

**CLASS XII**

**ATM PROJECT**

NAME: **ARYA MANDAL**

CLASS: **12K**

ROLL NO.: **18**

ADMISSION NUMBER: **2023/14235**

**ACKNOWLEDGEMENT**

In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with project.

Primarily I would thank god for being able to complete this project with success. Then I would like to thank myComputer Teacher Dhrita Ma’am, whose valuable guidance has been the ones that helped me patch this project and make it full proof success his suggestions and his instructions has served as the major contributor towards the completion of the project.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance has been helpful in various phases of the completion of the project.

Last but not the least I would like to thank my classmates who have helped me a lot.

**OBJECTIVE OF THE PROPOSED SYSTEM:**

The objective of this ATM system is to simulate basic banking operations, allowing users to perform tasks such as checking account balances, depositing funds, withdrawing money, and viewing transaction histories. The system will ensure secure authentication of users through PIN validation, and transactions will be recorded in a database for security and tracking purposes.

**INPUT AND OUPUT OF THE PROPOSED SYSTEM:**

**Input:**

User inputs include login credentials (Account Number, PIN), transaction requests (deposit, withdraw, balance inquiry) and other banking commands.

**Output:**

Outputs include confirmation of transactions, account balance information, updated transaction history, and error messages in case of invalid inputs or unsuccessful transactions.

**FUNCTIONS AND FEATURES:**

* Secure login with account number and PIN
* View balance
* Withdraw and deposit funds
* Transaction history
* PIN change functionality
* Error handling for invalid inputs

**FRONT – END AND BACK-END:**

**Front-end:**

Python (Terminal-based or graphical using libraries like Tkinter for GUI)

**Back-end:**

SQL for database management (storing user data, transaction logs, etc.)

**HARDWARE AND SOFTWARE USED:**

**Hardware:**

Standard PC or server setup

**Software:**

* Python (3.12 64-bit)
* SQLite/MySQL for database management
* Optional: Tkinter

**SCOPE AND LIMITATIONS:**

**Scope:**

Simulating basic ATM operations (authentication, transactions, and balance inquiry) in a secure environment. Useful for educational purposes or small-scale banking simulations.

**Limitations:**

Limited to basic ATM functionalities. No real-time online banking features.

It won’t connect to actual financial institutions or networks.

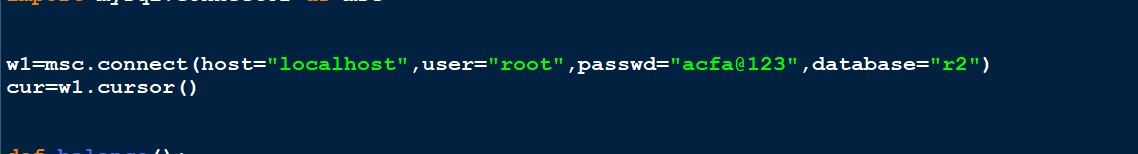
Security features are basic and may not include encryption or advanced fraud detection mechanisms.

