Visvesvaraya National Institute of Technology (VNIT), Nagpur

Embedded System
End Sem Report

Submitted by :
Swaroop Talakwar (BT20ECE101)
Semester 5

Submitted to :
Ankit Bhurane Sir
(Course Instructor)
Department of Electronics and Communication Engineering,
VNIT Nagpur

Problem Statement: ESP32 Trainer kit on Arduino IDE is used to simulate an ATM system for checking balances and credit/debit transactions with help of google sheets.

Assumptions: The following are the assumptions that were assumed while doing this assignment:

- Opening balance in each user's amount is 15000 rupees.
- We use Google Sheets to store the usernames and passwords of each user.
- Each user has a different Ledger GoogleSheet which is accessed when the user is logged into his/her amount.
- Telegram Bot is used to take inputs from the user as well as to display the output back to the user.

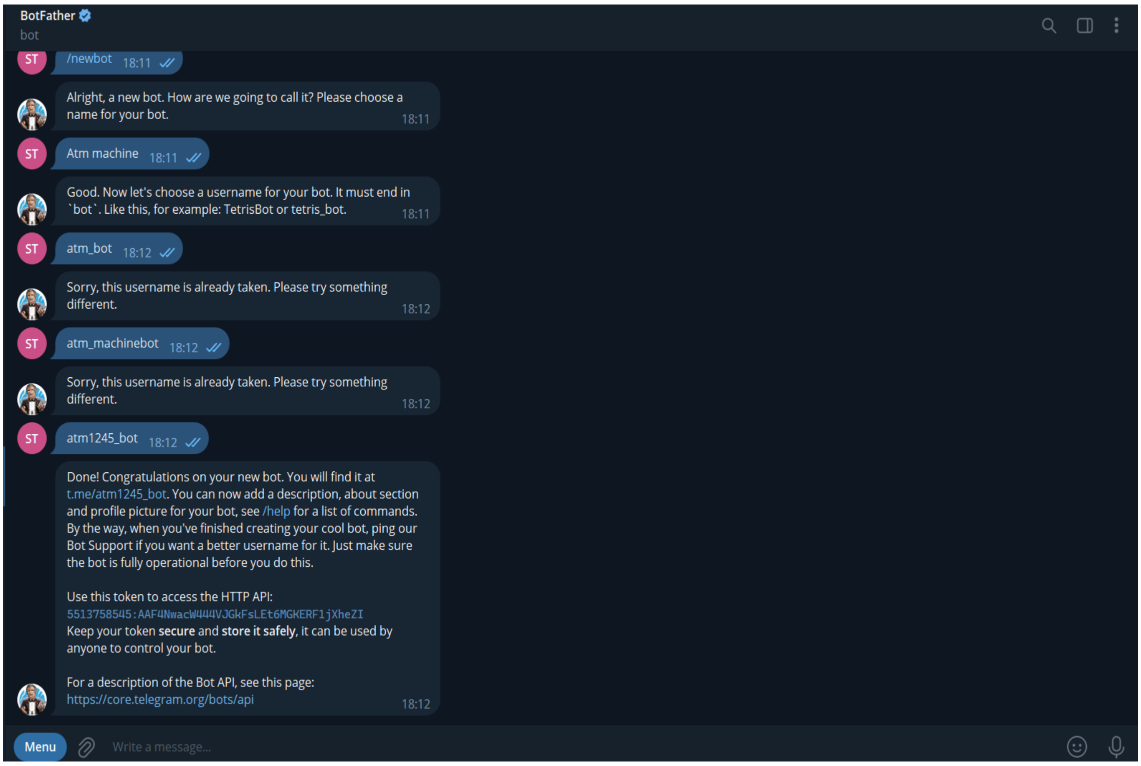
Flow of the task: The following is the flow of the task that was used to implement this project:

- Firstly, an internet connection is established between my mobile and ESP32 with the help of WiFi.h library.
- Then we repeatedly watch for inputs in the Telegram Bot. We welcome the user to our bank and ask him for his username and password which we store in two separate variables.
- Then we post this username and password to the ”Verification” google script and check if a user exists with the specified username and password and return the Ledger Script corresponding to that user back to the Arduino IDE and if there is no such user then we return a blank string.
- When the user is logged in, we provide him/her a list of commands i.e. debit, credit, statement, and logout. If the user types debit/credit, we ask the user for the amount that he/she wants to debit/credit and perform the transaction by sending this data to Ledger Script and adding a new record in the sheet, and returning the new balance.
- Else if the user selects the statement option we get the statement id with the help of the ”Verification” Script by using the Ledger script of the user as a parameter to search and return the required statement id. Then we make a get request to the Statement Id and get the entire tabular transaction data in the form of the string with we process back into the required format in the Arduino and display the data. In case of logout, We simply log the user out from his account.

