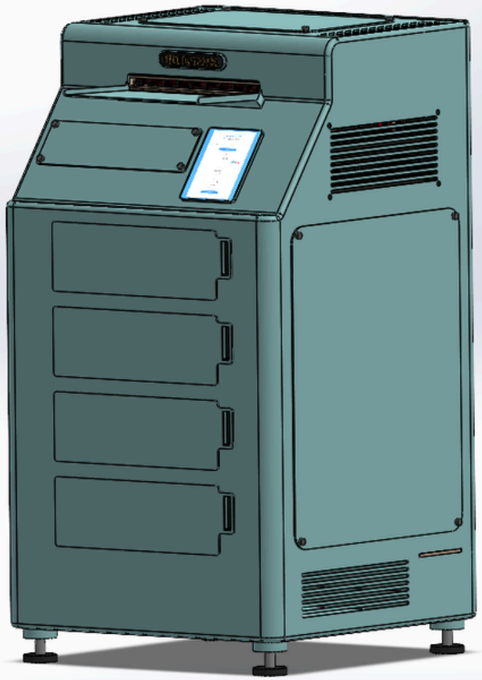
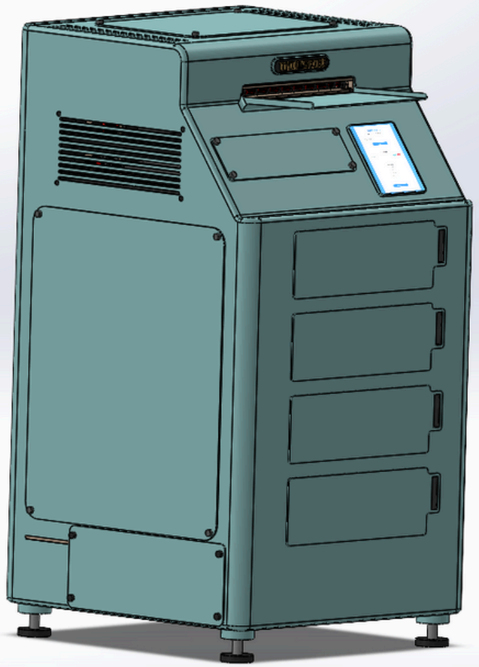
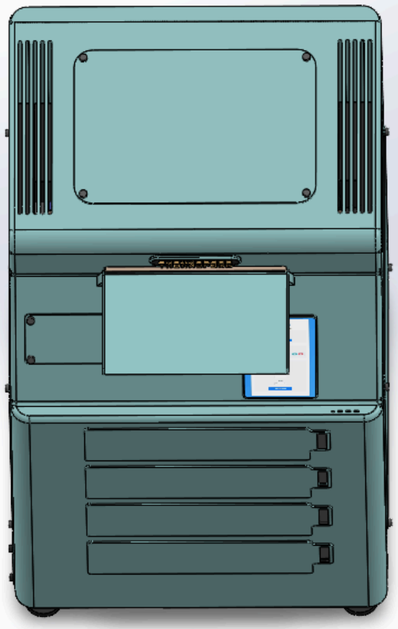
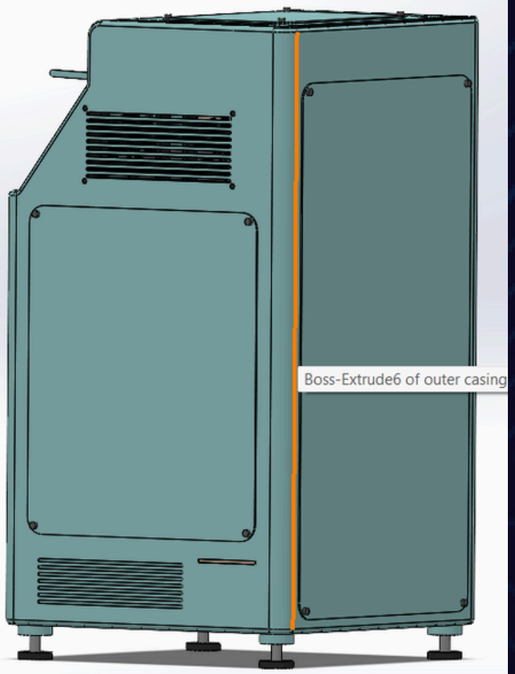
Team Thinkspire

