Visvesvaraya National Institute of Technology (VNIT), Nagpur

Embedded System (ECL 403)

Project Report

Submitted by :
Name: Chinmay Vijay Patil (BT20ECE074)
Semester 5

Submitted to :
Dr.Ankit Bhurane and Dr.Amit Agarwal
(Course Instructors)
Department of Electronics and Communication Engineering,
VNIT Nagpur

Aim:: Designing ATM machine interface using telegram bot and ESP-32.

Requirement::

1. ESP-32 kit
2. telegram App
3. Arduino IDE

Theory:: The task is to create a ATM inteface using telegram bot and creating a database in google spreadsheet. The ESP-32 is connected to hotspot of mobile phone for which the ssid and password is given. The ESP-32 here act as a station to send the data to the cloud. The google spreadsheet stores the data is feed from ESP32 into data base according to instructions. Google appscript is used as a link between ESP-32 and spreadsheet in which a code has to be written for reading a value or writing into some cell or updating any value in a cell. The task is to update any balance after transaction and to update the transaction history that is handled by the update and get update summary function in code. The interface starts with 5 options to user to :

1. Create a new Account.
2. Withdraw Money.
3. Credit Money.
4. Balance Check.
5. Transaction History.

The google appscript code and Arduino IDE are interlinked. Google appscrpt needs a URL from Arduino IDE or ESP-32 that has all parameter such as username,password and command and there values which are processed by google script using e.parameter command and according to instructions specified such as read, write , update, summary are executed. Creating account needs Write user function , withdraw needs read data function and update function same thing goes for credit. Simultaneously while withdrawing and crediting the transaction script is changed according that can be viewed in google spreadsheet.

Everytime the user tries to give command on first command it has to submit its unique ID and password which is verified and only then the balance is updated. This whole process can be viewed from telegram screen as esp sends telegram the message specified for the instruction specified. After the command is executed the function returns and the process starts again after inputting start/

Code In Google AppScript::

```
var sheet_id="1_mA3xwDgPM93mbqGr7lle9oR1QcLzKxraeTXwqZcfQk";
var sheet_name="BT20ECE074_ESP";

var ss=SpreadsheetApp.openById(sheet_id);
var sheet=ss.getSheetByName(sheet_name);

function doGet(e)
{
      var  unique_ id=sheet.getDataRange().getLastRow()+1;

      var command=String(e.parameter.command);
      if(command=="read")
      {
        return  getActiveValue(e.parameter.cell _location);
      }
      if(command=="write _user")
      {
         write_username(unique _id,e.parameter.user _name,e.parameter.password);
         return ContentService.createTextOutput(unique_id);

      }
      if(command=="update_balance")
      {
        setActiveValue(e.parameter.cell location, ...
            e.parameter.update balance);



      }
      if(command=="update_summary")
```

```
       {
         setActiveValue(e.parameter.cell_location, ...
             e.parameter.transaction_history);

       }


}

 function  write username(unique_ id,user_ name,password)
{

       sheet.appendRow([unique  _id,user _ name,password,15000]);


}


function getActiveValue(read) {
  // Retrieve and return the information requested by the sidebar.


  return ...
      ContentService.createTextOutput(sheet.getRange(read).getValue());
}

function  setActiveValue(cell -location,value)  {
  // Use data collected from sidebar to manipulate the sheet.

  sheet.getRange(cell-location).setValue(value);
}
```

Code in ESP32:

```

#include <WiFi.h>  //for wifi connect
#include <WiFiClient.h>
#include <UniversalTelegramBot.h>   // for telegram
#include <ArduinoJson.h> // for telegram
#include <HTTPClient.h> //for information transfer over url

// Replace with your network credentials
const char* ssid = "motorola";
const char* password = "motorola";
String ...
    GOOGLE _ SCRIPT_ ID="AKfycbxqKOLE_ 5zlsbcKoZbLqNx7t0rfenNbAf7H7LUR3S7e6zSt9u9LTlU2k