Initailization Steps:

Generation of Telegram Bot

Atm1245 bot, a telegram bot for user interface, input, and output, was built with the help of BotFather.



/newbot 18:11

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 18:11

Atm machine 18:11

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 18:11

atm_bot 18:12

Sorry, this username is already taken. Please try something different. 18:12

atm_machinebot 18:12

Sorry, this username is already taken. Please try something different. 18:12

atm1245_bot 18:12

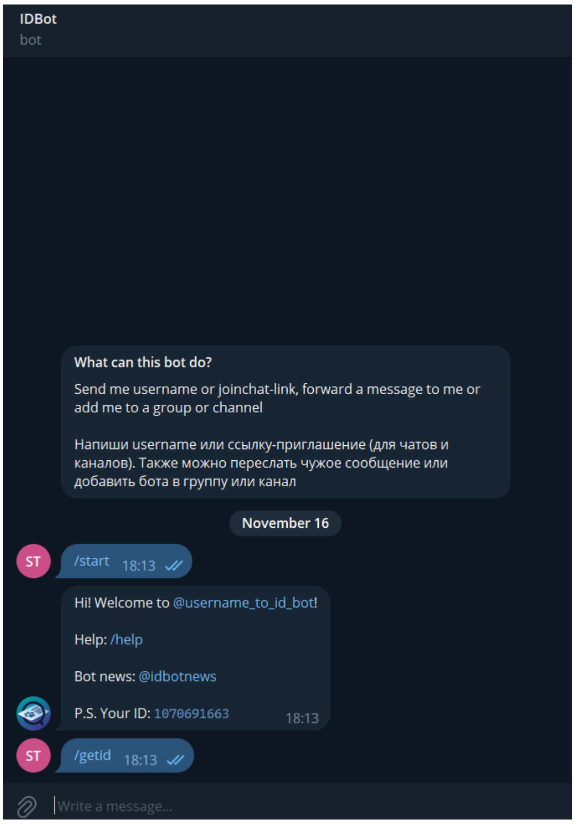
Done! Congratulations on your new bot. You will find it at t.me/atm1245_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
5513758545:AAF4NwacW444VJGkFsLEt6MGKERF1jXheZI
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 18:12

BOT created

IdBot was used to get the chat id to send a message back to the user.



IDBot
bot

What can this bot do?

Send me username or joinchat-link, forward a message to me or add me to a group or channel

Напиши username или ссылку-приглашение (для чатов и каналов). Также можно переслать чужое сообщение или добавить бота в группу или канал

November 16

ST /start 18:13

Hi! Welcome to @username_to_id_bot!

Help: /help

Bot news: @idbotnews

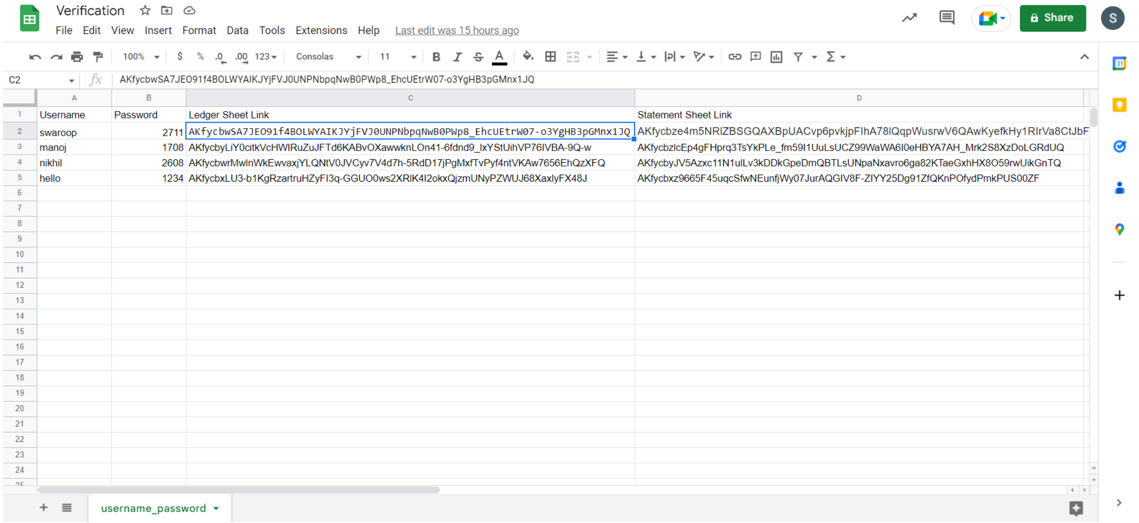
P.S. Your ID: 1070691663 18:13

ST /getid 18:13

Chat ID acquired.