**Let’s start with the code…**

At the beginning we need to import some modules,



Then, we connect with our database name as “r2” here,

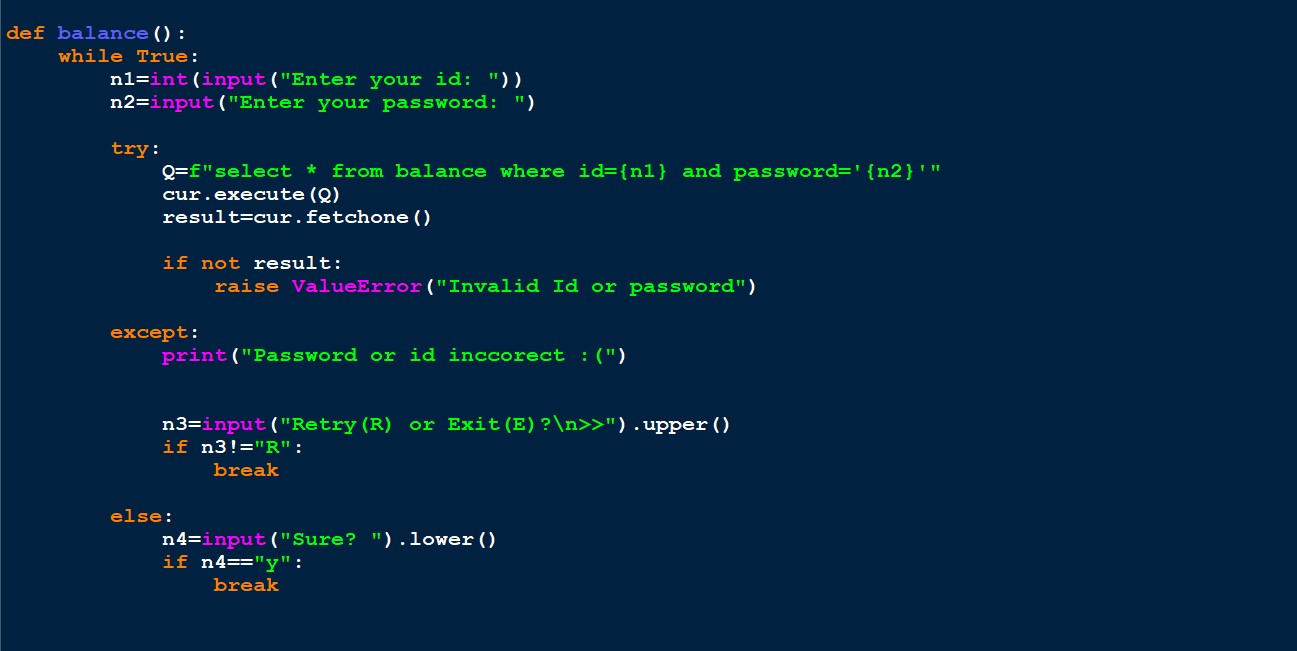


Now, for the main part of the program, we need to import five functions. Each function plays a critical role in ensuring the program operates efficiently. These functions are well-structured and modular, making the code easier to understand and maintain. Once imported, they will be integrated into the core logic to handle specific tasks. This approach not only enhances reusability but also simplifies debugging and testing.

# Balance Function:-

The **balance()** function allows a user to check their account balance by validating their ID and password. It repeatedly prompts the user to input their credentials and queries the database for a matching record. If the credentials are invalid, an error message is displayed, and the user can choose to retry or exit. Once valid credentials are provided and confirmed by the user, the loop ends, and the account details are printed. This ensures secure access and provides user-friendly error handling.