AUTOMATIC PAPER PRINTING KIOSK







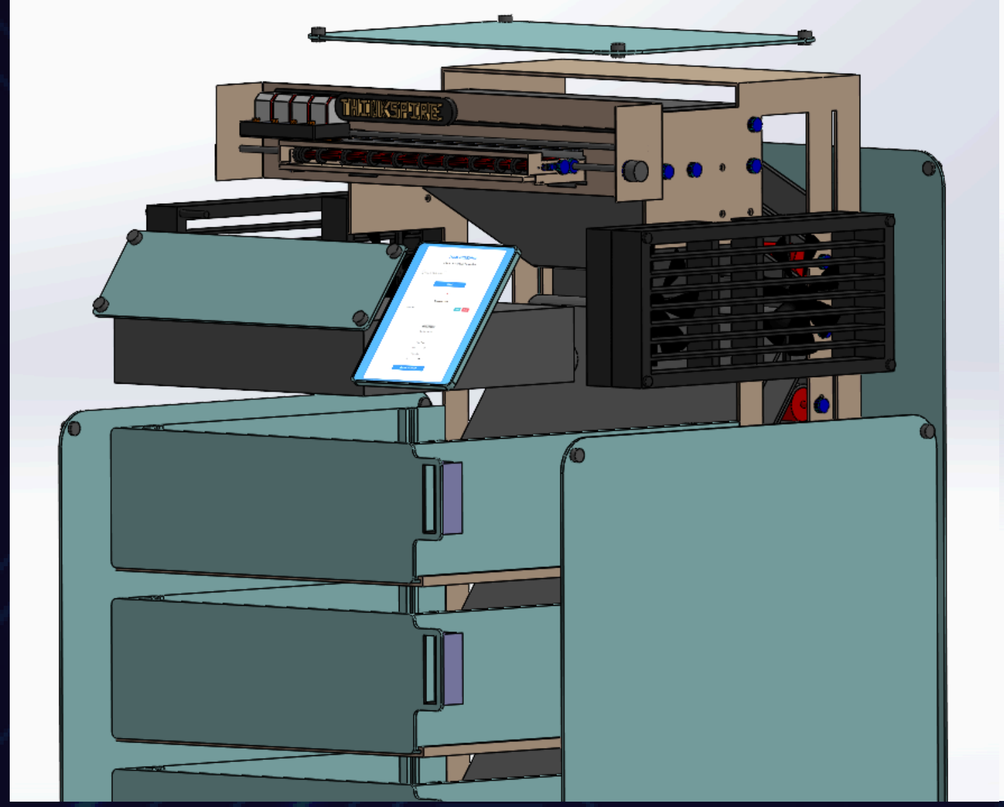


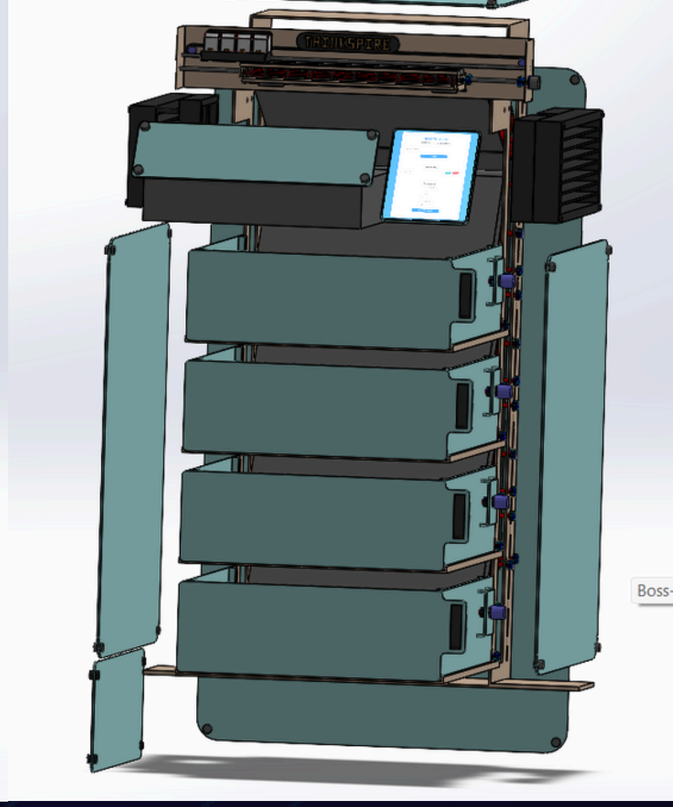
CAD MODEL

The pics of CAD model outer casing can be seen, the main features are,

1. Sturdy Body
2. Ventilation Vents
3. Large display with touchresponse
4. Storage Space for battery in case of power failure
5. 4 Paper trays with sensors and ajustable paper sizes A3, A4, etc
