2025-1-RO01-KA121-VET-000319082

4IT Erasmus+Project, 2025-1-RO01-KA121-VET-000319082

Beneficiary: "Gheorghe Vranceanu" National College, Bacau

Sending organization: "Roman Voda" National College, Bacau /

"Ferdinand I" National College, Bacau

MOBILITY PORTFOLIO

Valencia, Spain/ Paphos/Cyprus

Mobility period 28.10.2025- 10.11.2025

Practice period: 29.10.2025-08.11.2025

RECEIVING COMPANY: LAZAROU ICT PROFESSIONALS

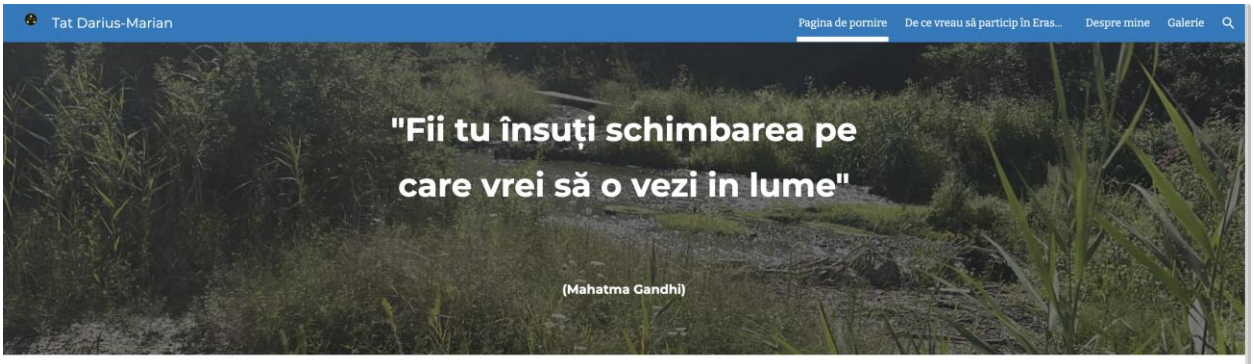
Practice tutor: Lazaros Lazarou

Participant: Tat Darius-Marian

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082

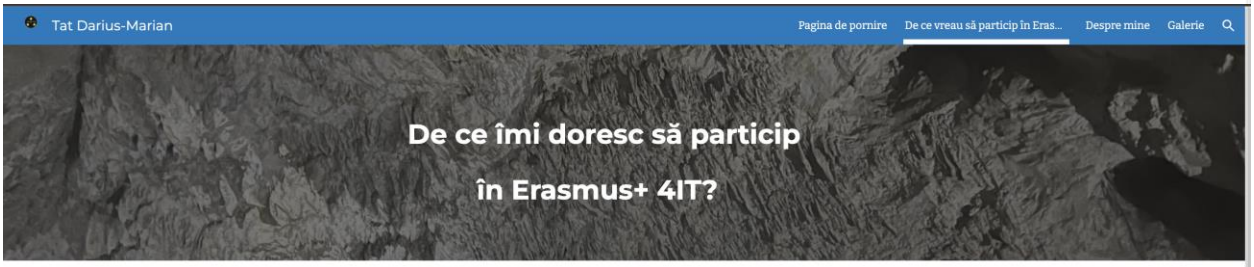
Site-ul meu de selectie: Tat Darius-Marian



Bună ziua!

Mă numesc Tat Darius-Marian, am 17 ani și sunt elev în clasa XI-a B al Colegiului Național "Roman-Vodă", Roman, profil matematică-informatică, intensiv și îmi doresc să particip la proiectul Erasmus+ 4IT. Consider ca sunt o persoană sociabila, ambițioasă, dornică să învețe lucruri noi mereu.

Eu privesc informatica ca pe o carte deschisă care așteaptă cu nerăbdare să fie citită, explorată, folosită. Sunt de părere că fiecare persoană are un scop în această lume, iar eu simt ca menirea mea este informatica.