```

```
#define BOTtoken "5647527330:AAHvsIGSxLxNx8WHACr7Ms35j4UYnNLMzGs"
 #define CHAT_ID "687022702"


WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

int botRequestDelay = 1000;
unsigned long lastTimeBotRan;



 String write_into_Spreadsheet(String user name, String password  )
{

   if(WiFi.status()==WL _CONNECTED)
   {

          String urlFinal = ...
              "https://script.google.com/macros/s/"+GOOGLE_SCRIPT_ID+"/exec?user_name
          HTTPClient http;
          Serial.println("Making a request");

      http.begin(urlFinal.c str()); //Specify the URL and certificate
      http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
      int httpCode = http.GET();

      String give_take;
      if (httpCode ==200) { //Check for the returning code
        give _take  = http.getString();
        Serial.println(httpCode);
        Serial.println(give_take);
      }
      else {
        Serial.println("Error on HTTP request");
      }
      http.end();
      return give_take;

   }


}


 String read_data(String unique _id, String parameter) // ...
     unique _id is taken from user and parameter is set according ...
     to requiremt
{
    if(parameter=="unique id") // assigning column A to unique ...
      id and so on
    {
     parameter="A";
    }

    else if(parameter=="user_name")
    {
      parameter="B";
    }
    else if(parameter=="password")
    {
      parameter="C";
    }
    else if(parameter=="available_balance")
    {
      parameter="D";
    }
    else if(parameter=="transaction_history")
    {
     parameter="E";
    }
    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;
      String loc=parameter+unique_id;
      String url = "https://script.google.com/macros/s/"+ ...
        GOOGLE_SCRIPT_ID + ...
        "/exec?cell_location="+loc+"&command=read";
      Serial.println("Making a request");
      http.begin(url.c_str()); //Specify the URL and certificate
      http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
      int httpCode = http.GET();
      String give_take;
    if (httpCode ==200) { //Check for the returning code
     give_take = http.getString();
     Serial.println(httpCode);
     Serial.println(give_take);
    }
    else {
     Serial.println("Error on HTTP request");
    }
    http.end();

    return give_take;
 }
 }
void update(String unique_id, String parameter) // for updating ...
   the avaialable balance
 {
  if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = "https://script.google.com/macros/s/"+ ...
            GOOGLE_SCRIPT_ID + ...
            "/exec?cell_location=D"+String(unique_id)+"&update_balance="+String(par
        Serial.println(" request is made..");
        http.begin(url.c_str()); //Specify the URL and certificate
        http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
        int httpCode = http.GET();
        Serial.print("HTTP status code");
        Serial.println(httpCode);
        http.end();
  }
}
void get_update_Summary(String unique_id ,String parameter)
{
  if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = "https://script.google.com/macros/s/"+ ...
            GOOGLE_SCRIPT_ID + ...
            "/exec?cell_location=E"+String(unique_id)+"&transaction_history="+Strin
        Serial.println("request is made..");
        http.begin(url.c_str()); //Specify the URL and certificate
        http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
        int httpCode = http.GET();
        Serial.print("HTTP status code");
        Serial.println(httpCode);
        http.end();

  }
}
String receive_from_user(){
  int ...
      numNewMessages=bot.getUpdates(bot.last_message_received+1); ...
       // taking one input from telegram from the user
  while(numNewMessages<1){
    numNewMessages=bot.getUpdates(bot.last_message_received+1);
  }
  String retVal=bot.messages[0].text;
  return retVal;
}


int telegram_writing(String text){
    String Buffer;
    String chat_id=CHAT_ID;
    if (text == "start") {  // First interface
      String Starting_Line="Welcome To ATM Machine Interface \n ...
          Please Select the option you want to proceed \n Press ...
```

```
             \"1\" for Create a account \n Press \"2\" to Withdrawl ...
             \n Press \"3\" to Credit \n Press \"4\" Balance Check ...
             \n Press \"5\" to Check transaction history";
         bot.sendMessage(chat_id,Starting_Line,"");
         Serial.println(Starting_Line);
         Buffer=receive_from_user();
         String First_comm=Buffer;

         if (First_comm == "1") { // After taking creating a ...
             account command
             bot.sendMessage(chat_id, "Please enter a User_name", ...
                 ""); // taking username
             Serial.println("Please enter a User_name ");
             String Buffer1=receive_from_user();        // saving it
             First_comm="Your User_name is "+Buffer1;
             bot.sendMessage(chat_id,First_comm, "");
             bot.sendMessage(chat_id, "Please enter a Password ", ...
                 ""); //entering password
             Serial.println("Please enter a Password ");
             String Buffer2=receive_from_user();        // receiving ...
                 and saving the password
             First_comm="Your Password is "+Buffer2;
             bot.sendMessage(chat_id,First_comm, "");
             String ...
                 uni_best=write_into_Spreadsheet(Buffer1,Buffer2); ...
                 // taking the unique _id created from appscript and ...
                 giving it to user
             uni_best="Your Unique_ID is :"+uni_best;
             bot.sendMessage(chat_id,uni_best, "");
             bot.sendMessage(chat_id,"Please keep the unique_id ...
                 very safe .....!!!" , "");
             bot.sendMessage(chat_id,"Account Created ...
                 Successfully!!" , "");
             bot.sendMessage(chat_id,"Atm Window Closing Thank You ...
                 for giving your time" , "");
             return 0;

           }

       else if(First_comm=="2")    // using unique id checking and ...
           verifying the password to withdraw the amount
       {
             bot.sendMessage(chat_id,"Please enter your unique ID " ...
                 , "");
             Buffer=receive_from_user();
             First_comm=Buffer.toInt();
             bot.sendMessage(chat_id,"Please enter your Password !! ...
                 " , "");
            String pass_code=read_data(First_comm,"password" );
```