6. Easy access to ink tank for refilling purposes
7. Top , side and back maintainance pannels for easy access to internal components.

INTERNAL COMPONENTS





01

Large Inktank for uninturrepted prints

02

Large header for fast and colourful prints

03

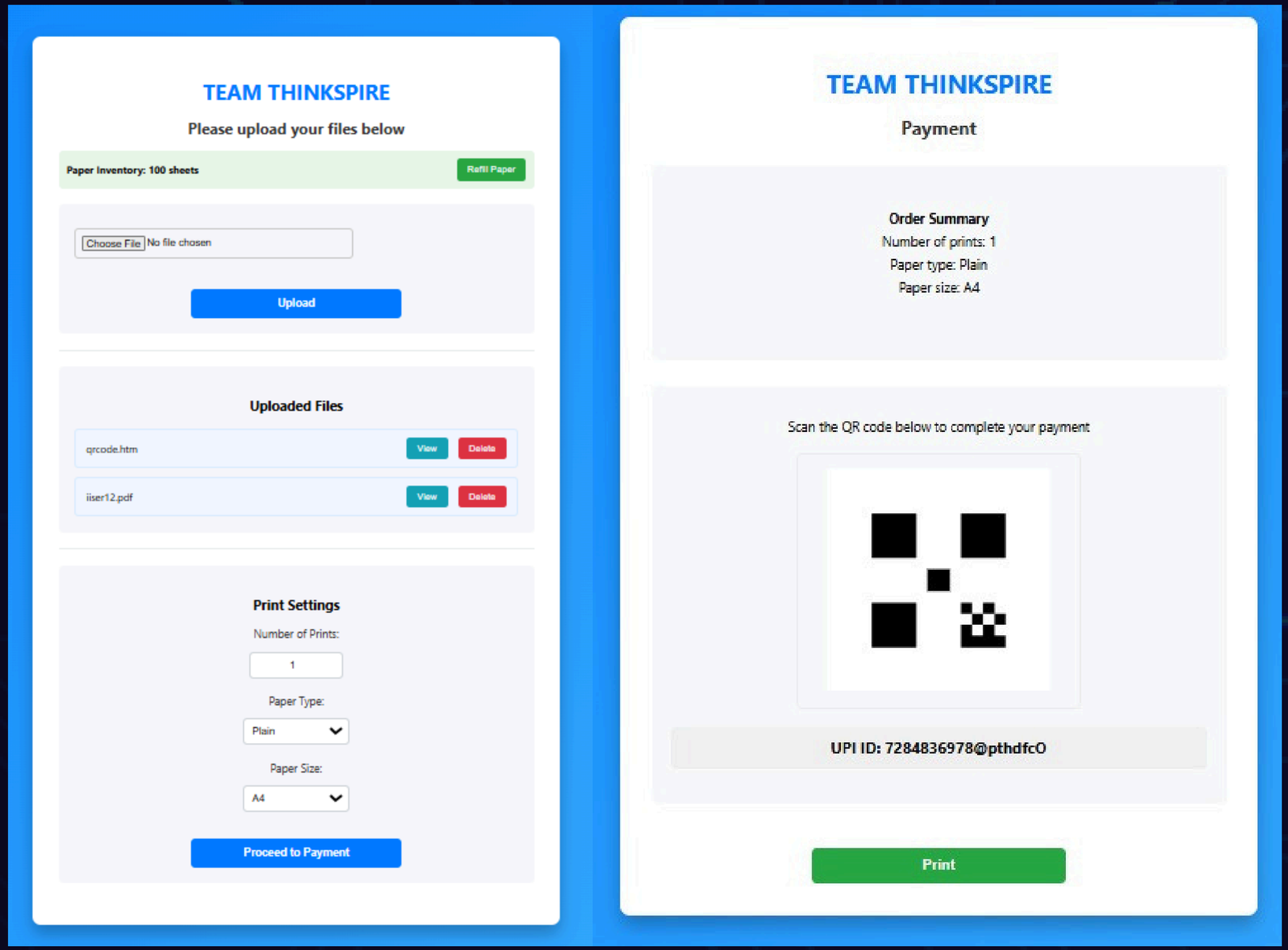
Roller and conveyor belt mechanism for precisely feeding papers form tray to printing head

