Visvesvaraya National Institute of Technology (VNIT), Nagpur

Embedded Systems
(ECL 403)
Ultimate task (End Semester) Report

Submitted by :
Rohitha Somuri BT20ECE097
5th Semester

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

Ultimate Task : Complete ATM Machine

ATM machine system using ESP32 development board

Aim :

Creating a functional ATM machine type atmosphere with ESP32 development board using google sheets and a web server.

Apparatus :

ESP32 development board, USB cable, functional Personal Computer.

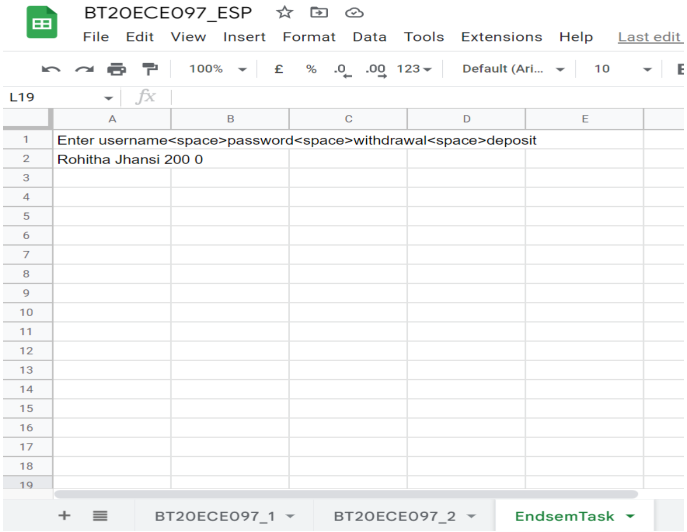
Theory :

This task mainly is based on ATM machine functionalities like deposit and withdrawal of money after successful login. We can use any mode to display the transaction statement like a telegram bot, web server or any other innovative idea. Here, Web server is used as it can be used on any device which have an internet facility, which is quite obvious now a days.

Logic :

- In this task the approach that is taken is that the username and password are already saved in the main code written in arduino ide, So here the user enters his details in google sheet block which is read by writing approriate google script with respect to that block.
- Here we have taken input as;
  username (space) password (space) withdrawal amount (space) deposit amount .
- Now, if the username and password are correct the initial balance i.e; Rs.15000 is displayed on serial monitor and entered deposit or withdrawal amount is displayed.
- Next, the calculated balance which is present now is displayed on serial monitor and on the web page we can see our present balance and the transaction amount.
- Here, we can see the account statement on serial monitor clearly, the web page shows the updated content after every refresh.

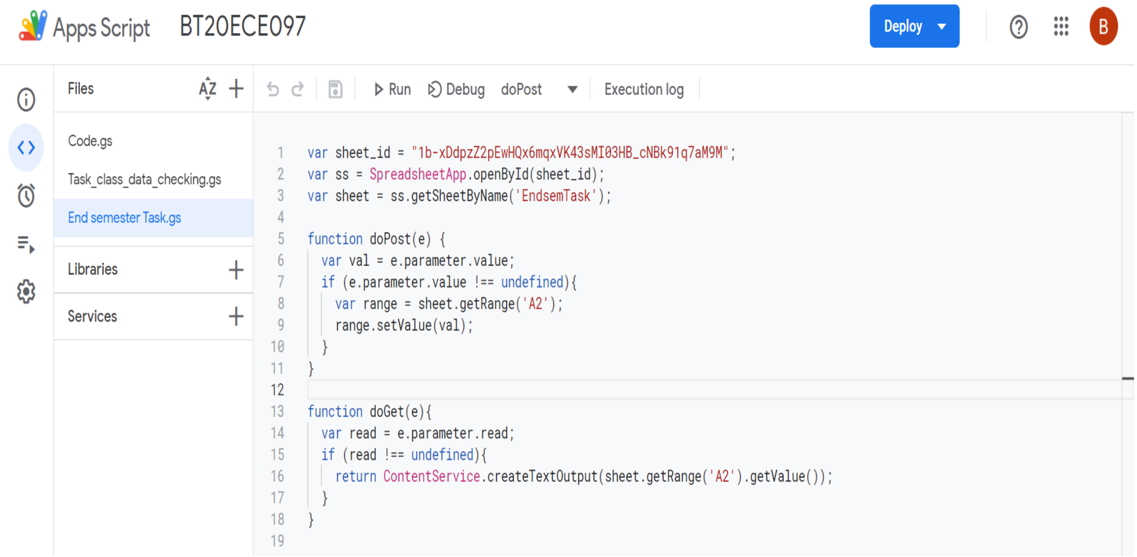
- Input is given from google sheets as shown;


