100+ Years of Glorious history inscribed in the yeomen service to the field of education

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

Estd. 2017
ಸ್ಥಾಪನೆ : 2017

Poojya Dr. Sharnbaswappa Appaji
Mahadashi Peethadhipathi
Sharnbasveshwar Samsthana
President,Sharnbasveshwar Vidya Vardhak Sangha
Chancellor, Sharnbasva University

Poojya Matoshri Dr. Dakshayini S. Appa
Chairperson,
Sharnbasveshwar Vidya Vardhak Sangha
Member of BOG Sharnbasva University

Poojya Chiranjeevi Doddappa Appa
Mahadasohi 9th Peethadipatigalu
Sharnbasveshwar Samsthana, Kalaburgi

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ
Sharnbasva University

Poojya Matoshri Godutai Avvaji

Poojya Doddappa Appa
Founder President
Sharnbasveshwar Vidya Vardhak Sangha

Kalaburagi - 585103, Karnataka - India
ಕಲಬುರಗಿ 585 103 ಕರ್ನಾಟಕ - ಭಾರತ

Phone / Fax No. 08472-277852, 277853, 277854, 277855 www.sharnbasvauniversity.edu.in - email : Sharnbasvauniversity@gmail.com

**PROJECT ON:**

# ATM USING PYTHON LANGUAGE

**Under the Guidance**

**Of:**

PROF. MANJULA.BIRADAR

**Submitted By:**

| | |
|---|---|
| SHRIHARI | SG21CSE137 |
| SHUBHAM | SG21CSE141 |
| SHRINIDHI | SG21CSE138 |
| VIJAY BUDDHI | SG21CSE170 |

## TABLE OF CONTENTS:

## ❖ ABSTRACT:

This application allows the customers to collect cash ,know the balance, account status, credited and debited status. It allows authorized users to access the system by entering a valid pin number. This ensures security and prevents unauthorized access. It also enables to make transaction from the account at any time of the day. ATM programs connects to banking networks to process transactions and update account balances.

## ❖ INTRODUCTION:

An **Automated Teller Machine** is an electronic telecommunications device that enables customers to perform financial transactions, such as cash withdrawals, deposits, balance inquiries at any time and without the need for direct interaction with bank staff . Using an ATM customers can access their bank account in order to make a variety of most notably cash withdrawals and balance checking. Customers are typically identified by inserting a plastic ATM card (or some other acceptable payment card) into the ATM, with authentication being by the customer entering a personal identification.

### ❖ Working of ATM:

The project is based upon using python language. In this application an  programmer uses python language to identify the code used in ATM machine, the project shows the following details when the user accesses his/her account by applying the PIN which he or she has and the code which will be confidential to only user. After updating the PIN the user can credit/debit the money, know the accounts.