04

Radiators and vents for easy cooling and heat dissipation.

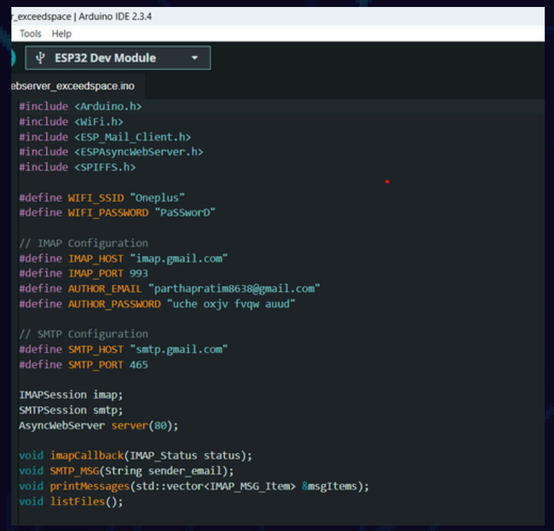
05

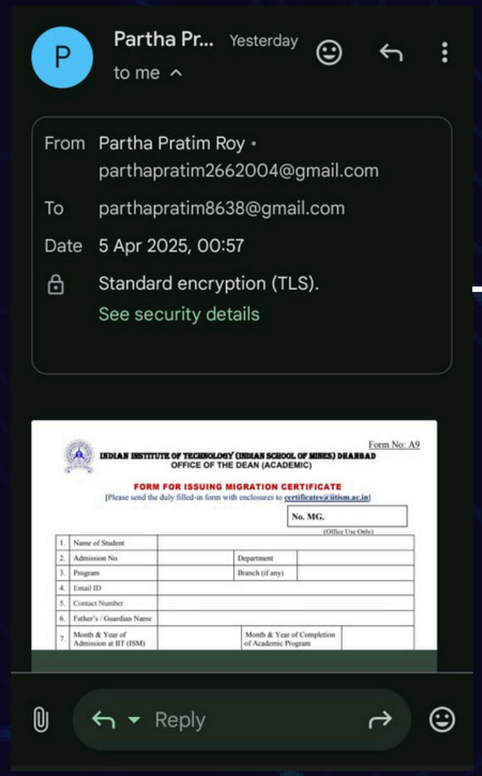
Battery compart-ment for emergency use and to ensure 24x7 prints