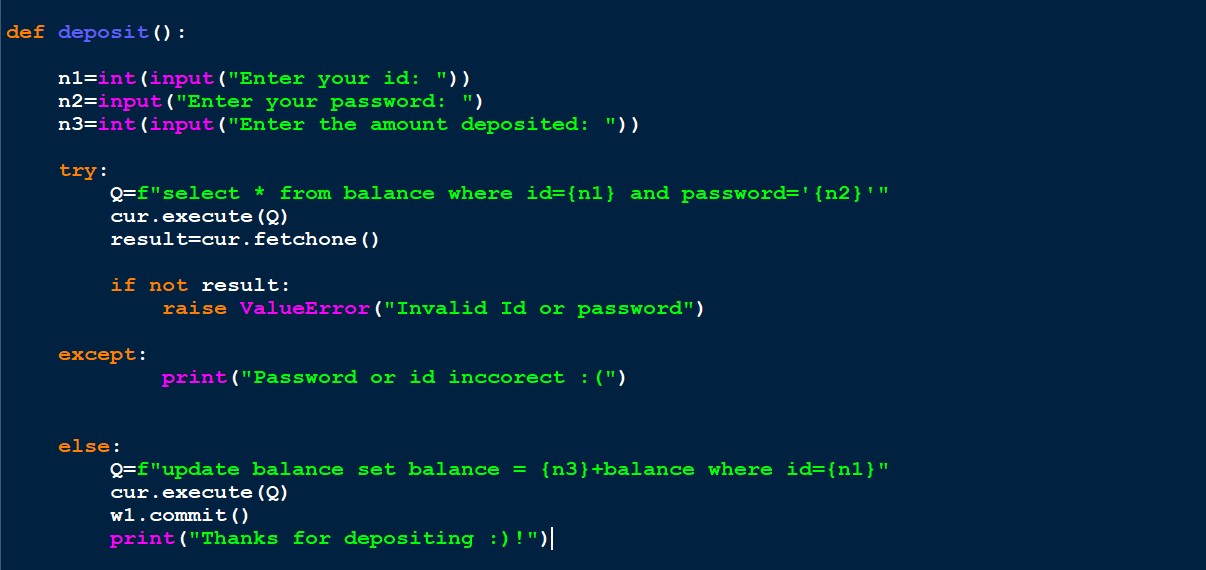
Here is the code for the following,



# Deposit Function:-

The **deposit()** function allows a user to add money to their account. It first prompts the user to input their ID, password, and the amount to deposit. Then, it queries the database to verify the credentials. If the credentials are invalid, an error message is displayed. If the credentials are valid, the function updates the account balance in the database by adding the deposited amount to the existing balance. Finally, it commits the transaction and displays a confirmation message thanking the user for their deposit.

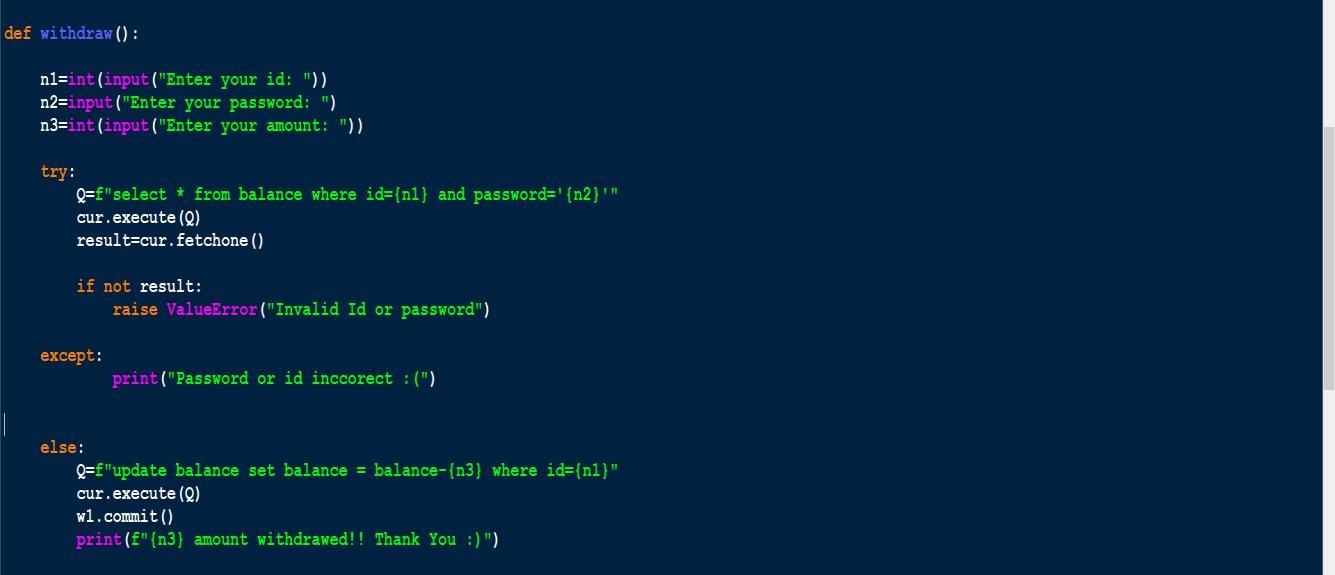
Here is the code for the following,



# Withdraw Function:-

This code handles the withdrawal of funds from a user's account. It first prompts the user for their ID, password, and the amount they wish to withdraw. It then queries the database to verify the credentials. If the credentials are invalid, an error message is displayed. If valid, the code deducts the specified amount **(n3)** from the user's balance in the database, updates the balance, commits the transaction, and prints a confirmation message indicating the amount withdrawn.