‘

The following picture is of a spreadsheet that has usernames, passwords, ledger script ids, and statement script ids of each of the users.



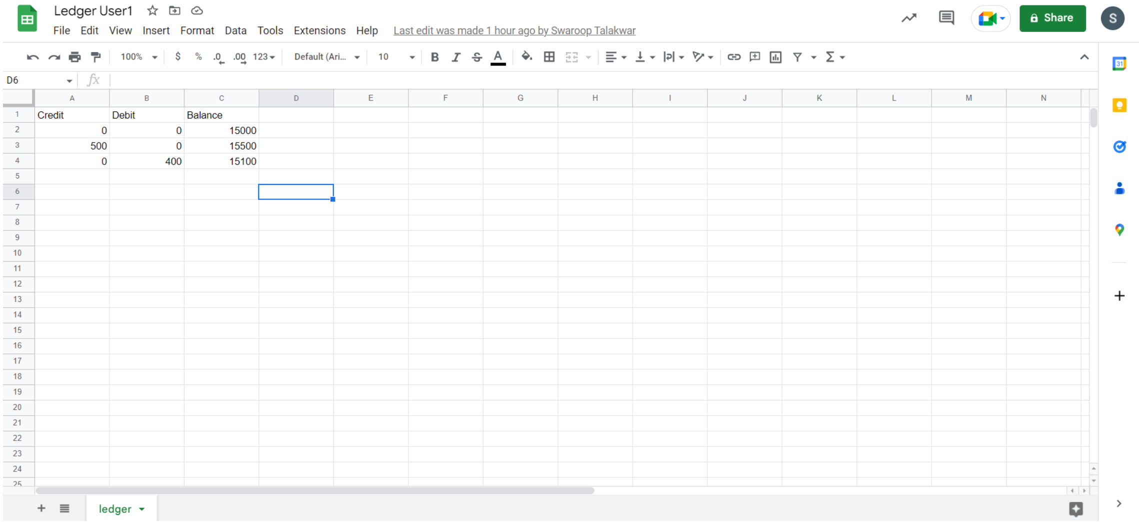
Verification
File Edit View Insert Format Data Tools Extensions Help Last edit was 15 hours ago
Share
100%
Consolas
11
username_password

C2 AKfycbwSA7JEO91f4BOLWYAIKJYjFVJ0UNPNbpqNwB0PWp8_EhcUEtrW07-o3YgHB3pGMnx1JQ

	A	B	C	D
1	Username	Password	Ledger Sheet Link	Statement Sheet Link
2	swaroop	2711	AKfycbwSA7JEO91f4BOLWYAIKJYjFVJ0UNPNbpqNwB0PWp8_EhcUEtrW07-o3YgHB3pGMnx1JQ	AKfycbze4m5NRlZBSGQAXBpUACvp6pvkjpFIhA78lQqpWusrwV6QAwKyefkHy1RIrVa8CtJbF
3	manoj	1708	AKfycbyLiY0citkVcHWIRuZuJFTd6KABvOXawwknLOn41-6fdnd9_IxYStUihVP76IVBA-9Q-w	AKfycbzlcEp4gFHprq3TsYkPLe_fm59I1UuLsUCZ99WaWA6I0eHBYA7AH_Mrk2S8XzDoLGRdUQ
4	nikhil	2608	AKfycbwrMwlnWkEwvaxjYLQNtV0JVCyv7V4d7h-5RdD17jPgMxfTvPyf4ntVKAw7656EhQzXFQ	AKfycbyJV5Azxc11N1ulLv3kDDkGpeDmQBTLsUNpaNxavro6ga82KTaeGxhHX8O59rwUikGnTQ
5	hello	1234	AKfycbxLU3-b1KgRzartruHZyFI3q-GGUO0ws2XRIK4I2okxQjzmUNyPZWUJ68XaxlyFX48J	AKfycbxz9665F45uqcSfwNEunfjWy07JurAQGIV8F-ZIYY25Dg91ZfQKnPOfydPmkPUS00ZF

Sample Verification Google Sheet

The following picture is of a sample Ledger Script of user 1 that shows how am I maintaining debit, credit, and balance records after each transaction.