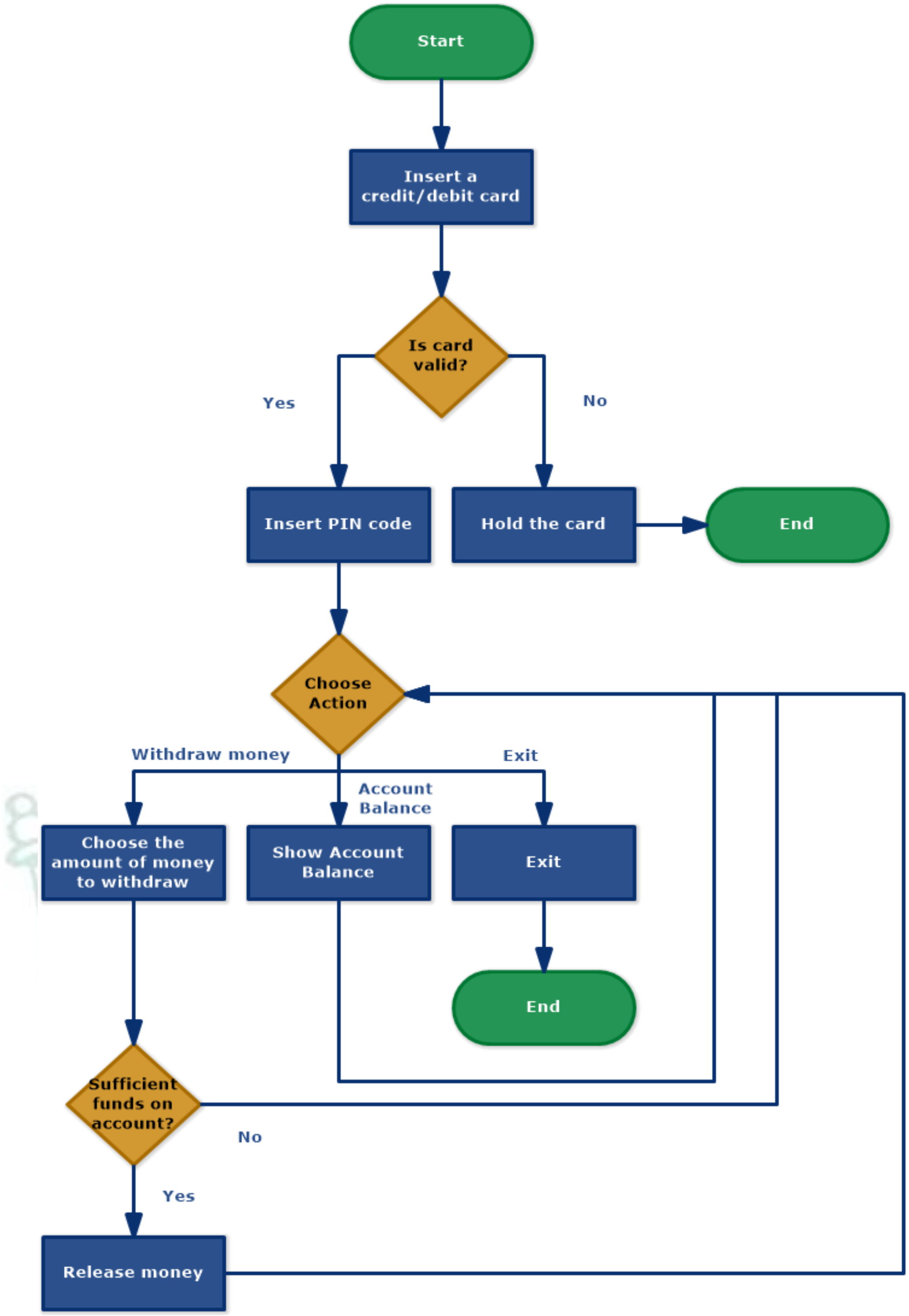
## ❖ IMPLEMENTATION:

```
import json


class ATM:
  def _init_(self, user_data_file):
    self.user_data_file = user_data_file
    self.load_user_data()

  def load_user_data(self):
    try:
      with open(self.user_data_file, "r") as file:
        self.users = json.load(file)
    except FileNotFoundError:
      self.users = {}

  def save_user_data(self):
    with open(self.user_data_file, "w") as file:
      json.dump(self.users, file, indent=4)

  def create_account(self, username, pin):
    if username not in self.users:
      self.users[username] = {"pin": pin, "balance": 0, "credit_limit": 0, "transactions": []}
      self.save_user_data()
      return "Account created successfully."
    else:
      return "Username already exists. Please choose a different username."
```

```
def authenticate_user(self, username, pin):
    if username in self.users and self.users[username]["pin"] == pin:
        return True
    else:
        return False

def check_balance(self, username):
    return self.users[username]["balance"]

def deposit(self, username, amount):
    if amount > 0:
        self.users[username]["balance"] += amount
        self.users[username]["transactions"].append(f"Deposited ${amount}")
        self.save_user_data()
        return f"Deposited ${amount}. New balance: ${self.users[username]['balance']}"
    else:
        return "Invalid amount for deposit."

def withdraw(self, username, amount):
    if amount > 0 and amount <= self.users[username]["balance"]:
        self.users[username]["balance"] -= amount
        self.users[username]["transactions"].append(f"Withdrew ${amount}")
        self.save_user_data()
        return f"Withdrew ${amount}. New balance: ${self.users[username]['balance']}"
    else:
        return "Insufficient funds or invalid amount for withdrawal."
    def mini_statement(self, username):
        return self.users[username]["transactions"][-5:]


def main():
    user_data_file = "user_data.json"
    atm = ATM(user_data_file)

    while True:
        print("1. Create Account")
        print("2. Log In")
        print("3. Exit")

        choice = input("Select an option: ")

        if choice == "1":
            username = input("Enter a username: ")
            pin = input("Enter a PIN: ")
            print(atm.create_account(username, pin))
        elif choice == "2":
            username = input("Enter your username: ")
            pin = input("Enter your PIN: ")
            if atm.authenticate_user(username, pin):
                while True:
                    print("1. Check Balance")
                    print("2. Deposit")
                    print("3. Withdraw")
                    print("4. Mini Statement")
                    print("5. Log Out")

                    user_choice = input("Select an option: ")

                    if user_choice == "1":
                        print("Your balance is:", atm.check_balance(username))
                    elif user_choice == "2":
                        amount = float(input("Enter the amount to deposit: "))
                        print(atm.deposit(username, amount))
                    elif user_choice == "3":
                        amount = float(input("Enter the amount to withdraw: "))
                        print(atm.withdraw(username, amount))
                    elif user_choice == "4":
                        statement = atm.mini_statement(username)
                        print("Mini Statement:")
                        for transaction in statement:
                            print(transaction)
                    elif user_choice == "5":
                        print("Logging out.")
                        break
                    else:
                        print("Invalid choice. Please select a valid option.")
            else:
                print("Authentication failed. Invalid username or PIN.")
        elif choice == "3":
            print("Exiting the ATM. Have a nice day!")
```

```
            break
        else:
            print("Invalid choice. Please select a valid option.")

if _name_ == "_main_":
    main()
```