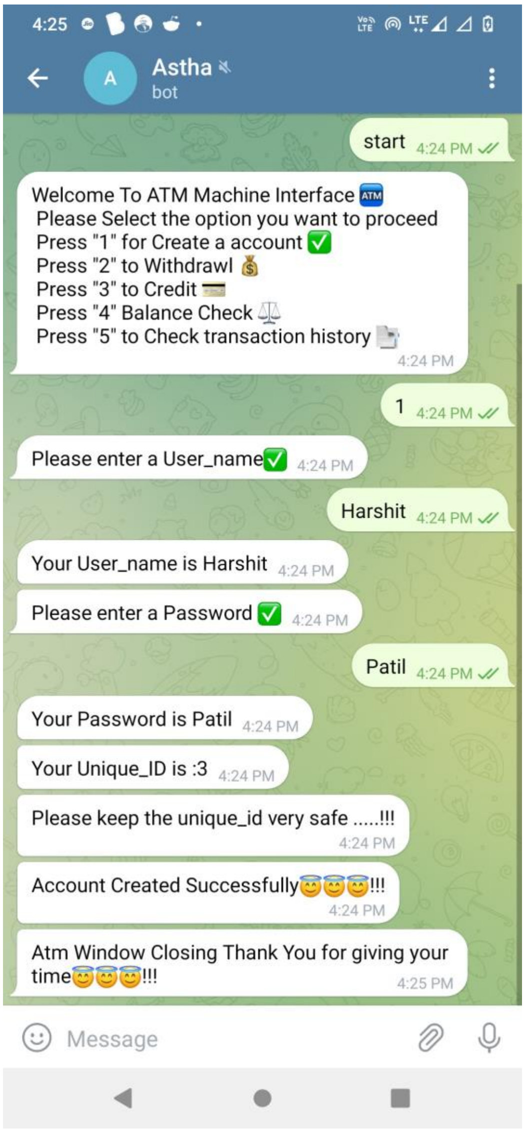
```

while(1)
{
    Buffer=receive_from_user();
    String temp=Buffer;
    if(pass_code!=temp)
    {
      bot.sendMessage(chat id,"Password is ...
          Incorrect!!!", "");
      bot.sendMessage(chat id,"Thank You for the ...
          Transaction ..... Please Try again" , "");
      return 0;
    }
    else
    {
      bot.sendMessage(chat id,"Password Verified ...
          Successfully " , "");
      delay(1000);
      break;
    }
}
point:
bot.sendMessage(chat id,"Please Enter Amount for ...
    Withdrawl! " , "");
String no_name=read_data(First_comm,"available_balance");
int read_amount=no_name.toInt();
Buffer=receive_from_user();
if(Buffer=="0")
{
  return 0;
}
if(Buffer.toInt()%100!=0)
        {
          bot.sendMessage(CHAT_ID,"Please Enter Valid ...
              Amount\nThe Amount Should be in multiple ...
              of 100","");
          bot.sendMessage(CHAT_ID,"Press '0' to ...
              exit!!","");
          goto point;
        }

int Amount=Buffer.toInt();
if(Amount >read_amount)
{

  bot.sendMessage(chat id,"Insufficient Amount ", "");
}
else{
```

```
            String ...
               read_history=read_data(First_comm,"transaction_history");
            read_history=read_history+'-'+Buffer;
            get_update_Summary(First_comm,read_history);
            Amount=read_amount-Amount;
            update(First_comm,String(Amount));
            delay(1000);
            String balance_remain="The remaining balance is: ...
               "+String(Amount);
            bot.sendMessage(chat id, balance remain , "");
          }
          bot.sendMessage(chat id,"Atm Window Closing Thank You ...
              for giving your time" , "");
          return 0;
    }
    else if(First_comm=="3")   // using unique id checking and ...
       verifying the password to withdraw the amount
    {
          bot.sendMessage(chat id,"Please enter your unique ID " ...
              , "");
          Buffer=receive_from_user();
          First_comm=Buffer;
          bot.sendMessage(chat id,"Please enter your Password!!! ...
              " , "");
          String pass_code=read_data(First_comm,"password");
          Serial.print("Your passcode is: ") ;
          Serial.println(pass_code);
          while(1)
          {
            Buffer=receive_from_user();
            String temp=Buffer;
          if(pass_code!=temp)
          {
              bot.sendMessage(chat id,"Password is ...
                 Incorrect!!!", "");
              bot.sendMessage(chat id,"Thank You for the ...
                 Transaction ..... Please Try again!! " , "");
              return 0;
          }
          else
          {
              bot.sendMessage(chat id,"Password Verified ...
                 Successfully!! " , "");
              delay(1000);
              break;
          }
          }
          pointer:
```

```
      bot.sendMessage(chat_id,"Please Enter Amount for ...
         Crediting!! " , "");
      String  no_name=read_data(First_comm,"available_balance");
      int  read_amount=no_name.toInt();
      Buffer=receive_from_user();
      if(Buffer=="0")
      {
        return 0;
      }
      if(Buffer.toInt()%100!=0) // if the number is not is ...
         not multiples of hundred taking another input
              {
                bot.sendMessage(CHAT_ID,"Please enter '0' to ...
                    exit ","");
                goto pointer;

              }
      String ...
         read_history=read_data(First_comm,"transaction_history");
      read_history=read_history+'+'+Buffer;
      get_update_Summary(First_comm,read_history); // ...