BT20ECE097_ESP
File Edit View Insert Format Data Tools Extensions Help Last edit
100% £ % .0 .00 123 Default (Ari... 10
L19
BT20ECE097_1
BT20ECE097_2
EndsemTask

	A
1	Enter username<space>password<space>withdrawal<space>deposit
2	Rohitha Jhansi 200 0

Figure 1: Google Sheets

- The google script written by method google sheet-extensions-apps script-new .gs file. They the file is deployed by the deploy option as shown in figure.



```
var sheet_id = "1b-xDdpzZ2pEwHQx6mqxVK43sMI03HB_cNBk91q7aM9M";
var ss = SpreadsheetApp.openById(sheet_id);
var sheet = ss.getSheetByName('EndsemTask');

function doPost(e) {
  var val = e.parameter.value;
  if (e.parameter.value !== undefined){
    var range = sheet.getRange('A2');
    range.setValue(val);
  }
}

function doGet(e){
  var read = e.parameter.read;
  if (read !== undefined){
    return ContentService.createTextOutput(sheet.getRange('A2').getValue());
  }
}
```

Figure 2: Google Script

- The code is pasted in code section.

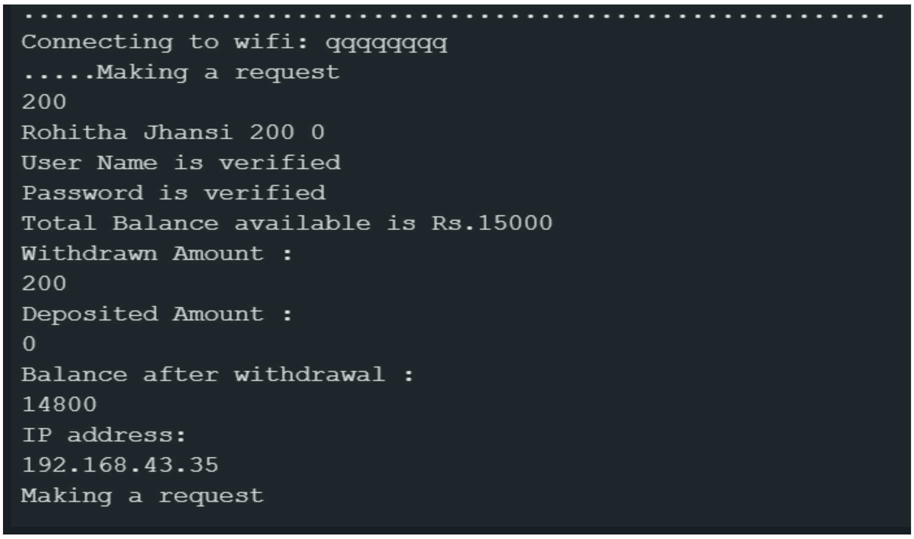
Code :