## ❖ FLOWCHART:


Start
Insert a credit/debit card
Is card valid?
Yes
No
Insert PIN code
Hold the card
End
Choose Action
Withdraw money
Account Balance
Exit
Choose the amount of money to withdraw
Show Account Balance
Exit
End
Sufficient funds on account?
No
Yes
Release money

## ❖ HARDWARE REQUIREMENTS:

The hardware requirements for running a python-based ATM machine are minimal. Since it is a simple console-based program, it can be run on any computer that supports python. The basic requirements are:

Processor: Any processor that supports Python

RAM: At least 512MB RAM

Storage: At least 200MB free storage space

Overall, the hardware requirements are very low, and almost any modern computer or laptop can easily run a Python-based ATM machine without any issues.

## ❖ SOFTWARE REQUIREMENTS:

Python Interpreter: You need the Python interpreter installed on your system to run Python code. The code you provided appears to be written in Python 3. To run it, you should have Python 3 installed.

Text Editor or Integrated Development Environment (IDE): You can write and edit Python code in any text editor, but using an IDE specifically designed for Python can provide additional features like code completion, debugging tools, and project management.

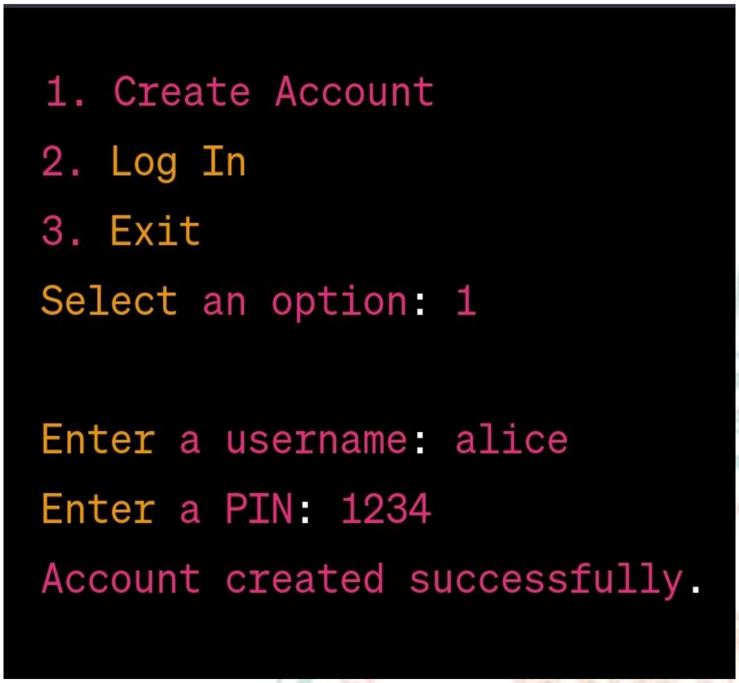
## ❖ ADVANTAGES:

- Users can create new accounts with unique user name and pin
- Users can perform transactions without the need to visit bank physically
- Users can conduct transactions in private without need for direct interaction with bank staff
- Automated transactions reduce operational cost for both user and bank compared to manual processes
- The ATM system allows users to access and performs transactions on their accounts at any time

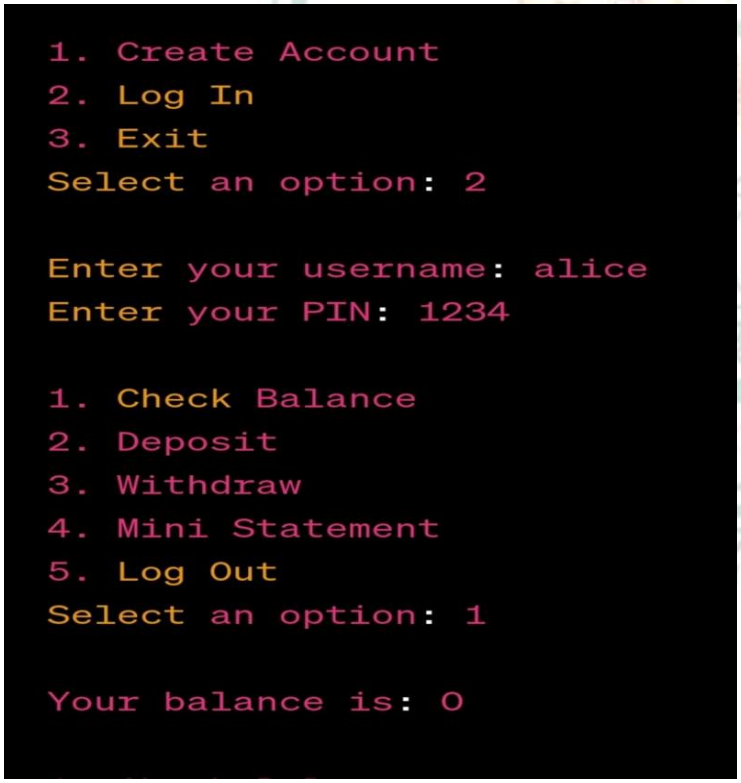
## ❖ DISADVANTAGES:

- Technical issues can occur preventing transactions or causing error that may temporarily
- We might face currency conversion fees or additional fees with international transaction
- If not properly secured ,the system can be vulnerable to data breaches and cyberattack leading to financial loss
- Poor data management might expose simulated user data, raising privacy issues.
- if system is not mentioned properly,there is a risk of data loss due to hardware or software failure