         updating transaction history

      int Amount=Buffer.toInt();
      Amount=read_amount+Amount;
      update(First_comm,String(Amount));
      delay(1000);
      String balance_remain=String(Amount);
      balance_remain="The remaining balance is : ...
         "+balance_remain;
      bot.sendMessage(chat_id, balance_remain , "");
      bot.sendMessage(chat_id,"Atm Window Closing Thank You ...
         for giving your time" , "");
      delay(100);
      return 0;
}
else if(First_comm=="4") // checking  Balance
{
  bot.sendMessage(chat_id,"Please enter your unique_id:", "");
  First_comm=receive_from_user();
  bot.sendMessage(chat_id,"Please enter your Password!!! " , ...
     "");
      String pass_code=read_data(First_comm,"password");
      Serial.print("Your passcode is: ") ;
      Serial.println(pass_code);

      while(1)
      {
        Buffer=receive_from_user();
        String temp=Buffer;
      if(pass_code!=temp)
      {
          bot.sendMessage(chat id,"Password is Incorrect!!! ...
             " , "");
          bot.sendMessage(chat id,"Thank You for the ...
             Transaction ..... Please Try again!!! " , "");

          return 0;
      }
      else
      {
          bot.sendMessage(chat id,"Password Verified ...
             Successfully! " , "");
          delay(1000);
          break;
      }
      }
  String  read_ amount=read_ data(First_ comm,"available_balance");
  read_amount="The available balance is: "+read_amount;
  bot.sendMessage(chat id,  read_amount,  "");
  bot.sendMessage(chat id,"Atm Window Closing Thank You for ...
     giving your time!" , "");
  return 0;
}
else if(First_comm=="5")  // Executing transaction Summary ...
   satement
{
  bot.sendMessage(chat id,"Please enter your unique_id:",  "");
  First_comm=receive_from_user();
   bot.sendMessage(chat id,"Please enter your Password!!!  " ...
      , "");
      String pass_code=read_data(First_comm,"password");   ...
         // taking password from the google sheet
      Serial.print("Your passcode is: ") ;
      Serial.println(pass_code);
   while(1)
      {
        Buffer=receive_from_user();
        String temp=Buffer;
      if(pass_code!=temp)  // verifing password with user
      {
          bot.sendMessage(chat id,"Password is Incorrect!!!,  ...
             "");
          bot.sendMessage(chat id,"Thank You for the ...
             Transaction ..... Please Try again! " , "");

          return 0;
      }
```

```
          else
          {
              bot.sendMessage(chat_id,"Password Verified ...
                 Successfully!! " , "");
              delay(1000);
              break;
          }
          }
      String ...
         read_amount=read_data(First_comm,"transaction history");  ...
         // reading transaction history from google sheet
      bot.sendMessage(chat_id,"The transaction done on your ...
         account were as follows \n !!!" , "");
      bot.sendMessage(chat_id, read_amount , "");
      bot.sendMessage(chat_id,"Atm Window Closing Thank You for ...
         giving your time!" , "");
      return 0;
    }
 }
}
void setup() {
  Serial.begin(115200);  //baud Rate is 115200 same is to be ...
     kept in Serial Monitor
  WiFi.mode(WIFI_STA);    // using as station
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi..");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  client.setCACert(TELEGRAM_CERTIFICATE_ROOT);// important ...
     command to start telegram information exchange
  Serial.println(WiFi.localIP());
}

void loop() {
  String text=receive_from_user(); // takes continuous input ...
     from user waiting for start
  telegram_writing(text); // calling main function
}
```

