AI & SUSTAINABILITY IN VET EDUCATION

Όνομα συγγραφέα/ων: GR Team

Τίτλος WP: WP3 – AI-SAVES@ProjectHub

Αναφορά δραστηριότητας: 3.2 – Project template for AI-SAVES@ProjectHub

Λογότυπο:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Πίνακας Περιεχομένων

Table of Contents **2**

1. Project Title 3
2. Project Category 3
3. Project Description 3
4. Keywords 4
5. Target Audience 4
6. Project Objectives 4
7. Project Timeline 4
8. Resources Provided 5
9. Tools and Technologies Used 5
10. Step-by-Step Implementation Plan 5
11. Evaluation Criteria 9
12. Languages Available 9
13. Contributors 9
14. Feedback and Continuous Improvement 10

Submission Notes **10**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

---

1. Τίτλος Project:

Web-Controlled Camera Turret

---

2. Κατηγορία Project:

AI / Robotics / Computer Vision – STEM Maker Project

3. Περιγραφή Project:

Το project δείχνει πώς μπορεί να κατασκευαστεί ένα web-controlled camera turret με χρήση Raspberry Pi και servo motors. Το σύστημα επιτρέπει στους χρήστες να ελέγχουν απομακρυσμένα τον προσανατολισμό της κάμερας μέσω web interface, προσφέροντας real-time monitoring και remote viewing.

Το Raspberry Pi λειτουργεί ως ένας μικρός server που κάνει stream το video από την κάμερα, ενώ ταυτόχρονα επιτρέπει στους χρήστες να ελέγχουν την οριζόντια και την κατακόρυφη κίνηση της κάμερας με buttons σε μια web page. Με αυτόν τον τρόπο δημιουργείται ένα απλό pan-tilt surveillance system, προσβάσιμο από browser.

Το project εισάγει τους μαθητές στη robotics, στο computer vision, στον έλεγχο servo motors και στα web-based interfaces. Μπορεί επίσης να επεκταθεί με AI features, όπως face detection, motion tracking ή automated surveillance systems.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Λέξεις-κλειδιά:

Raspberry Pi
camera turret
pan-tilt camera
robotics
web control
computer vision
remote monitoring
IoT surveillance

---

5. Κοινό-στόχος:

Μαθητές Vocational Education and Training (VET)

Εργαστήρια Robotics και STEM

Μαθητές/σπουδαστές που ενδιαφέρονται για IoT και computer vision

---

6. Στόχοι Project:

Να μάθουν οι συμμετέχοντες πώς κατασκευάζεται ένα pan-tilt camera system

Να κατανοήσουν πώς τα servo motors ελέγχουν τη μηχανική κίνηση

Να μάθουν να δημιουργούν web interfaces για έλεγχο συσκευών

Να προγραμματίσουν τα GPIO pins του Raspberry Pi

Να διερευνήσουν εφαρμογές computer vision και surveillance technology

---

7. Χρονοδιάγραμμα Project:

Εκτιμώμενη διάρκεια: 8–10 ώρες

Φάση	**Χρόνος**
Εισαγωγή στα remote monitoring systems	1 ώρα
Συναρμολόγηση hardware	2 ώρες
Καλωδίωση electronics	2 ώρες
Ρύθμιση software	2 ώρες
Δοκιμές και βελτιώσεις	1–2 ώρες

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. Παρεχόμενοι πόροι:

Οδηγίες σχεδίασης του camera turret

Οδηγός ρύθμισης Raspberry Pi

Διάγραμμα καλωδίωσης servo motor

Κώδικας web interface

Ρύθμιση video streaming

9. Εργαλεία και τεχνολογίες που χρησιμοποιούνται:

Raspberry Pi

Raspberry Pi Camera Module

2 servo motors (pan και tilt)

Servo mounting bracket

Jumper wires

Power supply

Software

Raspberry Pi OS
Python
GPIO libraries
Web server software
Video streaming tools

Camera stand ή βάση

10. Step-by-Step Implementation Plan:

Step 1: Ρύθμιση του Raspberry Pi και ενημέρωση του Operating System

Θεωρούμε ότι γνωρίζετε πώς να κάνετε αρχική ρύθμιση του Raspberry Pi. Αν όχι, υπάρχουν πολλοί αναλυτικοί οδηγοί σε άλλα σημεία που το εξηγούν καλύτερα. Μπορείτε πιθανότατα να

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

το κάνετε και με SSH, όμως προτείνεται να χρησιμοποιήσετε οθόνη και πληκτρολόγιο συνδεδεμένα στο Pi.