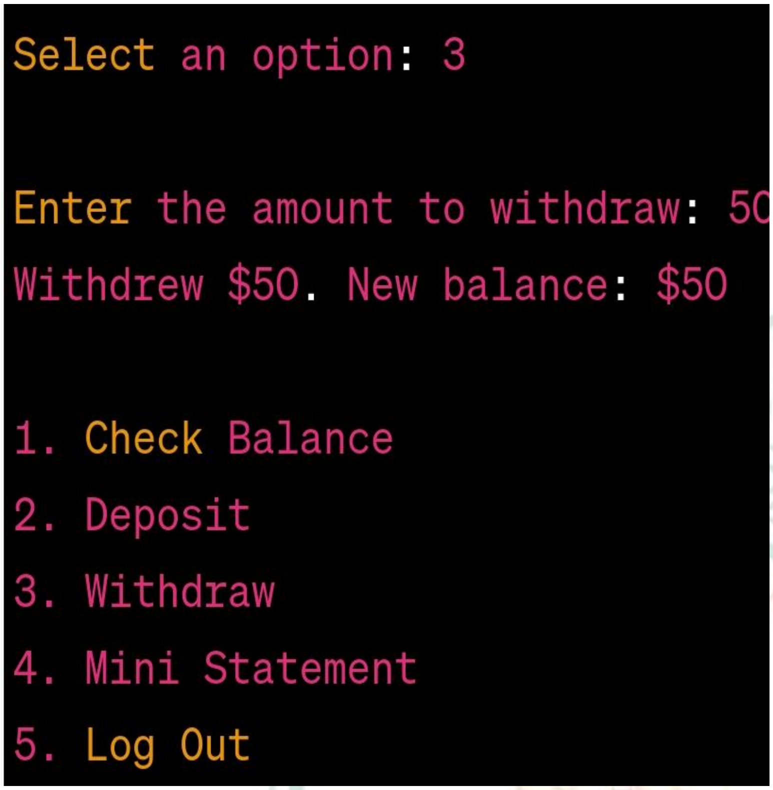
## ❖ OUTPUT:



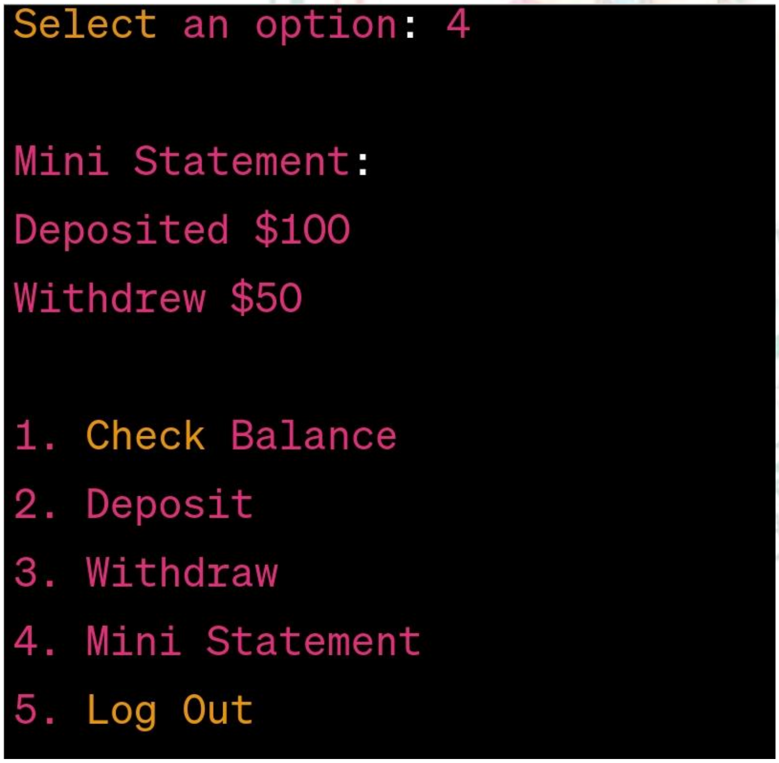
```
1. Create Account
2. Log In
3. Exit
Select an option: 1

Enter a username: alice
Enter a PIN: 1234
Account created successfully.
```



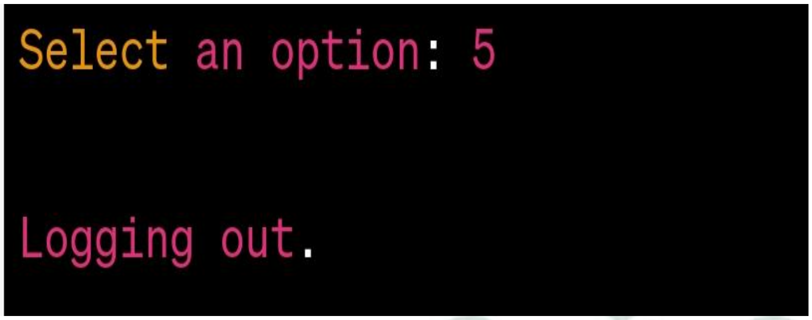
```
1. Create Account
2. Log In
3. Exit
Select an option: 2

Enter your username: alice
Enter your PIN: 1234

1. Check Balance
2. Deposit
3. Withdraw
4. Mini Statement
5. Log Out
Select an option: 1

Your balance is: 0
```