Simulation Diagrams::



start 4:24 PM

Welcome To ATM Machine Interface 🏧
Please Select the option you want to proceed
Press "1" for Create a account ✅
Press "2" to Withdrawl 💰
Press "3" to Credit 💳
Press "4" Balance Check ⚖️
Press "5" to Check transaction history 📄
4:24 PM

1 4:24 PM

Please enter a User_name✅ 4:24 PM

Harshit 4:24 PM

Your User_name is Harshit 4:24 PM

Please enter a Password ✅ 4:24 PM

Patil 4:24 PM

Your Password is Patil 4:24 PM

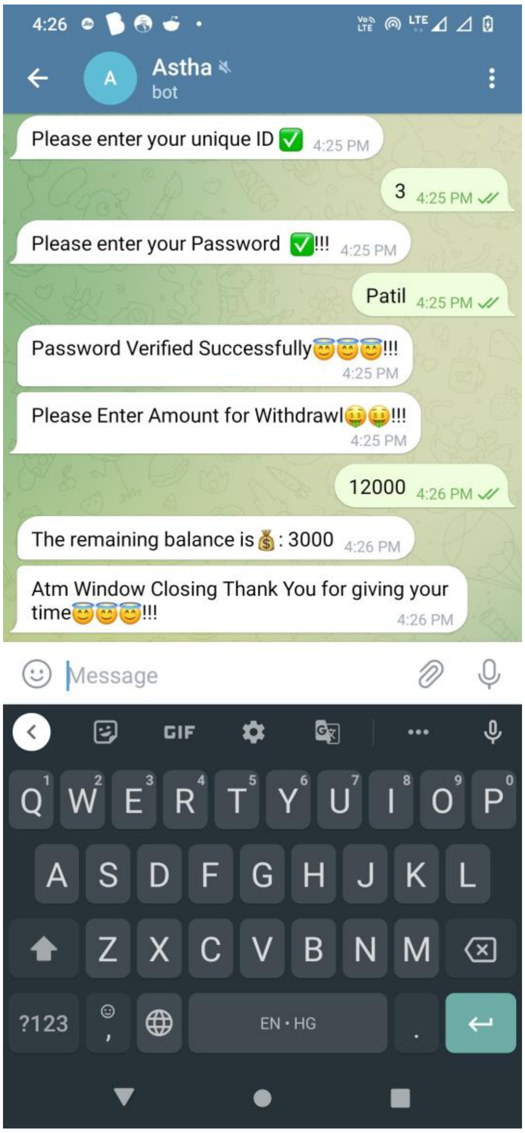
Your Unique_ID is :3 4:24 PM

Please keep the unique_id very safe .....!!! 4:24 PM

Account Created Successfully😇😇😇!!! 4:24 PM

Atm Window Closing Thank You for giving your time😇😇😇!!! 4:25 PM

Figure 1: First Interface and Creating Account



Please enter your unique ID ✅ 4:25 PM

3 4:25 PM

Please enter your Password ✅!!! 4:25 PM

Patil 4:25 PM