Ledger User1
File Edit View Insert Format Data Tools Extensions Help
Last edit was made 1 hour ago by Swaroop Talakwar
Share
100%
Default (Ari...
10
D6
ledger

Credit	Debit	Balance
0	0	15000
500	0	15500
0	400	15100

Sample Ledger sheet for user 1

Code: Code for the task written on Arduino IDE :

```

#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>


const char* NAME__OF__WiFi = nameWifi; //Enter the name of your WiFi
const char* PASSWORD__OF__WiFi = passWifi;//Enter the password of ...
    your WiFi

const char* TOKEN__BOT=botToken;  //Enter the Bot token that is ...
    generated by the telegram.

//Our Chat Id
const char* User__CHAT__id = chatID; //Enter your Telegram Chat Id

String USER__Google_SCRIPT=userScript; //Enter your User google ...
    sheet script link
String STATEMENT__Google_Id =statementScript; //Enter your user ...
```

```
    statement sheet link

String LEDGER___Google_SCRIPT=””;

String Entered_username=””;
String Entered_password=””;

WiFiClientSecure C_l_i_e_n_T;
UniversalTelegramBot TELEGRAM_bot(TOKEN__BOT, C_l_i_e_n_T);

int bot___REQUEST_Delay = 1000;
unsigned long Last_bot_RUN_TIME;


bool RepeaT=true;
bool user___INCOMING=false;
bool password___INCOMING=false;
bool Transcation___REPEAT=true;
bool credit___INCOMING=false;
bool debit___INCOMING=false;

//Verifies user
void handle_verify(int NEW_messages) {
  for (int i=0; i<NEW_messages; i++) {
    String Obtained_User___ID = ...
        String(TELEGRAM_bot.messages[i].chat_id);

    if (Obtained_User___ID != User___CHAT___id){
      TELEGRAM_bot.sendMessage(Obtained_User___ID, ”Unauthorized ...
          user”, ””);
      continue;
    }
    String response = TELEGRAM_bot.messages[i].text;

    if (response == ”/start”) {
      String s = ”Welcome To Our Bank.\n”;
      s += ”Use the following commands to control your ...
          outputs.\n\n”;
      s += ”Please Enter Your Username:”;
      user___INCOMING=true;
      TELEGRAM_bot.sendMessage(Obtained_User___ID, s, ””);
    }else{
      if(user___INCOMING){
        Entered_username=response;
        user___INCOMING=false;
        TELEGRAM_bot.sendMessage(Obtained_User___ID,”Please Enter ...
            Your Password:”);
        password___INCOMING=true;

      }else if(password___INCOMING==true){
        Entered_password=response;
        password___INCOMING=false;


        if (WiFi.status() == WL_CONNECTED) {
          HTTPClient http;
          String ACESSED___URL = ...
             "https://script.google.com/macros/s/" + ...
             USER___Google_SCRIPT + ...
             "/exec?data1="+Entered_username+"&data2="+Entered_password;
          http.begin(ACESSED___URL.c_str());
          http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
          int HTTP_return_Code = http.GET();
          String RES;
          if (HTTP_return_Code > 0) {
            RES = http.getString();

            if(RES==""){
              TELEGRAM_bot.sendMessage(Obtained_User___ID,"Wrong ...
                 Username or Password");
              return;
            }else{
              LEDGER___Google_SCRIPT=RES;
              TELEGRAM_bot.sendMessage(Obtained_User___ID, "Correct ...
                 UserId and Password");
              RepeaT=false;
              TELEGRAM_bot.sendMessage(Obtained_User___ID, "For ...
                 credit,type /credit \n For debit,type /debit \n ...
                 For Statement,type /statement \n To logout,type ...
                 /logout");
              transcation();
            }
          }
        }
      }
    }
  }
}

void setup() {
  Serial.begin(115200);

  #ifdef ESP8266
    configTime(0, 0, "pool.ntp.org");
   C_l_i_e_n_T.setTrustAnchors(&cert);
  #endif

  WiFi.mode(WIFI_STA);
```

```
    WiFi.begin(NAME__OF__WiFi, PASSWORD__OF__WiFi);
   #ifdef ESP32
     C_l_i_e_n_T.setCACert(TELEGRAM_CERTIFICATE_ROOT);
   #endif
    while (WiFi.status() != WL_CONNECTED) {
      delay(1000);
      Serial.println("Connecting to WiFi..");
    }

    Serial.println("Connected to WiFi Successfully!");

    verification();
}


void verification() {
    while(RepeaT==true){
      if (millis() > Last_bot_RUN_TIME + bot__REQUEST_Delay)  {
        int NEW_messages = ...