Below is the arduino ide code written for the operation

```
#include "WiFi.h" //including required libraries.
#include <HTTPClient.h>
#include <stdio.h>

// WiFi credentials
const char* ssid  = "qqqqqqqq"; // My Mobile SSID.
const char* password = "qqqqqqqq"; // Hotspot password.

// declaring a server
WiFiServer server(80);

// Google script ID and required credentials.
String GOOGLE_SCRIPT_ID =
    "AKfycbw6W8C6JUymyrlu0thOSpqb5lbadJzGuQhCxDaZaFozmW28Ldk1ctp-mqoFB3-Zhus9";

//declring some variables required globally.
int var;
int var1;
int with_amt;
int dep_amt;
int tag = 0; //  initializing  a tag variable  usefulin  upcoming code for updating balance into an
    array.
int total[100]  = {15000}; // array where balance is stored  after  ecah transaction.

// Function for withdrawal calculations.
int withdrawal(int w,int tag)
{
    int bal;
    int a = sizeof(total)/sizeof(total[0]);
    bal = total[tag]-w;
    total[tag+1]=bal; // Appending present balance to the total array.
    return bal;
}

//Function for deposit calculations.
int deposit(int d,int tag)
{
  int bal;
    int a = sizeof(total)/sizeof(total[0]);
    bal = total[tag]+d;
    total[tag+1]=bal; // Appending present balance to the total array.
    return bal;
}

//Setup function (executed once).
void setup () {
  delay (1000);
  Serial.begin(115200);
  delay(1000);
  // connect to WiFi
  Serial.println ();
  Serial.print("Connecting to wifi: ");
  Serial.println(ssid);
  Serial.flush() ;
  WiFi.begin (ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay (500);
    Serial.print(”.”); // if not connected
  }
  pinMode(2,OUTPUT);
}

// Loop function (executes repeatedly).
void loop () {
  if (WiFi.status () == WL_CONNECTED) {
    char a = 0;
    int b = 0;
    String list [100]; // this array is to tore the nput given from google sheets.
    int count = 0;
    String variable ;
    HTTPClient http;
    a = Serial.read();
    String url = ”https://script.google.com/macros/s/” + GOOGLE_SCRIPT_ID + ”/exec?read”;
    Serial.println(”Making a request”); // this message on serial monitor shows that process of
         fetching data is started.
    http.begin(url.c_str());
    http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
    int httpCode = http.GET();
    String dataFromSheet;
    if (httpCode > 0) {
      dataFromSheet = http.getString(); // string fetched from google sheet is stored in a variable.
      Serial.println(httpCode);
      Serial.println(dataFromSheet);

      // This while loop is for splitting the string to get separate data.
      while(dataFromSheet.length() > 0)
      {
        b = dataFromSheet.indexOf(' ');
        if(b == −1)
        {
          list[count++]=dataFromSheet;
          break;
        }
        else
        {
          list[count++]=dataFromSheet.substring(0,b);
          dataFromSheet = dataFromSheet.substring(b+1);
        }
      }
      var = list[2].toInt(); // string to integer convertion for withdrawal data.
      var1 = list[3].toInt(); // string to integer convertion for deposit data.
      if(list[0]==”Rohitha”)
      {
        Serial.println(”User Name is verified”); // verification of username.
        if(list[1]==”Jhansi”)
        {
          Serial.println(”Password is verified”); // verification of password.
          Serial.println(”Total Balance available is Rs.15000”); // Initial balance printing.
          with_amt = withdrawal(var,tag);
          Serial.println("Withdrawn Amount :"); // Withdrawn amount in this transaction.
          Serial.println(var);
          dep_amt = deposit(var1,tag);
          Serial.println("Deposited Amount :"); // Deposited amount in this transaction.
          Serial.println(var1);
          if(var != 0)
          {
            Serial.println("Balance after withdrawal :"); // Balance after withdrawal.
            Serial.println(with_amt);
          }
          if(var1 != 0)
          {
            Serial.println("Balance after deposit :"); // Balance after deposit.
            Serial.println(dep_amt);
          }
          Serial.println("IP address: ");
          Serial.println(WiFi.localIP()); // IP address of the webpage created is displayed.
          server.begin();
        }
        else
        {
          Serial.println("Password is incorrect");
        }
      }
      else
      {
        Serial.println("Username is incorrect");
      }
    }
    else {
      Serial.println("Error on HTTP request"); // for error in request.
    }
    // http.end();
  }

  // webpage code.
  WiFiClient client = server.available();
  if(client)
  {
    String request = client.readStringUntil('\r');
    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println("Connection:close");
    client.println();
    // start of html code.
    client.println("<!DOCTYPE html><html>");
    // setting page size to fit any device's screen.
    client.println("<head><meta name=\"viewport\" content=\"width=device-width,
        initial-scale=1\">");
    client.println("<link rel\"icon\" href=\"data:,\">");

    // writing css for styling the web page created.
    client.println("<style>body {text-align: center; font-family: Garamond, serif;font-size:
        x-large;}");
    // creating styling for the table.
    // color of table is blue and background is white.
    client.println("table { border-collapse: collapse; width:40%; margin-left:auto;
        margin-right:auto;border-spacing: 2px;background-color: white; border: 4px solid blue; }");
    //padding is also same color as table color and the matter in it  is  of white color.
    client.println("th { padding: 20px; background-color: #0000FF; color: white}");
    client.println("tr { border: 5px solid blue; padding: 2px; }");
    client.println("tr: hover {background-color:yellow; }");
    client.println("td { border: 4px; padding: 12px; }");
    // amount class specifications.
    client.println(".amount {color:black; font-weight: bold; background-color: white; padding: 1px;
        }");
    // writing body of html.
    client.println("</style></head></body><h1>ESP32 Web Server</h1>");
    client.println("<h2>ATM Machine</h2>");
    client.println("<table><tr><th>CATEGORY</th><th>AMOUNT</th></tr>");
    // entering details in table.
    client.println("<tr><td><b>Balance Now</b></td><td><span class=\"amount\">");
    // writing in table.
    client.println(with_amt);
    client.println("<tr><td><b>Amount withdrawn or deposited</b></td><td><span
        class=\"amount\">");
    client.println(var);
    client.println("<tr><td><b>Amount deposited</b></td><td><span
        class=\"amount\">");
    client.println(var1);
    client.println(" </span></td></tr>");
    client.println("</body></html>"); // end of the html page.
   Serial.println("Client disconnected.");
  }
  tag++; // incrementing the tag.
  delay (10000); // sufficient delay.
}
```