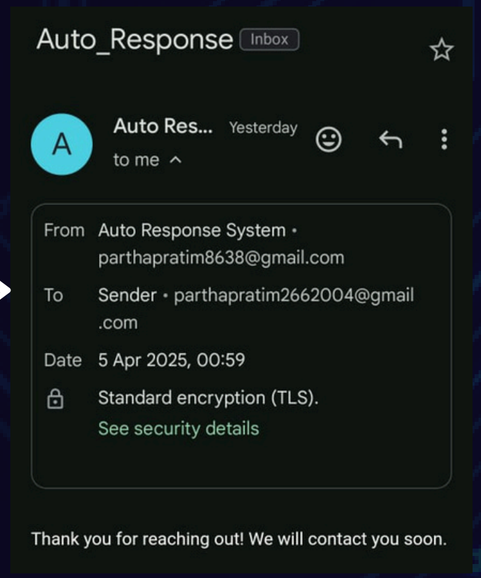


USER INTERFACE

- Users can upload files via the web server.
- Supported file formats include PDF, JPG, PNG, and others.
- A preview of the uploaded file is displayed.
- Users can manually delete uploaded files.
- Options are provided to select the number of copies, paper type, and paper size.
- A QR code scanner is integrated.
- The system automatically counts and displays the number of pages remaining in the paper container.







FILE UPLOAD VIA E-MAIL

01 Uses IMAP to parse the Email fetches the attachment to be printed.

02 Uses STMP to send Automated Email to the customer for reference.

03 IMAP works on online as well as offline mode .

ent_int_paypal.ino

```
#include <WiFi.h>
#define PubNub_BASE_CLIENT WiFiClient
#include <PubNub.h>

static char ssid[] = "Oneplus";
static char pass[] = "PaSSworD";
const static char pubkey[] = "pub-c-08a8d6f1-e17a-460b-9e4f-e50177468bff";
const static char subkey[] = "sub-c-238ca340-9be4-46b4-bfa4-5c3a7907f1cb";
const static char channel[] = "paymentTrigger";
String message;

void setup() {
    Serial.begin(115200);
    Serial.println("Attempting to connect...");
    WiFi.begin(ssid, pass);

    // Wait for connection with timeout
    int timeout = 0;
    while (WiFi.status() != WL_CONNECTED && timeout < 20) {
        delay(500);
        Serial.print(".");
        timeout++;
    }

    if (WiFi.status() != WL_CONNECTED) {
        Serial.println("Couldn't connect to WiFi.");
        while(1) delay(100);
    } else {
        Serial.println("");
        Serial.print("Connected to SSID: ");
        Serial.println(ssid);
        Serial.print("IP address: ");
        Serial.println(WiFi.localIP());
```

```
-> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
-> configsip: 0, SPIWP:0xee
-> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
-> mode:DIO, clock div:1
-> load:0x3fff0030,len:4888
-> load:0x40078000,len:16516
-> load:0x40080400,len:4
-> load:0x40080404,len:3476
-> entry 0x400805b4
-> Attempting to connect...
-> ...
-> Connected to SSID: Oneplus
-> IP address: 192.168.181.153
-> PubNub is set up.
-> []]Payment Received
```