            TELEGRAM_bot.getUpdates(TELEGRAM_bot.last_message_received ...
            + 1);

        while(NEW_messages) {
          handle_verify(NEW_messages);
          NEW_messages = ...
              TELEGRAM_bot.getUpdates(TELEGRAM_bot.last_message_received ...
              + 1);
        }
        Last_bot_RUN_TIME = millis();
      }
    }
}

void transcation(){
      while(Transcation__REPEAT=true){
        if (millis() > Last_bot_RUN_TIME + bot__REQUEST_Delay)  {
          int NEW_messages = ...
              TELEGRAM_bot.getUpdates(TELEGRAM_bot.last_message_received ...
              + 1);

          while(NEW_messages) {
            handle_transcation(NEW_messages);
            NEW_messages = ...
                TELEGRAM_bot.getUpdates(TELEGRAM_bot.last_message_received ...
                + 1);
          }
          Last_bot_RUN_TIME = millis();
        }
      }
```

```
}


void handle_transcation( int NEW_messages) {
  for (int i=0; i<NEW_messages; i++) {
    String Obtained_User__ID = ...
        String(TELEGRAM_bot.messages[i].chat_id);
    if (Obtained_User__ID != User__CHAT__id){
      TELEGRAM_bot.sendMessage(Obtained_User__ID, "Unauthorized ...
          user", "");
      continue;
    }
    String response = TELEGRAM_bot.messages[i].text;
    Serial.println(response);

    if (response == "/credit") {
      credit__INCOMING=true;
      TELEGRAM_bot.sendMessage(Obtained_User__ID, "Please Enter ...
          the Credit Amount:");
    }else if(response=="/debit"){
      debit__INCOMING=true;
      TELEGRAM_bot.sendMessage(Obtained_User__ID, "Please Enter ...
          the Debit Amount:");
    }else if(response=="/statement"){
      if (WiFi.status() == WL_CONNECTED) {
          HTTPClient http;
          String ACESSED__URL = ...
              "https://script.google.com/macros/s/" + ...
              STATEMENT__Google_Id + ...
              "/exec?data1="+LEDGER__Google_SCRIPT;
          http.begin(ACESSED__URL.c_str());
          http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
          int HTTP_return_Code = http.GET();
          String RES;
          if (HTTP_return_Code > 0) {
            RES = http.getString();
            ACESSED__URL="https://script.google.com/macros/s/" + ...
                RES + "/exec?read";
            http.begin(ACESSED__URL.c_str());
            http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
            HTTP_return_Code=http.GET();
            if(HTTP_return_Code>0){
              RES = http.getString();
              String t="   Credit Debit Balance \n 1.";
              int k=1;
              for(int q=0;q<RES.length();q++){
                if(RES[q]==';' && q!=RES.length()-1){
                  t+=" \n ";
                  k++;
                  t+=String(k);
                  t+=".  ";
                }else{
                  t+=RES[q];
                }
              }
              Serial.println(t);
              TELEGRAM_bot.sendMessage(Obtained_User___ID, t);
            }
          }
        }
    }else if( response=="/logout"){
      LEDGER___Google_SCRIPT="";
      TELEGRAM_bot.sendMessage(Obtained_User___ID, "Thanks for ...
         using our Bank.");
      verification();
    }else{
      if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String ACESSED___URL="";
        if(credit___INCOMING==true){
          ACESSED___URL = "https://script.google.com/macros/s/" + ...
             LEDGER___Google_SCRIPT + ...
             "/exec?data1="+response+"&data2=0";
          credit___INCOMING=false;
        }
        if(debit___INCOMING==true){
          ACESSED___URL = "https://script.google.com/macros/s/" + ...
             LEDGER___Google_SCRIPT + ...
             "/exec?data1=0&data2="+response;
          debit___INCOMING=false;
        }
        http.begin(ACESSED___URL.c_str());
        http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
        int HTTP_return_Code = http.GET();
        String RES;
        if (HTTP_return_Code > 0) {
          RES = http.getString();
          Serial.println("Current Balance:"+RES);
          TELEGRAM_bot.sendMessage(Obtained_User___ID,"Your Current ...
             Balance is "+RES);
        }
      }
    }
  }
}

void loop(){
  transcation();
```

```
}
```