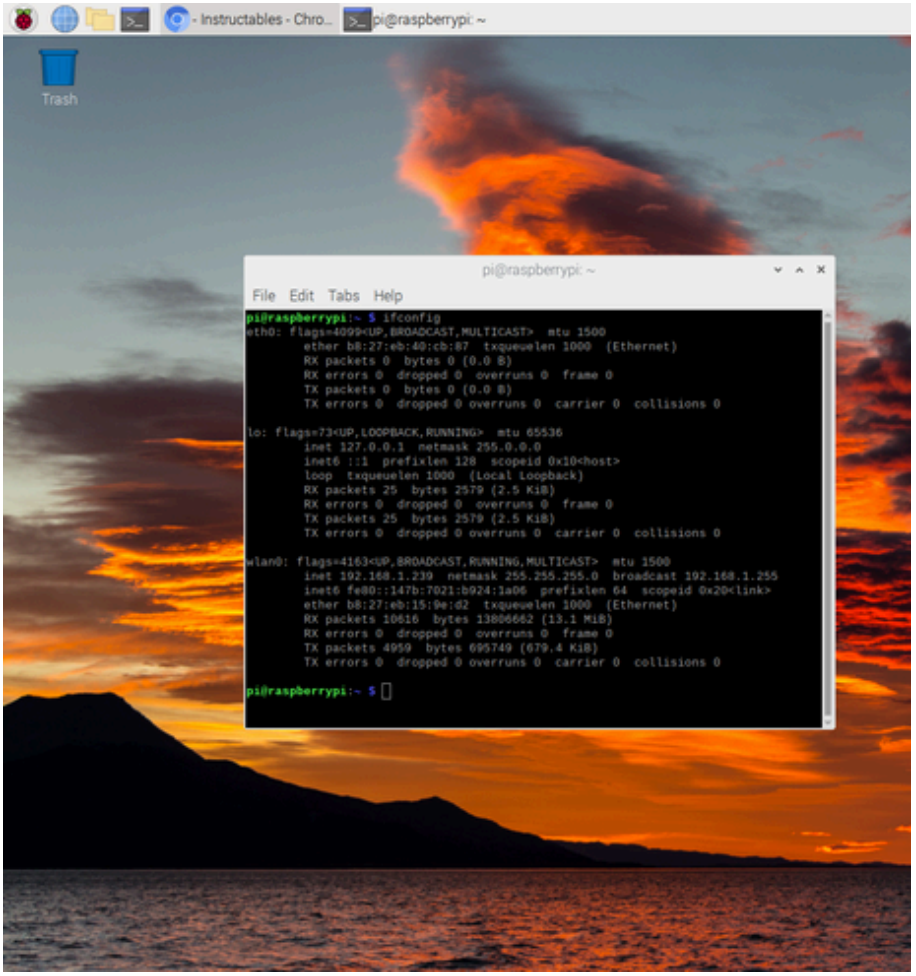
Συνδέστε το camera module στο Pi χρησιμοποιώντας το ribbon cable.

Αφού εκκινήσει το Pi, εκτελέστε τις παρακάτω εντολές σε ένα νέο terminal.

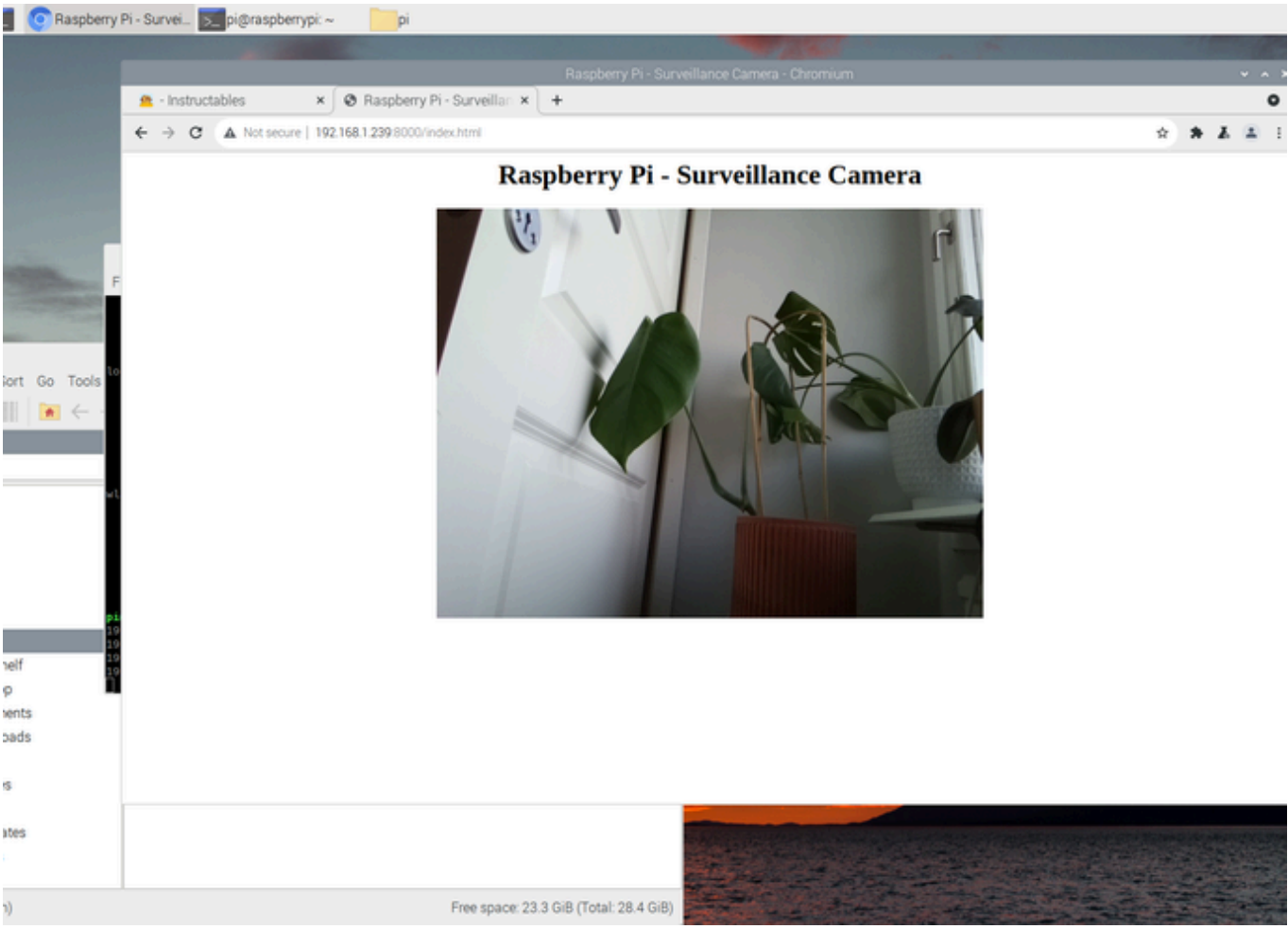
```
sudo apt-get update

sudo apt-get upgrade
```