PAYMENT INTEGRATION

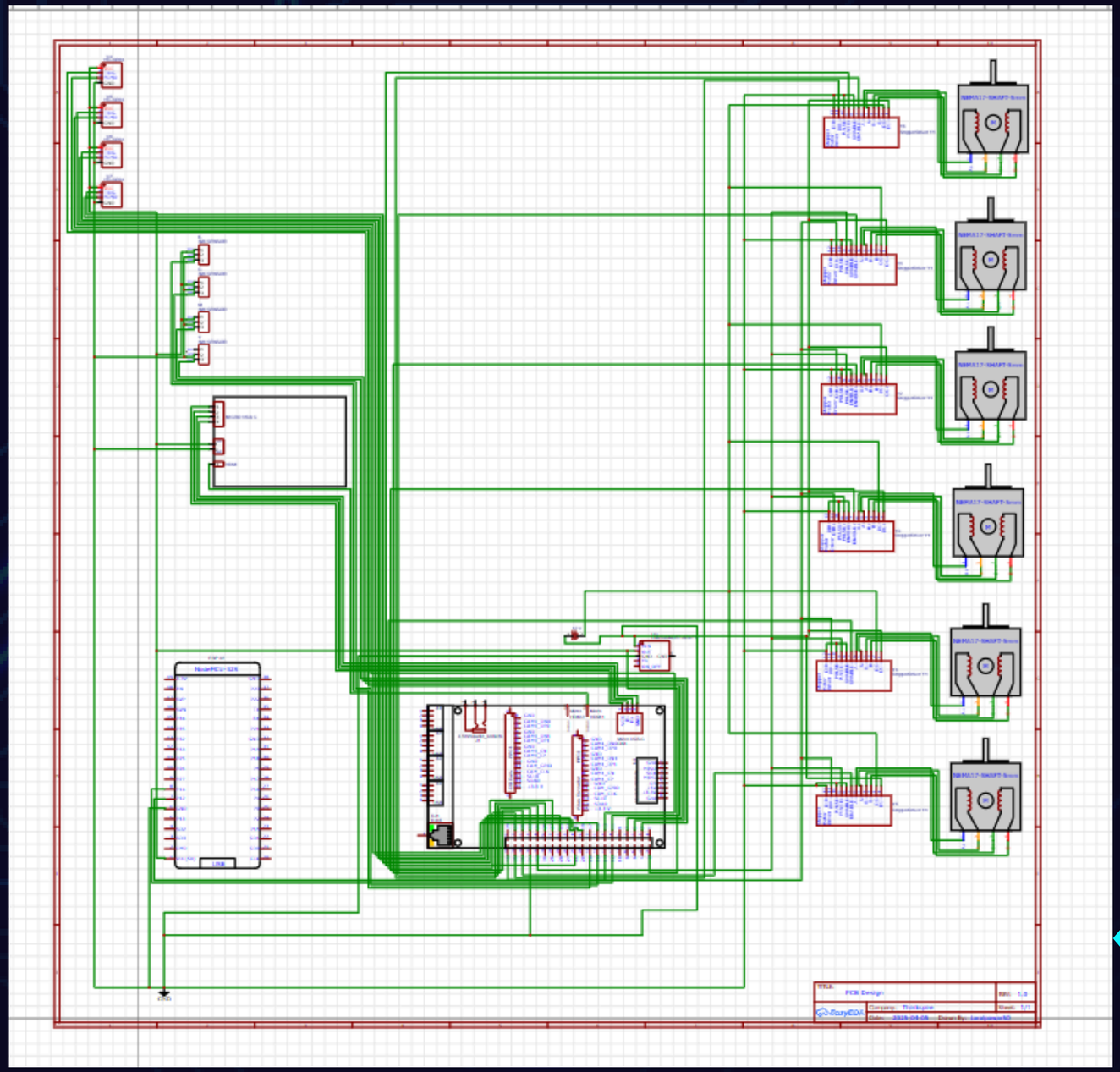
01 Uses Pubnub MQTT to demonstrate the payment process.

02 Uses the Paypal webhook simulator to show “Payment received”.

03 It works on ESP 32 using the Pubnub.h library.

PCB DESIGN

- Used ESP 32 and raspberry pi 5 boards for wireless transfer for files and for getting access to servers for uploading documents with the help of Wifi, USB or Email.
- 6 Stepper motors : 1 for head movement, 1 for conveyor belt, and 4 for each paper tray respectively
- 4 Ultrasonic sensors for each paper tray for giving alert for low paper quantity.
- 4 Ink sensors to alert user for low ink for respectively CMYK colours.
- LCD touchscreen display supporting HDMI (feedback is given through micro USB C)



MATLAB SIMULATION BOX MODEL

- This simulation represents a functional model of an inkjet printer developed in MATLAB and Simulink, simulating the process of image acquisition, preprocessing, and printhead control through an ESP32 microcontroller interface. The model captures the essential digital image handling and control logic used to drive a simulated inkjet printhead.