App Script code for Verifcation:

```
var sheet_id = ”1VUNxakEKP9uYDrmlR4rtPA4vf3REQv8Q2jGQlL8bHLk”;
var main__Spreadsheet = Spreadsheet App.openBy Id(sheet_id);
var sheet = main__Spreadsheet .getSheetBy Name('username_password');

function doGet(e){
  var username=e.parameter.data1;
  var password=e.parameter.data2;

  var data = sheet.getDataRange().getValues();
  for (var i = 1; i < data.length; i++) {
    // var sheet_username=sheet.getRange('A'+i+1).getValue();
    // var sheet_password=sheet.getRange('B').getValue();

    if((username==data[i][0]) && (password==data[i][1])){
      return ContentService.createTextOutput(data[i][2]);
    }
  }
  return ContentService.createTextOutput(””);
}
```

Statement Script code for Verification to get the statement id for the provided ledger id:

```
var sheet_id = ”1VUNxakEKP9uYDrmlR4rtPA4vf3REQv8Q2jGQlL8bHLk”;
var main__Spreadsheet = Spreadsheet App.openBy Id(sheet_id);
var sheet = main__Spreadsheet .getSheetBy Name('username_password');

function doGet(e){
  var user_script=e.parameter.data1;

  var data = sheet.getDataRange().getValues();
  for (var i = 1; i < data.length; i++) {
    if(data[i][2]==user_script){
      return ContentService.createTextOutput(data[i][3]);
    }
  }
  return ContentService.createTextOutput(””);
}
```

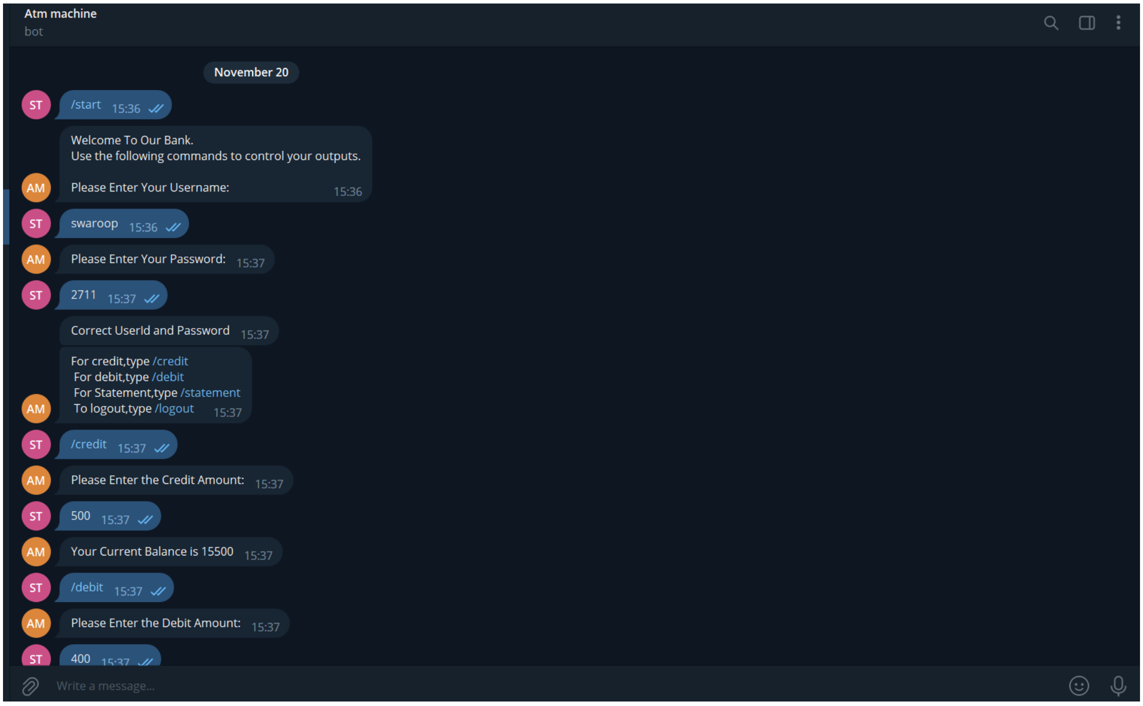
Handling the Debit and Credit for example user 1

```
var sheet_id = ”1NwkiDU6YFpiIBEYgZ7xYB3YyuA85SIcQjI1BEJh5–Tg”;
var LEDGER__SHEET = Spreadsheet App.openBy Id(sheet_id);
var sheet = LEDGER___SHEET.getSheetBy Name('ledger');

function doGet(e){
  var credit=parseInt(e.parameter.data1,10);
  var debit=parseInt(e.parameter.data2,10);
  // var balance=parseInt(e.parameter.data3,10);

  var Direction=SpreadsheetApp.Direction;
  var aLast ...
     =ss.getRange(”A”+(ss.getLastRow()+1)).getNextDataCell(Direction.UP).getRow();
  var previous_balance=sheet.getRange('C'+aLast).getValue();

  var new_balance=previous_balance+credit-debit;
  sheet.appendRow([credit,debit,new_balance]);
  return ContentService.createTextOutput(new_balance);
}
```