Step 2: Ρύθμιση Camera Server



AI & SUSTAINABILITY IN VET EDUCATION
ERASMUS 2023-1-LT01-KA220-VET-000155506



Πρώτα πρέπει να ενεργοποιήσετε την κάμερα στο Raspberry Pi. Πληκτρολογήστε την ακόλουθη εντολή στο terminal.

```
pi@raspberry:~ $ sudo raspi-config
```

Επιλέξτε interfacing options και ενεργοποιήστε το camera control.

Όσο βρίσκεστε στις ρυθμίσεις, ενεργοποιήστε και το i2c, γιατί θα το χρειαστούμε αργότερα.

Στη συνέχεια κάντε reboot το Pi.

Τώρα θα δημιουργήσουμε ένα Python script που θα δημιουργεί έναν server με camera stream. Ανοίξτε το Thonny ή τον code editor της επιλογής σας και εισαγάγετε τον ακόλουθο κώδικα:

```
# Web streaming example

# Source code from the official PiCamera package

# http://picamera.readthedocs.io/en/latest/recipes2.html#web-streaming


import io
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>Raspberry Pi - Surveillance Camera</title>
</head>
<body>
<center><h1>Raspberry Pi - Surveillance Camera</h1></center>
<center><img src="stream.mjpg" width="640" height="480"></center>
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace;
boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
            self.send_error(404)
            self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    camera.rotation = 180
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```

Αποθηκεύστε το αρχείο ως camera.py

Η αναφορά για αυτό το κομμάτι κώδικα ανήκει στο Random Nerd Tutorials. Μπορείτε να επισκεφθείτε το https://randomnerdtutorials.com/video-streaming-with-raspberry-pi-camera/#more-41863 για να δείτε την πηγή αυτού του μέρους. Εκεί θα βρείτε και άλλα ενδιαφέροντα projects.

Ανοίξτε ξανά το terminal και πληκτρολογήστε την ακόλουθη εντολή για να ξεκινήσει ο streaming server:

```
pi@raspberrypi:~ $ python3 camera.py
```

Προσέξτε το παρακάτω κομμάτι κώδικα προς το τέλος:

```
camera.rotation = 180
```

Αυτό περιστρέφει την κάμερα κατά 180 μοίρες, κάτι που θα χρειαστεί αργότερα επειδή η κάμερα θα τοποθετηθεί ανάποδα. Αν το video εμφανίζεται με λάθος προσανατολισμό, μπορεί να χρειαστεί να προσαρμόσετε αυτή την τιμή (σε μοίρες).

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Τώρα πρέπει να βρείτε την IP του Pi. Πληκτρολογήστε την ακόλουθη εντολή:

```
pi@raspberry:~ $ ifconfig
```

Η IP του Pi εμφανίζεται στην αρχή του τρίτου block κειμένου, μετά το "inet addr:". Αντιγράψτε την και πληκτρολογήστε την σε έναν web browser, ακολουθούμενη από ":8000", δηλαδή http://<Your_Pi_IP_Address>:8000

Αν εμφανιστεί μια σελίδα με λειτουργικό camera stream, συνεχίστε στο επόμενο βήμα.

Step 3: Δημιουργία Apache Web Server

Αυτό το βήμα σας καθοδηγεί στη διαδικασία εγκατάστασης του Apache web server, στη δημιουργία web page, στη δημιουργία database και στην προσθήκη δυνατότητας επεξεργασίας της database από την web page.

Σε ένα νέο terminal window, εισαγάγετε τον ακόλουθο κώδικα:

```
pi@raspberry:~ $ sudo apt-get install apache2
```

Μεταβείτε στον παρακάτω φάκελο πληκτρολογώντας την εντολή και εμφανίστε τα αρχεία:

```
pi@raspberry:~ $ cd /var/www/html
pi@raspberrypi:/var/www/html $ ls
```

Αν εμφανιστεί ένα αρχείο με όνομα index.html, όλα λειτούργησαν σωστά.

Ανοίξτε έναν νέο web browser και πληκτρολογήστε την IP του Pi όπως πριν, αλλά χωρίς το ":8000" στο τέλος. Αν όλα έχουν γίνει σωστά, θα εμφανιστεί η Apache Debian Default Page. Διαφορετικά, κάτι πήγε στραβά και θα πρέπει να επαναλάβετε το προηγούμενο βήμα.

Για να δοκιμάσετε τον νέο server, ανοίξτε το αρχείο index.html στον text editor της επιλογής σας. Εδώ χρησιμοποιείται το Geany.

Πληκτρολογήστε το κείμενο "Hello World" και αποθηκεύστε το. Κάντε refresh στη web page και θα πρέπει να εμφανιστεί το μήνυμά σας.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 4: Εγκατάσταση PHP

Αν δεν είστε εξοικειωμένοι με την PHP, πρόκειται για programming language που χρησιμοποιείται για το back end των web pages. Εξακολουθεί να χρησιμοποιείται ευρέως, παρότι όλο και περισσότερες σελίδες χρησιμοποιούν server frameworks με JavaScript ή Python.

Η διαδικασία είναι τόσο απλή όσο η πληκτρολόγηση:

```
pi@raspberrypi:/var/www/html $ sudo apt-get install php
```

Όταν ολοκληρωθεί η εγκατάσταση, διαγράψτε το αρχείο index.html πληκτρολογώντας:

```
pi@raspberrypi:/var/www/html $ sudo rm index.html
```

Στη συνέχεια δημιουργήστε ένα νέο αρχείο με όνομα index.php, στον ίδιο φάκελο, με τον ίδιο χαιρετισμό όπως πριν αλλά δημιουργημένο μέσω PHP. Μπορείτε να το κάνετε πληκτρολογώντας:

```
pi@raspberrypi:/var/www/html $ sudo nano index.php
```

και μετά εισαγάγετε τον κώδικα

```
<?php echo "Hello World, but with PHP"; ?>
```

Στη συνέχεια πατήστε CTRL + X και μετά Y για αποθήκευση.

Εναλλακτικά, μπορείτε να δημιουργήσετε ένα νέο αρχείο στο Geany, να εισαγάγετε τον κώδικα και να το αποθηκεύσετε ως index.php. Και οι δύο τρόποι λειτουργούν.

Πληκτρολογήστε την ακόλουθη εντολή στο terminal window για να κάνετε refresh τον server:

```
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
```

Κάντε refresh στο browser window και θα πρέπει να εμφανιστεί το κείμενο που μόλις εισαγάγατε.

Step 5: Εγκατάσταση MySQL

Ήρθε η ώρα να εγκαταστήσετε τη MySQL, μια πολύ δημοφιλή database.

Απλώς πληκτρολογήστε:

```
pi@raspberrypi:/var/www/html $ sudo apt-get install mariadb-server php-mysql
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
pi@raspberrypi:/var/www/html $ sudo mysql_secure_installation
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Θα σας ζητηθεί να ορίσετε ένα root password. Επιλέξτε ένα καλό password και θυμηθείτε το, γιατί θα το χρειαστείτε σε λίγο. Στη συνέχεια θα σας ζητηθεί να το πληκτρολογήσετε ξανά.

Μετά από αυτό θα εμφανιστεί μια σειρά ερωτήσεων. Πατήστε απλώς Y για να απαντήσετε Yes σε όλες. Όταν ολοκληρωθεί η διαδικασία, θα εμφανιστεί το μήνυμα "Thanks for using MariaDB!".

Τώρα θα χρειαστεί να ενημερώσετε τα permissions στα αρχεία του web server. Εισαγάγετε τον ακόλουθο κώδικα στο terminal:

```
pi@raspberrypi:~ $ cd /home
pi@raspberrypi:~ $ ls -lh /var/www/
pi@raspberrypi:~ $ sudo chown -R pi:www-data /var/www/html/
pi@raspberrypi:~ $ sudo chmod -R 770 /var/www/html/
pi@raspberrypi:~ $ ls -lh /var/www/
```

Τώρα όλα είναι έτοιμα. Ώρα να προσθέσετε πιο σύνθετο κώδικα στον νέο server.

Step 6: Δημιουργία Web Page και σύνδεση με Database

Τώρα θα δημιουργήσουμε μια νέα database και έναν table μέσα σε αυτή. Σε αυτόν τον table θα δημιουργήσουμε μία row με 2 columns, όπου θα αποθηκεύεται η θέση των servos που θα ελέγχουν το turret αργότερα.

Ανοίξτε το αρχείο index.php και διαγράψτε όλο τον προηγούμενο κώδικα.