Observations :

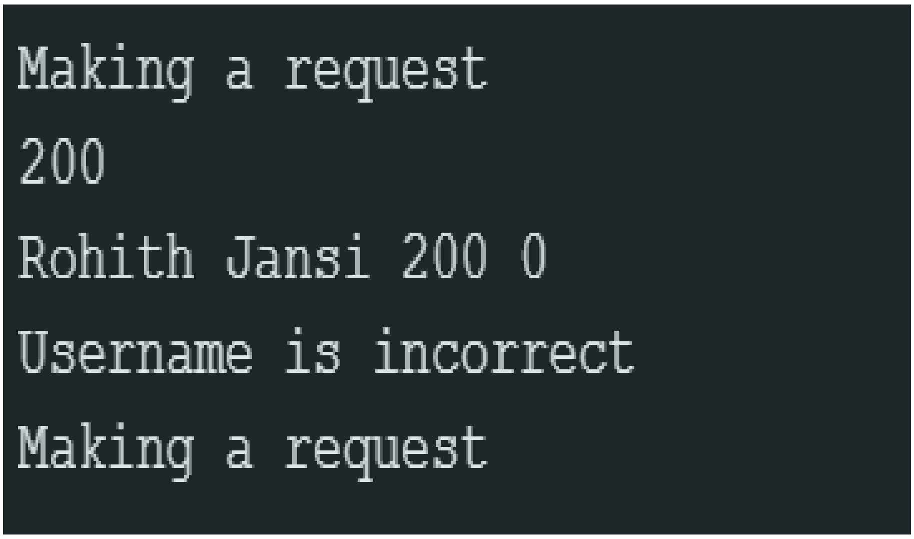
Serial monitor output for first transaction is shown below;



```
..........................................................
Connecting to wifi: qqqqqqqq
.....Making a request
200
Rohitha Jhansi 200 0
User Name is verified
Password is verified
Total Balance available is Rs.15000
Withdrawn Amount :
200
Deposited Amount :
0
Balance after withdrawal :
14800
IP address:
192.168.43.35
Making a request
```

Figure 3: serial monitor

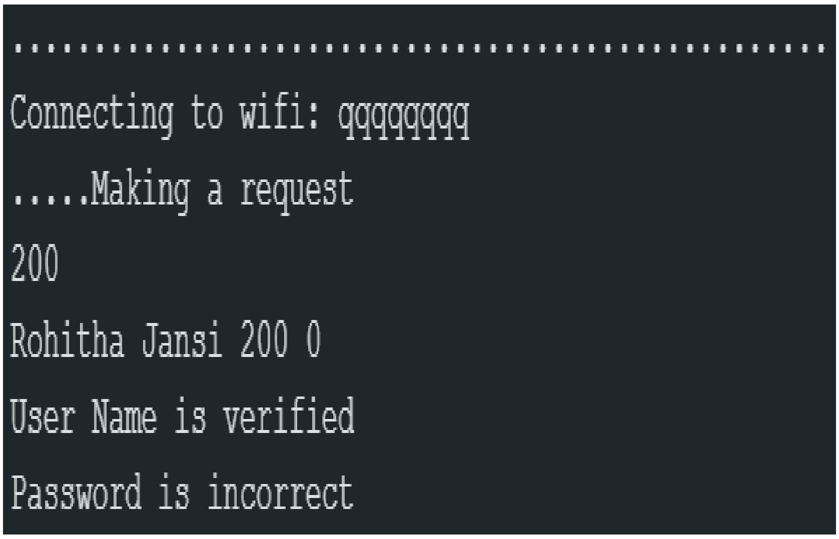
- Serial monitor output for incorrect username entered;



```
Making a request
200
Rohith Jansi 200 0
Username is incorrect
Making a request
```