Handling the statement request for example user 1

```
var sheet_id = ”1NwkiDU6YFpiIBEYgZ7xYB3YyuA85SIcQjI1BEJh5–Tg”;
var LEDGER__SHEET = Spreadsheet App.openBy Id(sheet_id);
var sheet = LEDGER___SHEET.getSheet ByName('ledger');

function doGet(e){
  var data = sheet.getDataRange().getValues();
  var str=””;
  for (var i = 1; i < data.length; i++) {
    str+=data[i][0]+” ”+data[i][1]+” ”+data[i][2]+”;”;
    // Logger.log('Product number: ' + data[i][1]);
    // Logger.log('Product number: ' + data[i][2]);
  }
  return ContentService.createTextOutput(str);
}
```

Output Photos: The following is an example transaction by a user.



Atm machine
bot

November 20

ST /start 15:36

AM Welcome To Our Bank.
Use the following commands to control your outputs.

Please Enter Your Username: 15:36

ST swaroop 15:36

AM Please Enter Your Password: 15:37

ST 2711 15:37

Correct UserId and Password 15:37

AM For credit,type /credit
For debit,type /debit
For Statement,type /statement
To logout,type /logout 15:37

ST /credit 15:37

AM Please Enter the Credit Amount: 15:37

ST 500 15:37

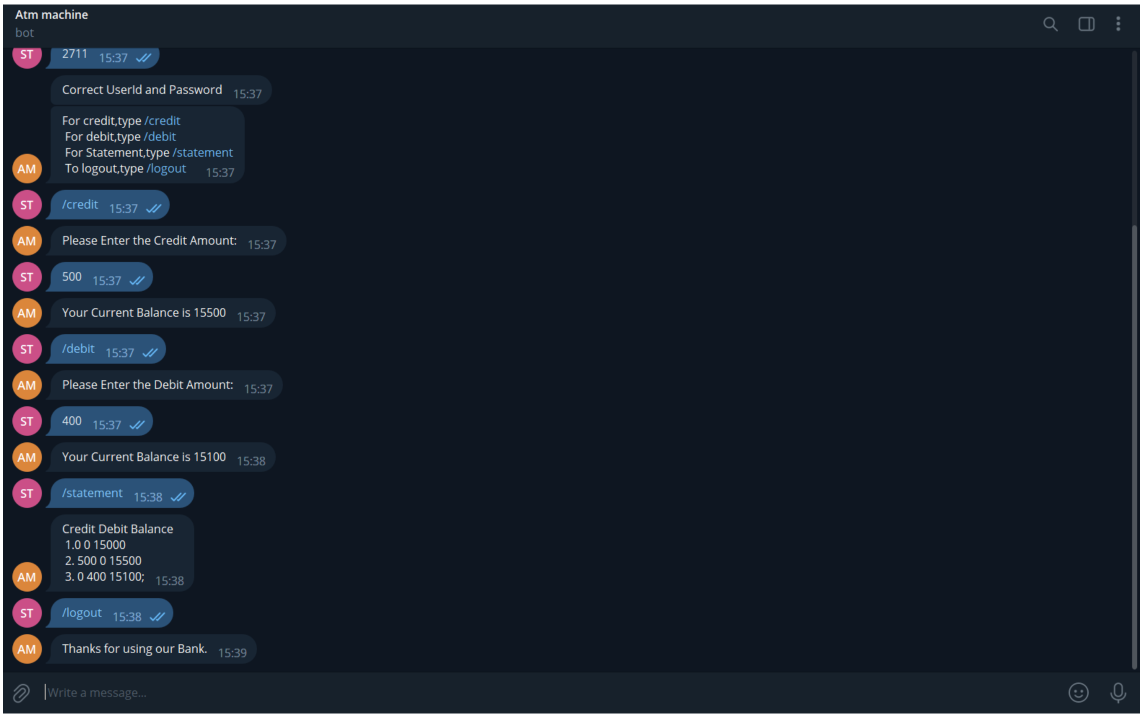
AM Your Current Balance is 15500 15:37

ST /debit 15:37

AM Please Enter the Debit Amount: 15:37

ST 400 15:37

Telegram Bot Output Pic 1



Atm machine
bot

ST 2711 15:37

Correct UserId and Password 15:37

AM For credit,type /credit
For debit,type /debit
For Statement,type /statement
To logout,type /logout 15:37