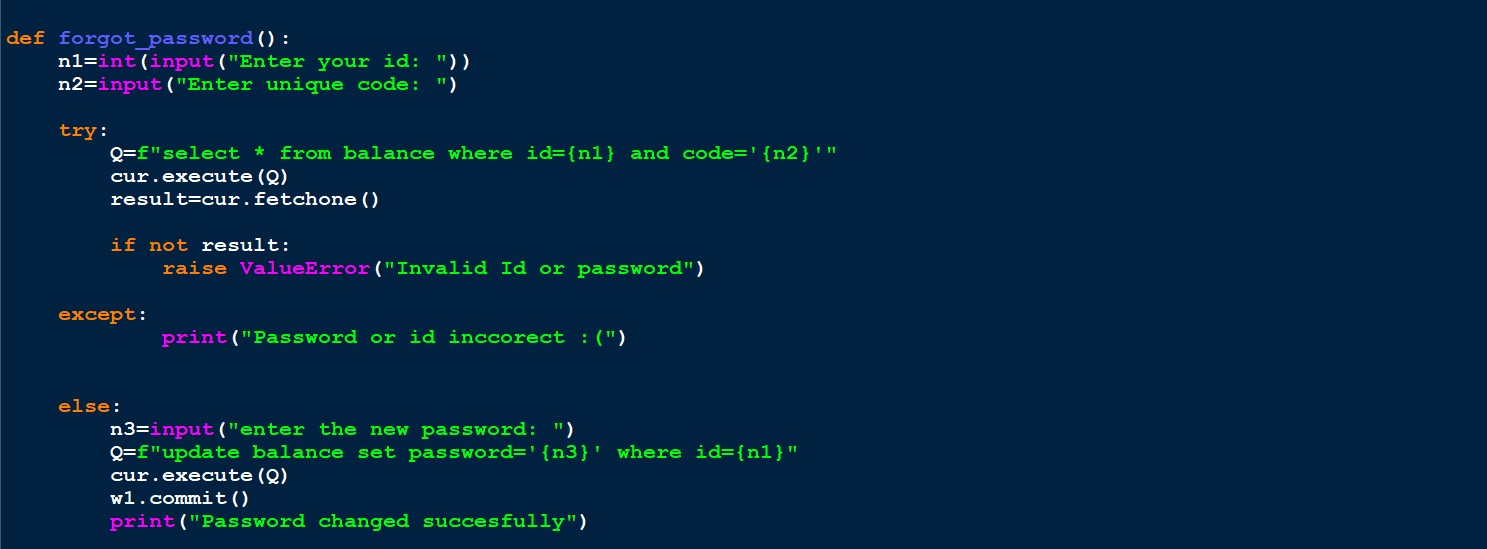
Here is the code for the following,



# Forgot password:-

The **forgot\_password()** function helps users reset their password securely. The user is prompted to input their ID and a unique code (used as an additional layer of verification). The function then queries the database to validate the provided ID and unique code. If the credentials are invalid, an error message is displayed. If valid, the user is prompted to input a new password, which is then updated in the database. Finally, the function commits the changes and confirms the password reset with a success message.