Figure 4: Incorrect username

- Serial monitor output for incorrect password entered;

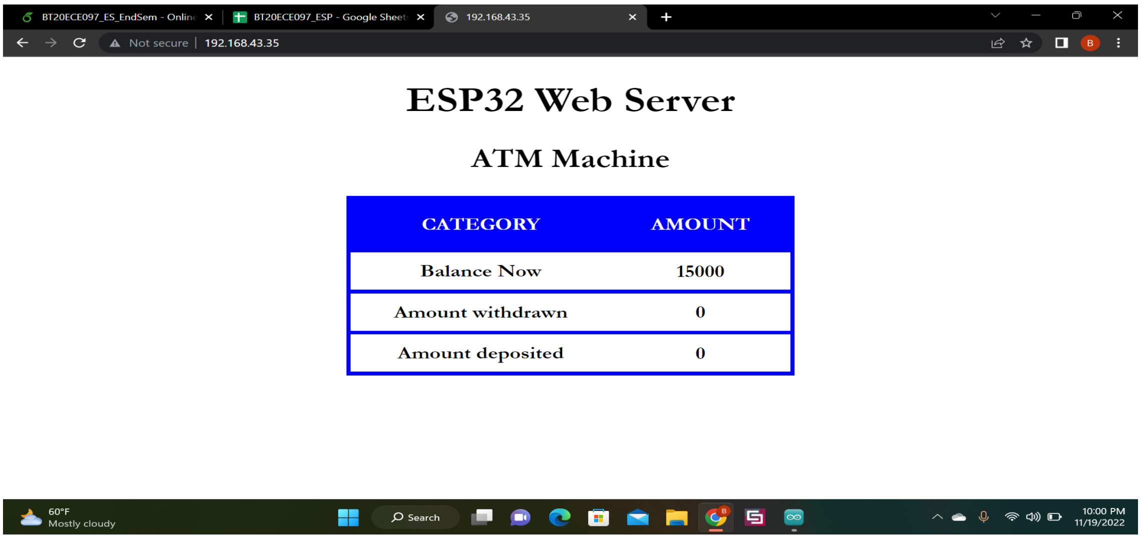


```
..........................................
Connecting to wifi: qqqqqqq
.....Making a request
200
Rohitha Jansi 200 0
User Name is verified
Password is incorrect
```

Figure 5: Incorrect password

- So, from these diagrams it can be said that if incorrect password or username is entered it does not display the amount in the account. When the correct credentials are entered it starts calculating the balance.

- Web page content before any transaction.


BT20ECE097_ES_EndSem - Onlin
BT20ECE097_ESP - Google Sheet
192.168.43.35
Not secure | 192.168.43.35
60°F
Mostly cloudy
Search
10:00 PM
11/19/2022