Εισαγάγετε τον ακόλουθο κώδικα στο αρχείο.

```
<?php
$servername = "localhost";
$username = "root";
$password = "raspberry";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
 die("Connection failed: " . $conn->connect_error);
}
echo "";

//Create database camproj2
$sql = "CREATE DATABASE IF NOT EXISTS camproj2";
if ($conn->query($sql) === TRUE) {
   echo "";
} else {
   echo "Error vreating database: " . $conn->error;
}

$conn->close();

//This time we will connect again but directly to the newly created database
$servername = "localhost";
$username = "root";
$password = "raspberry";
$dbname = "camproj2";

//Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
 die("Connection failed: " . $conn->connect_error);
}
echo "";

//Create a table called angles
$sql = "CREATE TABLE IF NOT EXISTS angles (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
pan INT(6) NOT NULL,
tilt INT(6) NOT NULL
)";
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
if ($conn->query($sql) === TRUE) {
   echo "";
} else {
   echo "Error creating table: " . $conn->error;
}

//Create a row in table angles. You only need to do this once
//If yo let this snippet be in the code after the initial creation,
//You will create another row everytime you run the script
$sql = "INSERT INTO angles (pan, tilt) VALUES ('90','90')";

if ($conn->query($sql) === TRUE) {
   echo "";
} else {
   echo "Error: " . $sql . "<br>" . $conn->error;
}

//Check if the HTML form have submitted new values updating current fields
if(!empty(($_POST['pan']))) $pan = $_POST['pan'];
if(!empty(($_POST['tilt']))) $tilt = $_POST['tilt'];
$sql = "UPDATE angles SET pan=$pan, tilt=$tilt WHERE id=1";
$conn->query($sql);

//Checking if the newly stored values are valid and messages the user
otherwise
$sql = "SELECT pan, tilt FROM angles WHERE id=1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   while ($row = $result->fetch_assoc()) {
      $pan = $row["pan"];
      if ($pan <0){
         $pan = 0;
         $panMessage = "Angles must be greater than 0 and smaller than 180";
      } elseif ($pan > 180){
         $pan = 180;
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
            $panMessage = "Angles must be greater than 0 and smaller than 180";
        } else {
            $panMessage = "";
        }

        $tilt = $row["tilt"];
        if ($tilt <0){
            $tilt = 0;
            $tiltMessage = "Angles must be greater than 0 and smaller than 180";
        } elseif ($tilt > 180){
            $tilt = 180;
            $tiltMessage = "Angles must be greater than 0 and smaller than 180";
        } else {
            $tiltMessage = "";
        }
    }
} else {
    echo "0 reults";

}


$conn->close();


//HTML code
echo <<<_END
    <html>
    <img src="http://192.168.1.239:8000/stream.mjpg">

    <h1>
        Pan is: $pan<br>
        Tilt is: $tilt<br>
    </h1>
    <form method="post" action="index.php">
    Pan
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
   <input type="int" name="pan">
   <p>$panMessage</p>
   <br>Tilt
   <input type="int" name="tilt">
   <p>$tiltMessage</p>
   <input type="submit">
   </form>
   </html>
_END;



?>
```

Τα code chunks που έχω σημειώσει ως χρήση μόνο μία φορά χρειάζεται να χρησιμοποιηθούν μόνο μία φορά και μπορούν να αφαιρεθούν από το script μόλις αυτό εκτελεστεί με επιτυχία. Δεν θα επηρεάσει ιδιαίτερα την απόδοση αν τα αφήσετε, οπότε αν δεν αισθάνεστε άνετα με την PHP και δεν θέλετε να τα πειράξετε, μπορείτε να τα αφήσετε. Το χειρότερο που μπορεί να συμβεί είναι να δημιουργείται μία επιπλέον row κάθε φορά που εκτελείτε το script, αλλά επειδή θα χρησιμοποιούμε μόνο την πρώτη, αυτό δεν έχει σημασία.

Κάντε ξανά restart τον Apache server πληκτρολογώντας:

```
pi@raspberry:~ $ sudo service apache2 restart
```

Στη συνέχεια κάντε refresh στο web browser window με την IP address στη γραμμή αναζήτησης.

Αν όλα έχουν γίνει σωστά, θα εμφανιστεί κάτι παρόμοιο με αυτό.

Δοκιμάστε να εισαγάγετε νέες τιμές στα πεδία pan και tilt και πατήστε submit (ή Enter).

Αν το κείμενο ενημερωθεί, τότε η database λειτουργεί.

Παρατηρήστε όμως ότι η κάμερα δεν κινείται ακόμη. Αυτό θα το διορθώσουμε στο επόμενο βήμα.