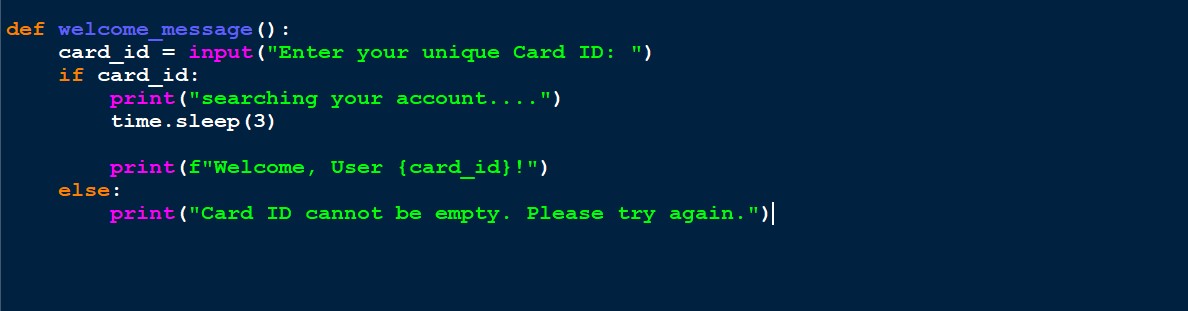
Here is the code for the following,



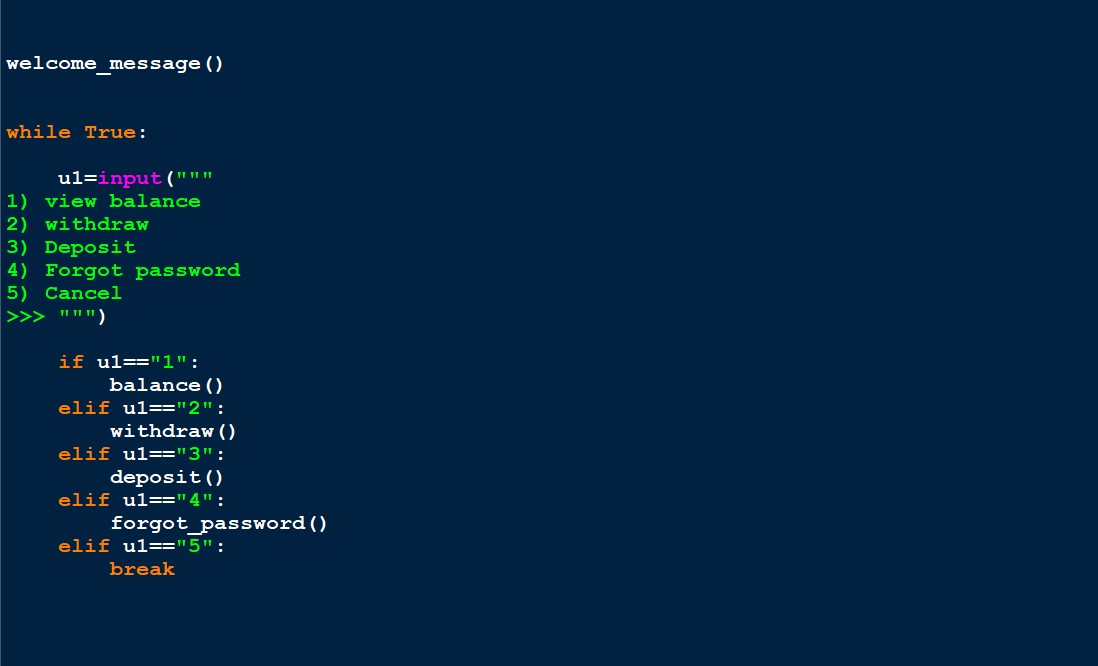
# Welcome Function:-

The **welcome\_message()** function greets a user based on their unique Card ID. It first prompts the user to input their Card ID. If a Card ID is provided, it simulates searching for the account by printing a message and pausing for 3 seconds. Afterward, it welcomes the user by displaying their Card ID. If the Card ID is not provided, it displays an error message indicating that the field cannot be empty and prompts the user to try again. This function ensures a user-friendly and engaging start to the interaction.

Here is the code for the following,



Finally we compile all of them for the main body of the program as followed,



I would like to express my heartfelt gratitude to my computer teacher for providing us with the invaluable opportunity and freedom to explore new avenues in developing this computer project. Their guidance and encouragement have been instrumental in shaping this work. I am also deeply thankful to CBSE for offering a platform that allows us to channel our creativity and showcase our potential. A special thanks to my friends for their unwavering support and cooperation, which made this journey both enriching and enjoyable. Most importantly, I am profoundly grateful to my parents, whose constant guidance, patience, and encouragement have been the cornerstone of my efforts. This project stands as a testament to their belief in my abilities and their unconditional support.

# *CERTIFICATE*

***This is hereby to certify that the original and genuine investigation work has been carried out to investigate about the subject matter and the related data collection and investigation has been completed solely, sincerely and satisfactorily by ARYA MANDAL XII-K for the academic year 2024-25.***

***TEACHER SIGN***

***(ALAK KUMAR GARU SIR)***

# *CERTIFICATE*

***This is hereby to certify that the original and genuine investigation work has been carried out to investigate about the subject matter and the related data collection and investigation has been completed solely, sincerely and satisfactorily by ARYA MANDAL XII-K for the academic year 2024-25.***

***TEACHER SIGN***

***(DHRITA MA’AM)***