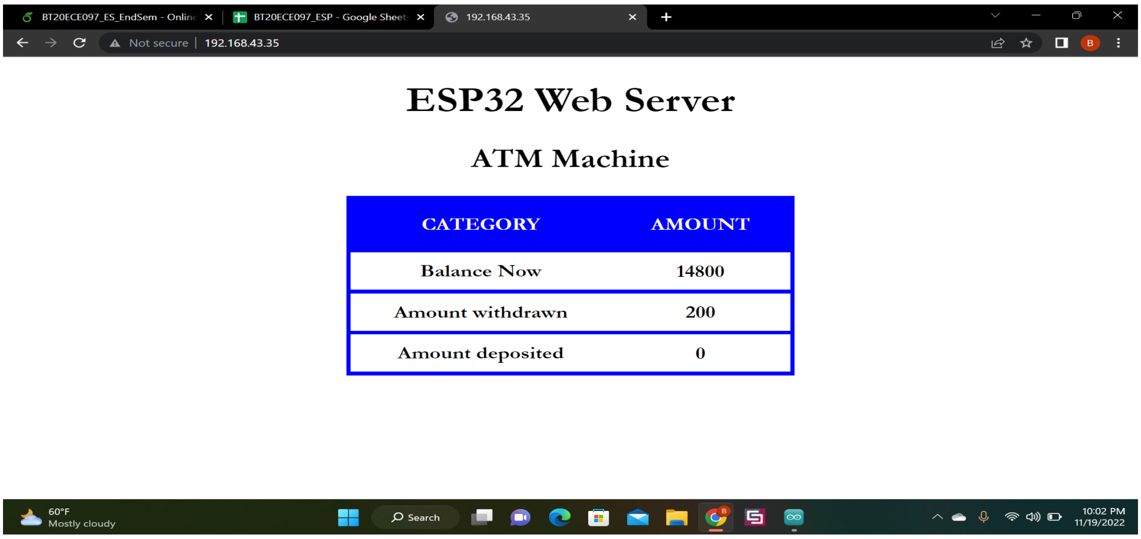
ESP32 Web Server

ATM Machine

CATEGORY	AMOUNT
Balance Now	15000
Amount withdrawn	0
Amount deposited	0

Figure 6: Web page results before transaction.

- Web page results for first transaction of Rs.200 withdrawal from Rs.15000. Now balance is Rs.14800.


BT20ECE097_ES_EndSem - Onlin
BT20ECE097_ESP - Google Sheet
192.168.43.35
Not secure | 192.168.43.35
60°F
Mostly cloudy
Search
10:02 PM
11/19/2022

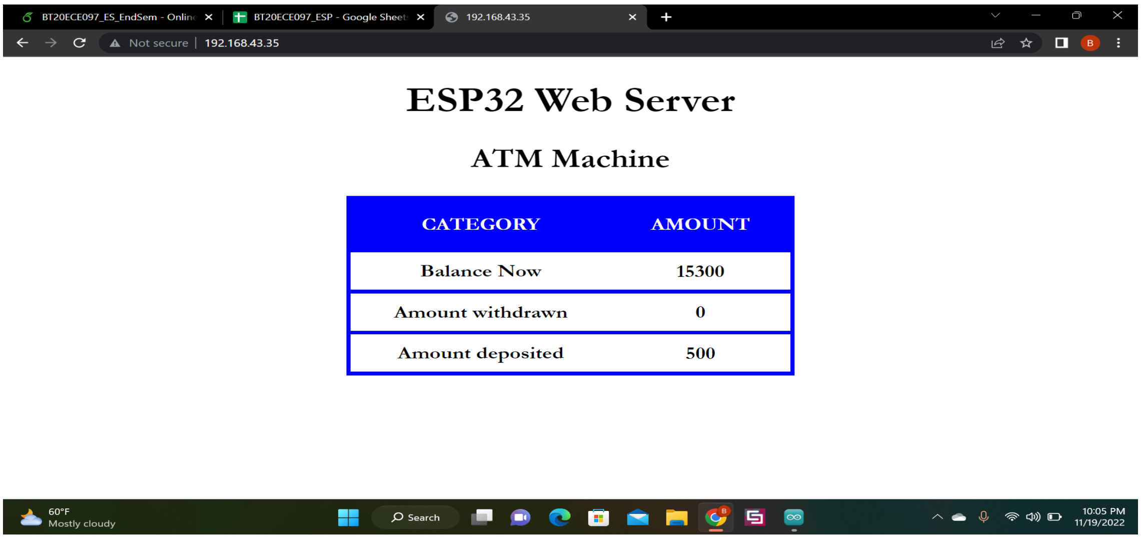
ESP32 Web Server

ATM Machine

CATEGORY	AMOUNT
Balance Now	14800
Amount withdrawn	200
Amount deposited	0

Figure 7: Web page results for transaction 1

- Web page results for second transaction of Rs.500 deposit from Rs.14800. Now balance is Rs.15300.


BT20ECE097_ES_EndSem - Onlin
BT20ECE097_ESP - Google Sheet
192.168.43.35
Not secure | 192.168.43.35
60°F
Mostly cloudy
Search
10:05 PM
11/19/2022

ESP32 Web Server

ATM Machine

CATEGORY	AMOUNT
Balance Now	15300
Amount withdrawn	0
Amount deposited	500

Figure 8: Web page results for transaction 2

Other ideas :

Other ideas can be taking the password and username from google sheets and amount from the touch pins of ESP32 and web server can be replaced by other platform like telegram bot or a GUI.

Conclusion :

Therefore, it can be concluded that the username and password getting verified properly, the total balance is getting updated correctly and the web page is working smoothly.

YouTube link of explanation :

https://youtu.be/eaRQGenkoSo