Step 7: Τοποθέτηση των Motors και της κάμερας στο Turret

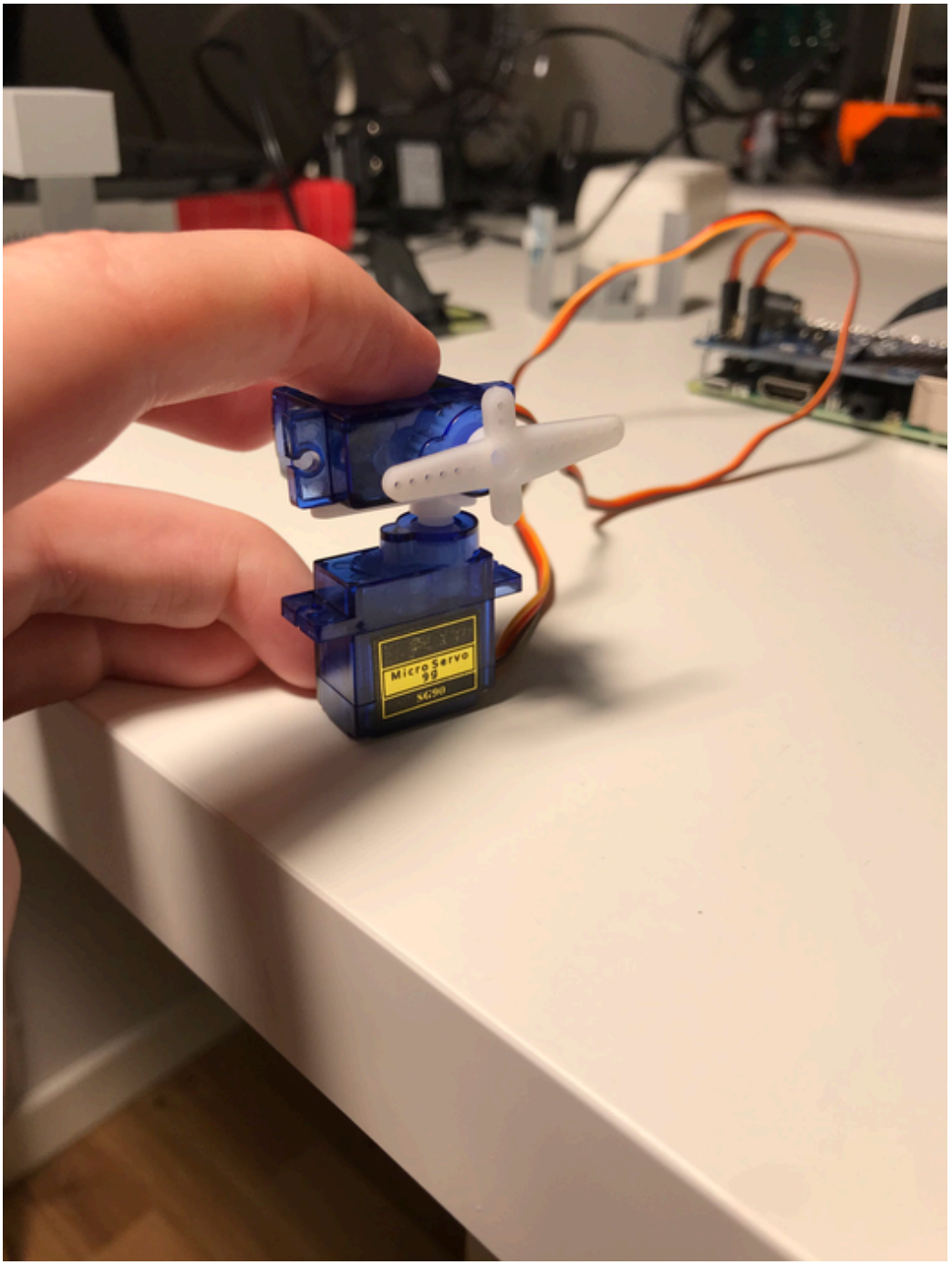
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Co-funded by
the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Αν έχετε πρόσβαση σε 3D printer, μπορείτε να εκτυπώσετε τα διαθέσιμα αρχεία. Αν δεν έχετε 3D printer, θα χρειαστεί να αυτοσχεδιάσετε και να δημιουργήσετε μια παρόμοια κατασκευή με cardboard ή άλλο υλικό. Αν θέλετε μια όσο το δυνατόν πιο minimal λύση, μπορείτε απλώς να κολλήσετε ένα motor στο τραπέζι (ή, προτιμότερα, σε μια σανίδα), να κολλήσετε το δεύτερο motor πάνω σε αυτό και μετά να κολλήσετε την κάμερα στο δεύτερο motor. Το σημαντικό είναι να επιτύχετε κίνηση σε δύο άξονες και να μην συγκρούεται τίποτα κατά την κίνηση.