Password Verified Successfully😇😇😇!!! 4:25 PM

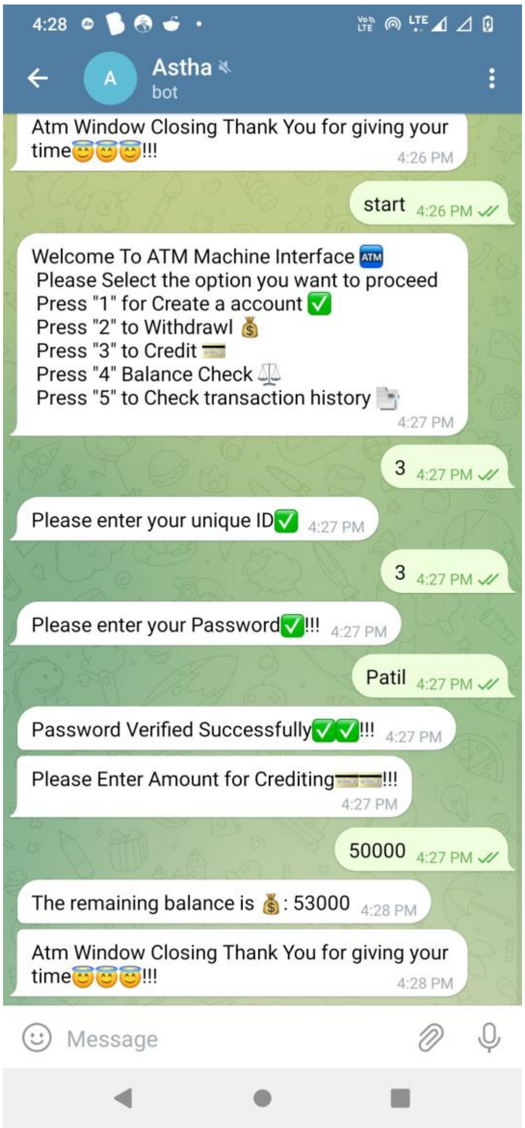
Please Enter Amount for Withdrawl🤑🤑!!! 4:25 PM

12000 4:26 PM

The remaining balance is💰: 3000 4:26 PM

Atm Window Closing Thank You for giving your time😇😇😇!!! 4:26 PM

Figure 2: Withdrawing Money Interface



Atm Window Closing Thank You for giving your time😇😇😇!!! 4:26 PM

start 4:26 PM

Welcome To ATM Machine Interface 🏧
Please Select the option you want to proceed
Press "1" for Create a account ✅
Press "2" to Withdrawl 💰
Press "3" to Credit 💳
Press "4" Balance Check ⚖️
Press "5" to Check transaction history 📑
4:27 PM

3 4:27 PM

Please enter your unique ID✅ 4:27 PM

3 4:27 PM

Please enter your Password✅!!! 4:27 PM

Patil 4:27 PM

Password Verified Successfully✅✅!!! 4:27 PM

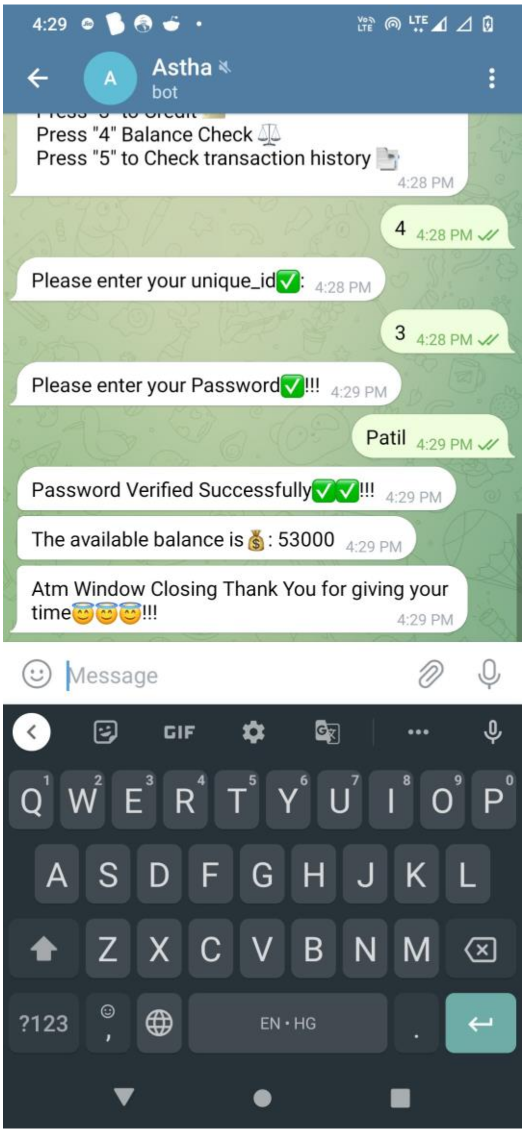
Please Enter Amount for Crediting💳💳!!! 4:27 PM