ST /credit 15:37

AM Please Enter the Credit Amount: 15:37

ST 500 15:37

AM Your Current Balance is 15500 15:37

ST /debit 15:37

AM Please Enter the Debit Amount: 15:37

ST 400 15:37

AM Your Current Balance is 15100 15:38

ST /statement 15:38

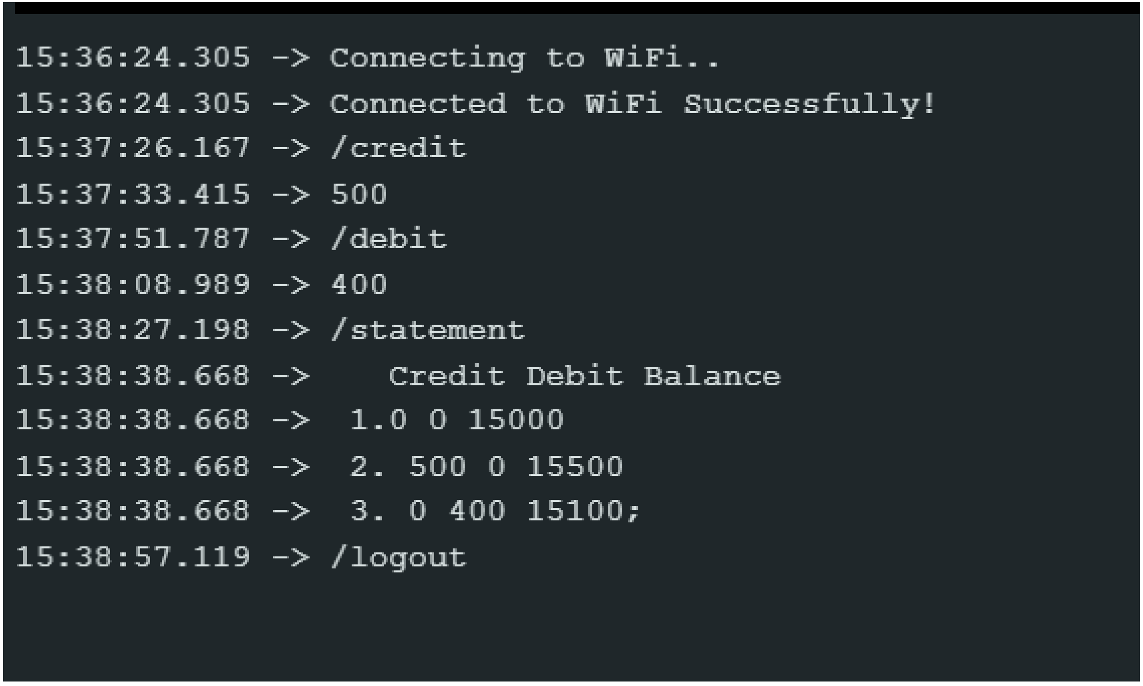
AM Credit Debit Balance
1.0 0 15000
2. 500 0 15500
3. 0 400 15100; 15:38

ST /logout 15:38

AM Thanks for using our Bank. 15:39

Telegram Bot Output Pic 2

The following is the output displayed on the Serial Monitor



```
15:36:24.305 -> Connecting to WiFi..
15:36:24.305 -> Connected to WiFi Successfully!
15:37:26.167 -> /credit
15:37:33.415 -> 500
15:37:51.787 -> /debit
15:38:08.989 -> 400
15:38:27.198 -> /statement
15:38:38.668 ->    Credit Debit Balance
15:38:38.668 ->  1.0 0 15000
15:38:38.668 ->  2. 500 0 15500
15:38:38.668 ->  3. 0 400 15100;
15:38:57.119 -> /logout
```

Serial Monitor Output

Value Addition:

- We can also use Touch pins or keypads for taking input.
- An HTML web page with forms can be used in place of the Telegram bot to enter the password and the amount to be withdrawn/credited.
- We can add an internal fund transfer mechanism between the users present in the bank.
- We can send the statement or send an email or message to the user's mobile number or email address if the amount to be debited/credited is of large value.
- We can use an LCD display to display balance.

Conclusion: The Telegram bot was used to imitate the ATM system on the ESP32 platform. It was used to collect inputs, display results, and combine Apps Script with Google Sheets to carry out basic operations including user authentication, money crediting or debiting, and statement generation.