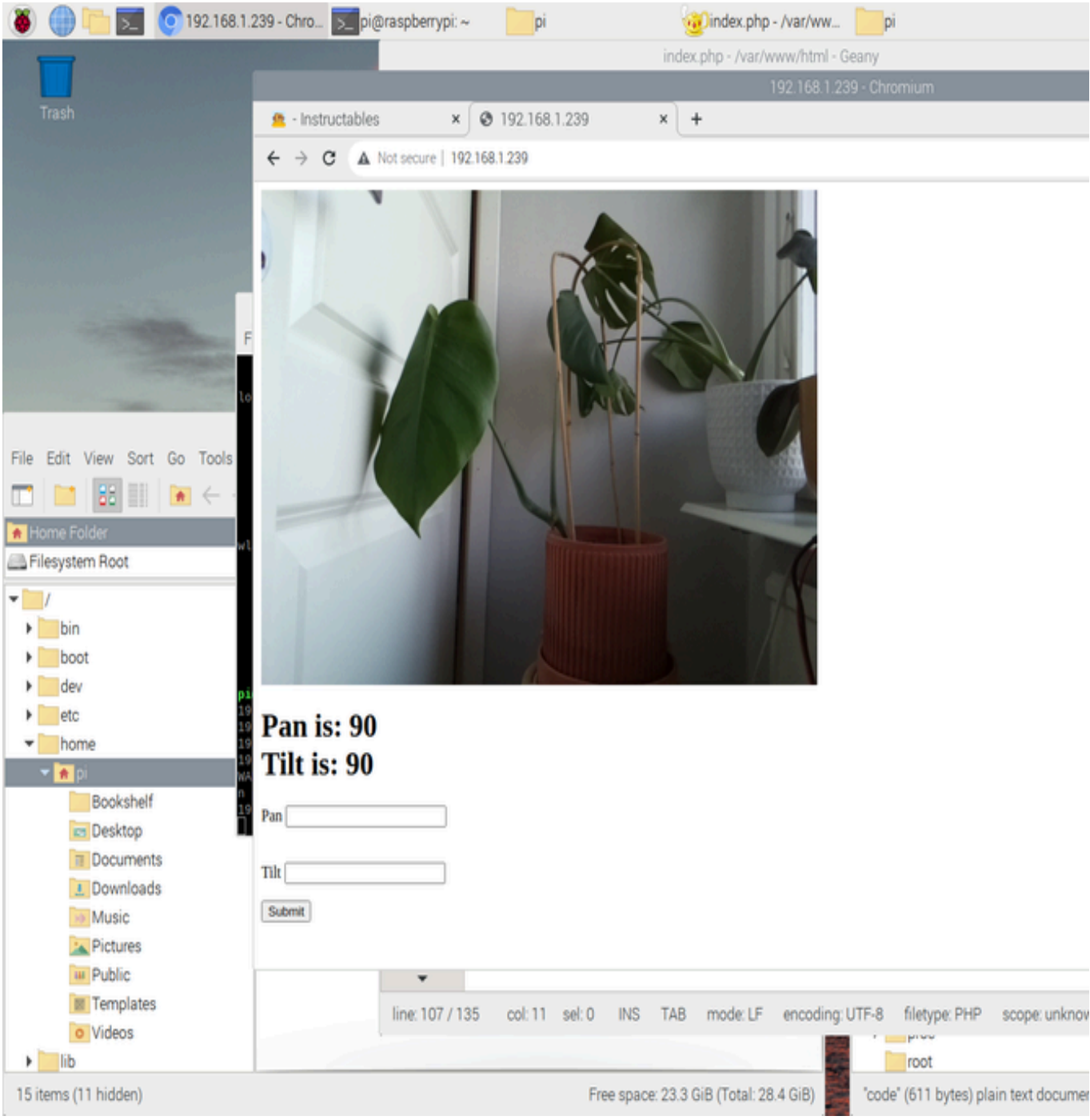
Αν στο τέλος του tutorial όλα λειτουργούν αλλά όταν προσπαθείτε να αλλάξετε το tilt αλλάζει το pan, μπορείτε να το προσαρμόσετε αργότερα στον κώδικα ή απλώς να αλλάξετε ποιο motor κάνει ποια λειτουργία.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Παρατήρησα ότι χρειάστηκε να δέσω το motor με zip tie στο frame, λόγω της ακαμψίας των καλωδίων. Αυτό μπορεί να μην αποτελεί πρόβλημα στη δική σας κατασκευή.

Step 8: Έλεγχος των Servo Motors



Αν δεν έχετε ήδη ενεργοποιήσει το i2c στο Pi, πληκτρολογήστε

```
pi@raspberry:~ $ sudo raspi-config
```

στο terminal και ενεργοποιήστε το i2c, επειδή το Pi χρησιμοποιεί το i2c protocol για να επικοινωνεί με το hat. Στη συνέχεια κάντε reboot το Pi.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI & SUSTAINABILITY IN VET EDUCATION
ERASMUS 2023-1-LT01-KA220-VET-000155506

Πριν γράψουμε τον απαραίτητο Python code, πρέπει να εγκαταστήσουμε ένα package που χρειάζεται ώστε το Pi να μπορεί να ελέγχει τα servos. Στο terminal πληκτρολογήστε την ακόλουθη εντολή

```
pi@raspberry:~ $ sudo pip3 install
adafruit-circuitpython-servokit
```

Στη συνέχεια ανοίξτε ένα νέο αρχείο στο Thonny, ονομάστε το π.χ. motors.py και εισαγάγετε τον ακόλουθο κώδικα.

```
from adafruit_servokit import ServoKit
import mysql.connector
import time

kit = ServoKit(channels = 16)

pan = 90
tilt = 90

#Function for moving motors
def goTo(port, angle, target):
   while angle < target:
       angle += 1
       kit.servo[port].angle = angle
       time.sleep(0.01)
   while angle > target:
       angle -= 1
       kit.servo[port].angle = angle
       time.sleep(0.01)

   return angle

#Main loop
while True:
   #Connecting do database
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
mydb = mysql.connector.connect(
    host = "localhost",
    user = "root",
    password = "raspberry",
    database = "camproj2"
    )

cur = mydb.cursor()

#This is the code which fetches the angles from the database
cur.execute("SELECT pan, tilt FROM angles WHERE id=1")

result = cur.fetchall()

#Checking if angles have changed
if result[0][0] != pan:
    newPan = result[0][0]
    pan = goTo(0, pan, newPan)

if result[0][1] != tilt:
    newTilt = result[0][1]
    tilt = goTo(3, tilt, newTilt)

#Important to close database after operation
cur.close()
mydb.close()
```

Βεβαιωθείτε ότι το power cable είναι συνδεδεμένο στο servo hat, διαφορετικά δεν θα μπορεί να ελέγξει τα motors.