50000 4:27 PM

The remaining balance is 💰: 53000 4:28 PM

Atm Window Closing Thank You for giving your time😇😇😇!!! 4:28 PM

Figure 3: Crediting Money Interface



Press "3" to Credit
Press "4" Balance Check ⚖️
Press "5" to Check transaction history 📑
4:28 PM

4 4:28 PM ✓✓

Please enter your unique_id✅: 4:28 PM

3 4:28 PM ✓✓

Please enter your Password✅!!! 4:29 PM

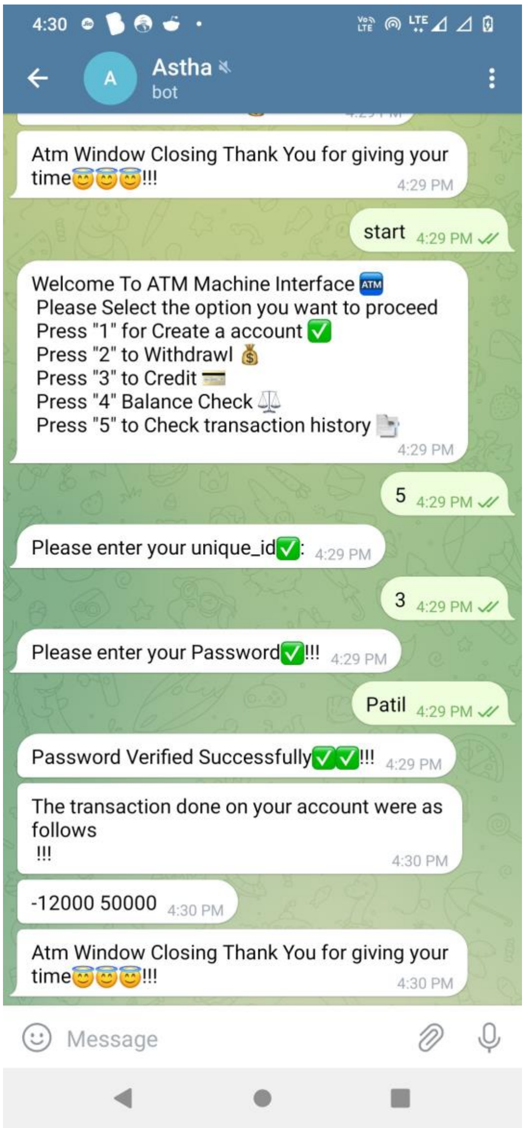
Patil 4:29 PM ✓✓

Password Verified Successfully✅✅!!! 4:29 PM

The available balance is 💰: 53000 4:29 PM

Atm Window Closing Thank You for giving your time😇😇😇!!! 4:29 PM

Figure 4: Balance Check



Atm Window Closing Thank You for giving your time😇😇😇!!! 4:29 PM

start 4:29 PM

Welcome To ATM Machine Interface 🏧
Please Select the option you want to proceed
Press "1" for Create a account ✅
Press "2" to Withdrawl 💰
Press "3" to Credit 💳
Press "4" Balance Check ⚖️
Press "5" to Check transaction history 📑
4:29 PM