1. Image Acquisition:-
- RGB image is uploaded in MATLAB (via script or GUI).
- Acts as the source for the printing simulation.

2. Preprocessing in MATLAB:-
- Convert RGB to grayscale.
- Resize to match printer resolution.
- Convert to double precision (required by Simulink).

3. Time Series Conversion:
- Reshape image so each column = one time step (line-by-line simulation).

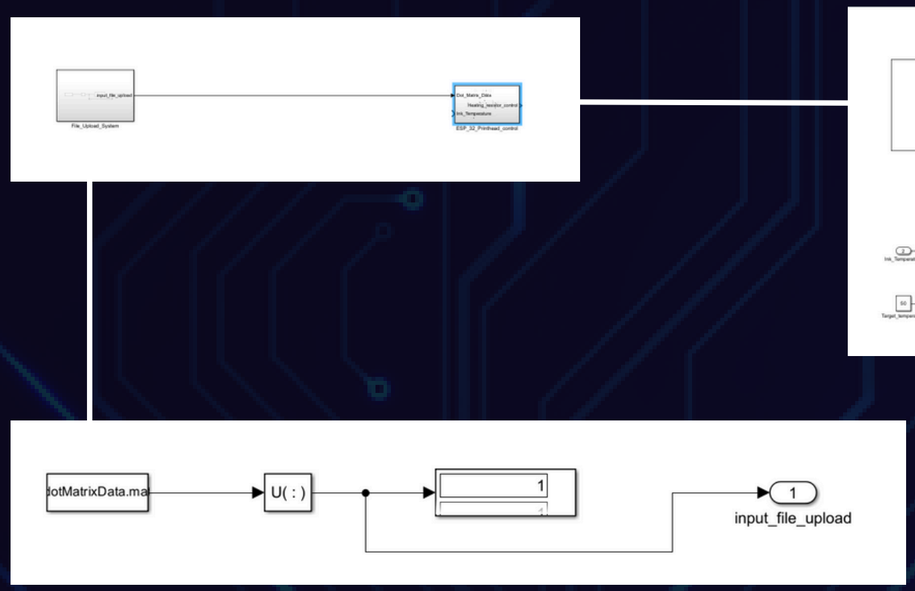
4.Data Export:
- Save as .mat file (v7.3) for Simulink compatibility.

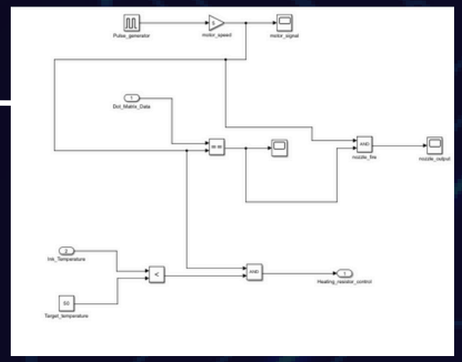
Editor - C:\Users\hp\OneDrive\Desktop\pdf_matrix.m

pdf_matrix.m

```
% Step 1: Read and preprocess the image
img = imread('12th marksheet.jpg');
grayImg = rgb2gray(img);
bw = imbinarize(grayImg);      % Binary image (logical)

% Step 2: Resize for realistic printer resolution (optional)
bw = imresize(bw, [24 NaN]);   % Resize to 24 rows (pin height)

% Step 3: Convert to double (required by Simulink)
dotMatrix = double(bw);        % Still [24 x N] matrix

% Step 4: Convert to time series (each column is a sample in time)
ts = timeseries(dotMatrix', 1:size(dotMatrix,2));  % Transpose for Simulink!

% Step 5: Save as -v7.3
save('dotMatrixData.mat', 'ts', '-v7.3');