Τώρα μεταβείτε ξανά στον web browser και εισαγάγετε νέες τιμές για τις γωνίες pan και tilt. Τα motors (τουλάχιστον τα πιο συνηθισμένα) έχουν περιστροφή 180

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

μοιρών, οπότε το script τα ορίζει στις 90 μοίρες αν δεν δοθεί άλλη τιμή από τη database.

Αν όλα έχουν γίνει σωστά, η κάμερα θα κινηθεί και θα μπορείτε να τη δείτε στη web page. Αυτό λειτουργεί σε οποιαδήποτε συσκευή είναι συνδεδεμένη στο ίδιο network με το Pi. Μπορείτε να χρησιμοποιήσετε port forwarding για να έχετε πρόσβαση στη σελίδα από άλλες συσκευές, αλλά αυτό είναι πιο προχωρημένο και δεν προτείνεται, επειδή μπορεί να αφήσει το network σας ευάλωτο. Αν παρ' όλα αυτά θέλετε να το δοκιμάσετε, το επόμενο βήμα εξηγεί σύντομα πώς γίνεται.

- **Web Controlled Camera Turret - Step #8**Web Controlled Camera Turret - Step #8

  DownloadΛήψη

Step 9: (Προχωρημένο) Port Forward για πρόσβαση στη σελίδα από οποιαδήποτε συσκευή

Σημειώστε ότι σε αυτό το project δεν υπάρχει authentication ή άλλη μορφή ασφάλειας, οπότε οποιοσδήποτε αποκτήσει την IPv4 address σας μπορεί να δει το camera feed. Κάντε αυτό το βήμα με δική σας ευθύνη.

Αυτό το βήμα εξαρτάται αρκετά από το router που χρησιμοποιείτε, οπότε εδώ εξηγείται η διαδικασία για Netgear router.

Ανοίξτε ένα νέο browser window, πληκτρολογήστε την address 192.168.1.1 και συνδεθείτε με τα δικά σας credentials.

Πατήστε το button "WAN" στην αριστερή στήλη και έπειτα πατήστε "Virtual server / Port forwarding" στην επάνω μπάρα.

Στη συνέχεια πατήστε το button "Add profile"

Μπορείτε να αφήσετε τα περισσότερα πεδία κενά ή όπως είναι, αλλά βεβαιωθείτε ότι το protocol έχει οριστεί σε TCP. Στο πεδίο "Service name" εισαγάγετε κάτι όπως "Camera control server" ή ένα όνομα της επιλογής σας, ώστε να αναγνωρίζετε εύκολα την εφαρμογή.

Στο πεδίο "External Port" εισαγάγετε κάτι όπως 8001 (ή οποιοδήποτε port number που δεν χρησιμοποιείται, αφού έχουμε χρησιμοποιήσει το 8000 για τον camera server, μην χρησιμοποιήσετε αυτό).

Εισαγάγετε την IP address του Pi στο πεδίο "Internal IP address" και πατήστε OK.

Τώρα πρέπει να μάθετε την IPv4 address του network σας. Αυτό ξεπερνά λίγο την παρούσα καθοδήγηση και ίσως υπάρχουν καλύτεροι ή ευκολότεροι τρόποι, αλλά ένας απλός τρόπος είναι να αναζητήσετε στο Google "what is my IP address" και να την αντιγράψετε από εκεί. Έπειτα μπορείτε να πληκτρολογήσετε αυτή την IP address σε browser, ακολουθούμενη από ":8001" (αν αυτό είναι το port που δηλώσατε προηγουμένως).

Τώρα πρέπει να τροποποιήσετε λίγο τον κώδικα στο index.php: αντικαταστήστε την προηγούμενη address από την οποία λαμβανόταν το camera feed (η IP address του Pi ακολουθούμενη από :8000) με την IP address του network σας ακολουθούμενη από :8000.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Τώρα θα πρέπει να μπορείτε να έχετε πρόσβαση στο camera feed από οποιαδήποτε συσκευή στο Internet χρησιμοποιώντας αυτή την address, να βλέπετε το feed και να ελέγχετε την κάμερα. Σημειώστε ότι το Pi μπορεί να δυσκολευτεί με το streaming σε αυτή τη ρύθμιση και το framerate μπορεί να μειωθεί.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

11. Κριτήρια αξιολόγησης:

Κριτήριο	Περιγραφή
Μηχανική συναρμολόγηση	Σωστή συναρμολόγηση του camera turret
Electronics	Σωστή καλωδίωση servo motor
Programming	Το web interface επικοινωνεί με το Raspberry Pi
Λειτουργικότητα	Η κάμερα κινείται και κάνει επιτυχώς stream το video
Καινοτομία	Πρόσθετα AI ή automation features

---

12. Διαθέσιμες γλώσσες:

---

13. Συντελεστές:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Feedback και συνεχής βελτίωση:

AI object tracking

facial recognition security system

smart home monitoring

integration with mobile apps

.

Σημειώσεις υποβολής:

Βεβαιωθείτε ότι όλα τα απαιτούμενα έγγραφα και οι πόροι έχουν επισυναφθεί στην υποβολή σας. Η ομάδα αξιολόγησης του AI-SAVES@ProjectHub θα αξιολογήσει την υποβολή με βάση τις παρεχόμενες πληροφορίες. Μετά την έγκριση, το project θα δημοσιευθεί στο hub και οι partners θα ενημερωθούν.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.