5 4:29 PM

Please enter your unique_id✅: 4:29 PM

3 4:29 PM

Please enter your Password✅!!! 4:29 PM

Patil 4:29 PM

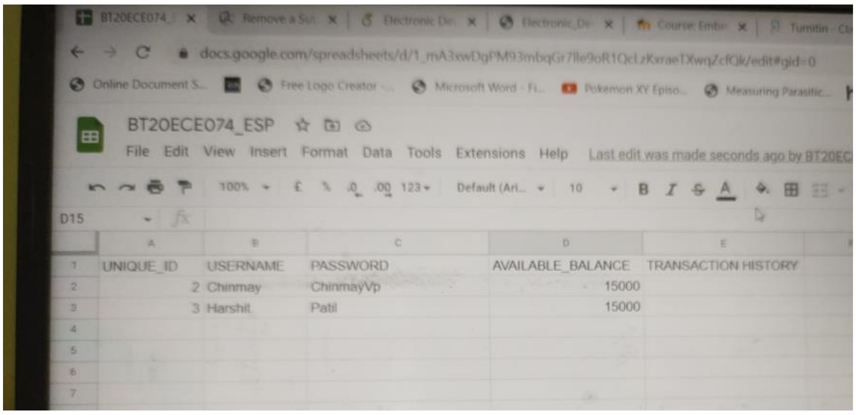
Password Verified Successfully✅✅!!! 4:29 PM

The transaction done on your account were as follows
!!!
4:30 PM

-12000 50000 4:30 PM

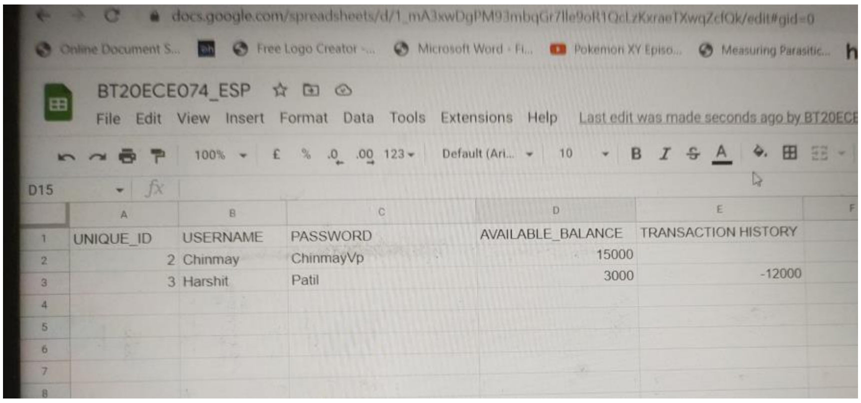
Atm Window Closing Thank You for giving your time😇😇😇!!! 4:30 PM

Figure 5: Transaction History


docs.google.com/spreadsheets/d/1_mA3xwDgPM93mbqGr7lle9oR1QcLzKxraeTXwqZcfQk/edit#gid=0
BT20ECE074_ESP
File Edit View Insert Format Data Tools Extensions Help
Last edit was made seconds ago by BT20EC
D15

	A	B	C	D	E
1	UNIQUE_ID	USERNAME	PASSWORD	AVAILABLE_BALANCE	TRANSACTION HISTORY
2	2	Chinmay	ChinmayVp	15000	
3	3	Harshit	Patil	15000	

Figure 6: Google Spreadsheet Update


docs.google.com/spreadsheets/d/1_mA3xwDgPM93mbqGr7lle9oR1QcLzKxraeTXwqZcfQk/edit#gid=0
BT20ECE074_ESP
File Edit View Insert Format Data Tools Extensions Help
Last edit was made seconds ago by BT20ECE
D15

	A	B	C	D	E
1	UNIQUE_ID	USERNAME	PASSWORD	AVAILABLE_BALANCE	TRANSACTION HISTORY
2	2	Chinmay	ChinmayVp	15000	
3	3	Harshit	Patil	3000	-12000

Figure 7: Google Spreadsheet Updating

Problem Faced if any::

1. The main problem faced was the connectivity problem of ESP 32 , it was very difficult to connect for some devices and many times the connection went out. Due to this problem there was lag in testing of the code .
2. Linking telegram also was difficult as if some command is missing from what we found on google it was difficult to identify the problem.
3. Getting the right appscript code was difficult and the appscript was new compiler so its functions are also different but the functions were found on google appscript add on files in which many functions and there uses were written like in update code in appscript required setValue command to update in cell.

Conclusion:: The ATM interface using ESP32 is designed and many functions such as Creating Account, Withdraw, Credit, Balance Check, Transaction summary are executed using telegram Bot and the values are updated in the Google Spreadsheet which were matching our results.

Video Link :

https://youtu.be/u4iUJvPQry4