```

MATLAB SIMULATION BOX MODEL



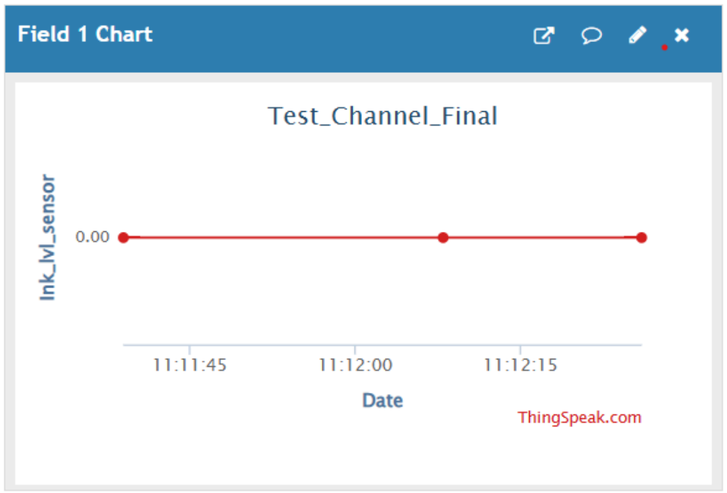


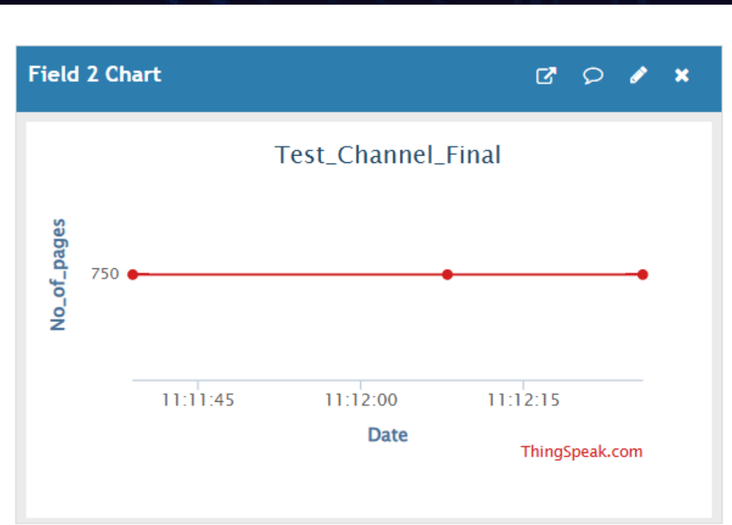
5. Simulink File Upload Block:-
   - Loads time series data into the Simulink model.
   - Feeds sequential pixel data to the controller.

6. ESP32 Printhead Control Block:-
   - Simulates ESP32 logic to control the printhead.
   - Converts pixel values into actuator (PWM/GPIO) signals for ink droplet ejection.

Model Objectives & Capabilities:

- Simulates a basic digital-to-physical print pipeline.
- Demonstrates how image data is processed and streamed to a microcontroller-based printhead driver.
- Enables testing of:
- Printhead activation logic.
- Timing and control signal accuracy.
- Data resolution and throughput handling.





SENSOR-BASED MONITORING OF INK AND PAPER LEVELS IN PRINTERS

- The ESP32 is integrated with ThingSpeak for real-time data monitoring and visualization.
- An ink level sensor is used to track the remaining ink in the printer.
- An ultrasonic sensor measures and reports the number of pages left in the container.
- The data is sent to ThingSpeak, where graphical trends can be visualized.
- Further analysis of the collected data can be performed using MATLAB.

OUR TEAM

Taral Pawar

Designing CAD model, PCB design

Patel Parth

Designing user interface, Matlab simulation box model .

Partha Pratim Roy

Matlab simulation box model, User Interface, PCB design.

THANK YOU