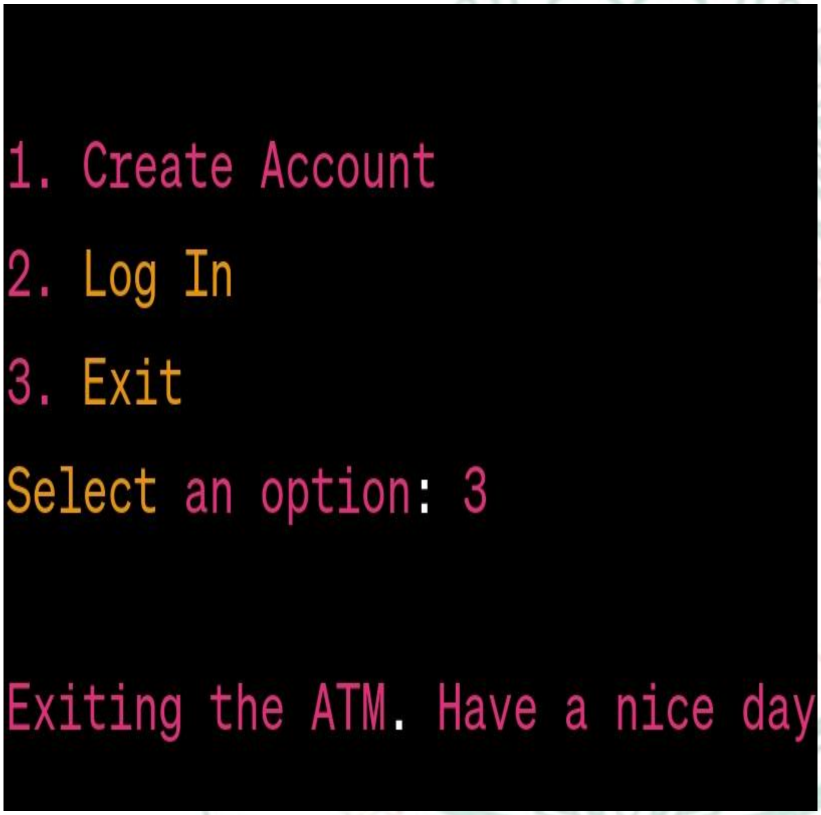
```
Select an option: 3

Enter the amount to withdraw: 50
Withdrew $50. New balance: $50

1. Check Balance
2. Deposit
3. Withdraw
4. Mini Statement
5. Log Out
```



```
Select an option: 4

Mini Statement:
Deposited $100
Withdrew $50

1. Check Balance
2. Deposit
3. Withdraw
4. Mini Statement
5. Log Out
```



```
Select an option: 5


Logging out.
```



```
1. Create Account
2. Log In
3. Exit
Select an option: 3


Exiting the ATM. Have a nice day
```

## ❖ FUTURE WORK:

User Authentication Enhancement: Implement a more secure authentication mechanism, such as using hashed passwords and salting for PINs instead of storing them in plain text.

GUI Implementation: Create a graphical user interface (GUI) for the bank account management system using libraries like Tkinter or PyQt. This can enhance the user experience and make the application more user-friendly.

## ❖ CONCLUSION:

In conclusion, the ATM project successfully demonstrates the design and implementation of a functional Automated Teller Machine system. Through this project, important concepts of software development and security measures were explored. The project's achievement lies in its ability to simulate real-world ATM operations, providing users with a seamless and secure banking experience. Overall, the project serves as a valuable learning experience, showcasing the integration of various technologies to create an efficient and user-friendly system.

## ❖ REFERENCES:

1. https:/www.github.com/apna-college/
2. Automate the boring stuff using python
3. https:/www.chat.openai.com/