: Create a function withdraw\_amount to decrease the account's balance by a specified amount, provided there are sufficient funds. It should print the new balance or a warning message in case of insufficient funds.

1. **Account Information and Authentication:**

* **Data Storage**: Use a Python dictionary to store the user's ID, encrypted password, and current balance. This ensures quick access and manipulation of the user's data.
* **Login System**: Implement a function login\_user that verifies the user's credentials (ID and password) against stored values. Successful authentication allows access to further services.

1. **User Interface:**

* **Menu System**: Design a loop that displays a menu with options for logging in, depositing, withdrawing, viewing balance, changing password, and logging out. Handle invalid inputs gracefully and include a hidden option for displaying account details for debugging purposes.

1. **Input Handling:**

* **Initial Setup**: On application start, prompt the user to set up a new account by entering a user ID, password (which gets encrypted before storage), and an initial balance.
* **Transaction Inputs**: Ensure the application gracefully handles all user inputs, including validation of menu choices and transaction amounts.

1. **Output Formatting:**

* **Financial Figures**: Format all monetary values to two decimal places when displayed to ensure clarity and consistency in financial transactions.

1. **Documentation:**

* **Code Comments**: Your code should be well-documented, including a brief description of the program, annotations for each function and major block of code, and explanations for the purpose of variables.

## Project Constraints:

* Adhere strictly to best practices and concepts discussed in class.
* Avoid using any Python features or libraries marked as off-limits for this course.
* Ensure your application functions correctly across a variety of test cases to confirm reliability.
* Name your source code file project2.py.

Deliverables

* The Python source code file (project2.py).
* A PDF document detailing your algorithm and design approach.
* **Note:** This project is an individual effort; ensure all submitted work is entirely your own. Your submission will be scrutinized for originality to maintain academic integrity.

## Sample output:

Welcome to the Simple Banking System!

Enter new user ID: johndoe123  
Enter password: password123  
Enter initial balance: $1000.00

Account created successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MENU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Select a service:   
1. Login to Account                             
2. Deposit to Account                           
3. Withdraw from Account                        
4. Print Balance                                
5. Change Password                              
6. Exit/Logout                                  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter choice> 9  
Account details are hidden from public view.

Enter choice> 1  
Enter user ID: johndoe123  
Enter password: password123  
User johndoe123 logged in successfully.

Enter choice> 2  
Enter deposit amount: $500.00  
New balance: $1500.00

Enter choice> 3  
Enter withdrawal amount: $200.00  
New balance: $1300.00

Enter choice> 4  
johndoe123 balance: $1300.00

Enter choice> 5  
Enter new password: newPassword123  
Password updated successfully.

Enter choice> 6  
User johndoe123 logged out of the account.  
Thank you for using the Simple Banking System!