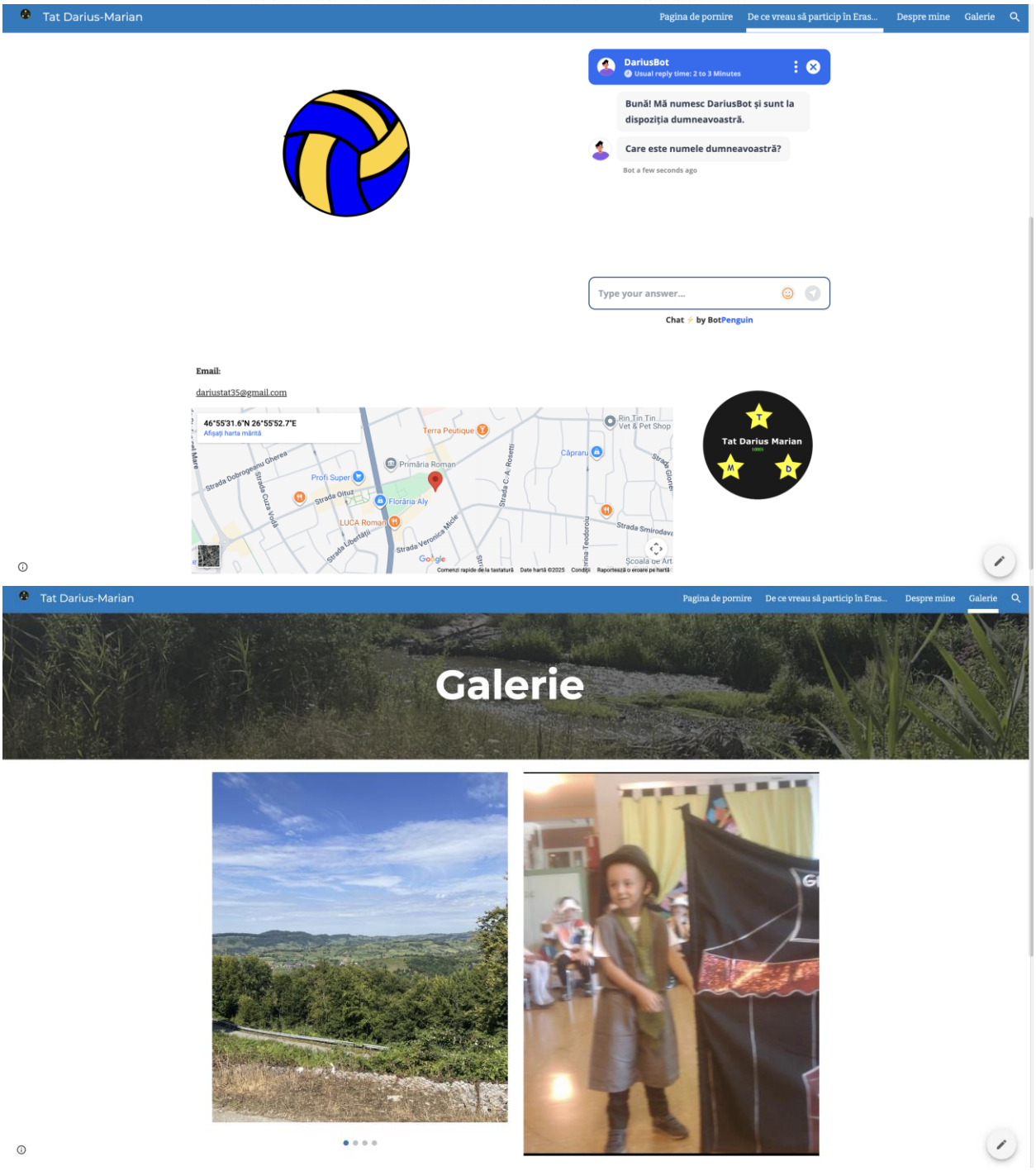
Încă din clasa a X-a când elevii participanți anul trecut au prezentat activitățile desfășurate, am simțit că această experiență este pentru mine, deoarece consider că acest proiect nu constă doar într-o mobilitate, ci este o oportunitate pentru dezvoltarea mea personală. Această experiență nu ne oferă doar șansa de a crește din punct de vedere al informaticii, ci ne oferă ocazia de a descoperi o cultură nouă, oameni noi, obiceiuri diferite.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082



Pregătire culturală:

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082

2025-1-RO01-KA121-VET-000319082

Stăpânirea bizantină și perioada venețiană

realizat de: Tat Darius-Marian



Context:

După împărțirea Imperiului Roman în jumătatea de est și cea de vest, Cipru a intrat sub stăpânirea Imperiului Roman de Răsărit. În acea perioadă, episcopul său, deși era încă sub autoritatea Bisericii Creștine, a fost declarat autocefal (independent) de către Primul Sinod de la Efes, în anul 431.



DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082

Musulmanii arabi au invadat Ciprul în forță în anii 650, dar în anul 688, împăratul bizantin Iustinian al II-lea și califul omeiad ʿAbd al-Malik ibn Marwān au ajuns la un acord fără precedent. În următorii 300 de ani, Ciprul a fost guvernat în comun de arabi și bizantini, ca un condominiu, în ciuda războaielor aproape continue dintre cele două puteri pe continent. Bizantinii au recâștigat controlul asupra insulei pentru perioade scurte după aceea, dar situația anterioară era mereu restabilită.



Această perioadă a durat până în anul 965, când Niketas Chalkoutzes a cucerit insula pentru Imperiul Bizantin aflat în renaștere. În 1185, ultimul guvernator bizantin al Ciprului, Isaac Comnenul din Cipru, provenit dintr-o ramură minoră a familiei imperiale, s-a răsculat și a încercat să uzurpe tronul. Lovitura sa de stat a eșuat, dar Comnenul a reușit totuși să păstreze controlul asupra insulei. Acțiunile bizantinilor împotriva lui au eșuat deoarece acesta se bucura de sprijinul regelui Wilhelm al II-lea al Siciliei. Împăratul bizantin avea un acord cu sultanul Egiptului pentru a închide porturile cipriote cruciaților.



Chalkoutzes și însoțitorii săi evadează în timpul unei bătălii între bizantini și arabi. Miniatură din manuscrisul Madrid Skylitzes.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

Stăpânirea Veneţiană

După o perioadă de dominație a familiei franceze Lusignan (1192–1489), insula Cipru a trecut sub controlul Republicii Veneția în anul 1489. Aceasta s-a întâmplat după ce Ecaterina Cornaro, ultima regină a Ciprului și de origine venețiană, a fost forțată să abdice și să cedeze insula statului venețian. Astfel, Ciprul a devenit o colonie strategică aflată sub controlul direct al Veneției.

emându-se de expansiunea Imperiului Otoman, venețienii au investit masiv în fortificarea insulei.

Au reconstruit zidurile capitalei Nicosia în stil renascentist italian (cu 11 bastioane în formă de stea) și au întărit orașul-port Famagusta, care a devenit una dintre cele mai solide fortărețe ale Mediteranei.

Aceste lucrări aveau scopul de a transforma Ciprul într-un bastion de apărare împotriva otomanilor.



Hartă a orașului Nicosia din Cipru, realizată de cartograful venețian Giacomo Franco, care arată zidurile venețiene ale Nicosiei, construite de venețieni pentru a apăra orașul în cazul unui atac otoman.

Tradiţii ale culturii

1 Sărbători și festivaluri

- Kataklysmos (Festivalul Potopului): O sărbătoare legată de purificare, reînnoire și renaștere, cu rădăcini în cultul apei, care include muzică, dansuri și concursuri.



- Unul dintre cele mai mari festivaluri de vin din Mediterana, unde cramele locale își prezintă cele mai bune vinuri. Vizitatorii pot participa la degustări, spectacole de muzică tradițională și demonstrații de tărire a strugurilor



DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082

Tradiții ale culturii

2 Gastronomia cipriotă

- Meze - Este o selecție de mai multe tipuri de mâncăruri. Se servește pe o farfurie mare sau în jurul mesei și există câte ceva din toate. De la pui, măsline negre, salată până la caracatiță. Meze se servește pentru 2 sau mai multe persoane, deoarece este foarte multă mâncare



- Brânză Halloumi - Este un tip unic de brânză care se produce numai în Cipru. Este făcută din lapte de vacă sau de oaie și poate fi savurată, de exemplu, la grătar, prăjită la tigaie acasă sau coaptă într-un sandwich.



Personalități

1 Marcos Baghdatis

Marcos Baghdatis (greacă: Μάρκος Παγδατής, pronunțat ['markos paɣða'tis]) (n. 17 iunie 1985, Limassol, Cipru) este un jucător profesionist cipriot de tenis de origine libaneză, fost finalist la Australian Open.



DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082

Personalități

2 Makarios al III-lea



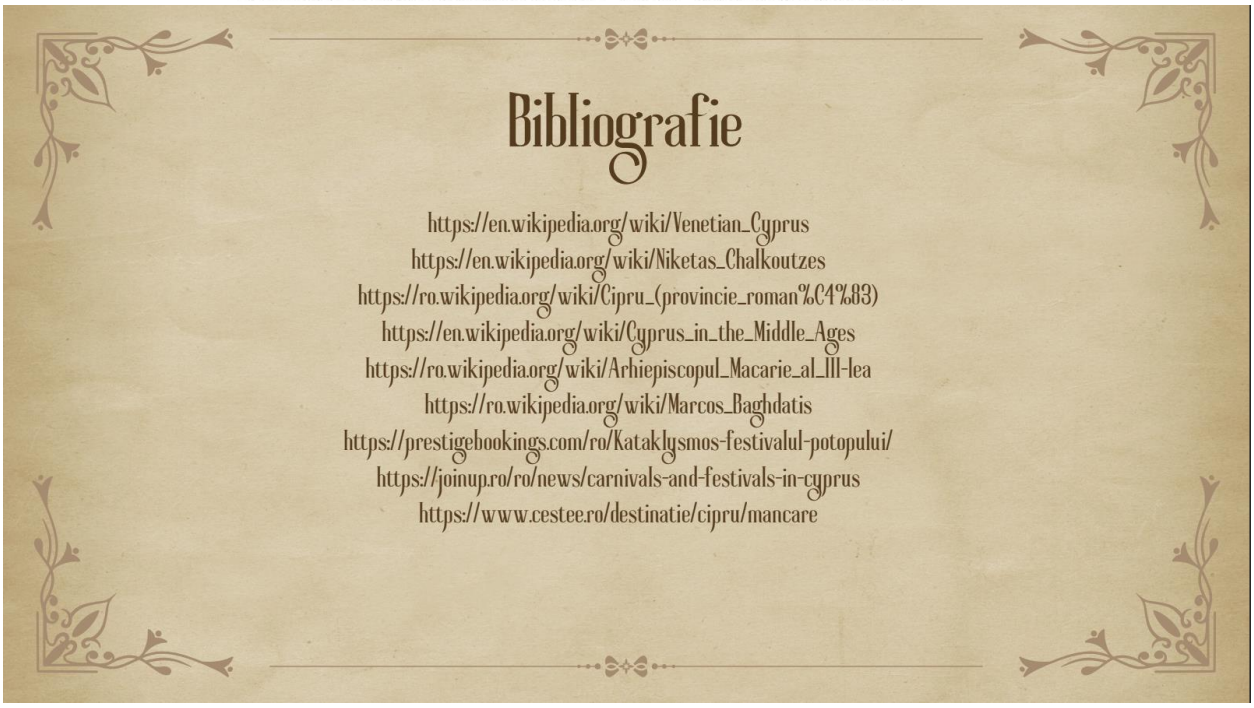
Makarios al III-lea (în limba greacă: Μακάριος Γ), născut Mihail Christodoulou Mouskos (în limba greacă: Μιχαήλ Χριστοδούλου Μούσκος) (n. 13 august 1913, Pano Panagia Pafou[d], Eparchia Pafou, Cipru — d. 3 august 1977, Nicosia, Cipru de Nord), a fost arhiepiscop și primat al Bisericii Ortodoxe a Ciprului (1950-1977) și primul președinte al Republicii Cipru (1960-1974 și 1974-1977).

Concluzie

Cipru este o insulă cu o istorie bogată și o cultură unică, modelată de numeroase civilizații care au trecut pe teritoriul său. În perioada bizantină, insula a devenit un important centru al creștinismului ortodox, păstrând până astăzi biserici și mănăstiri valoroase, iar în epoca venețiană, s-au dezvoltat fortificațiile și arhitectura. Astfel, Cipru rămâne un loc unde istoria, cultura și oamenii formeaza o legătură puternică, recunoscută în întreaga lume.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082



Secțiunea 2: Activități practice

1.

```
int main() {
    int MAXSIZE = 3;
    float *exam1 = new float[MAXSIZE], *exam2 = new float[MAXSIZE], *exam3 = new float[MAXSIZE], *avg = new float[MAXSIZE];

    for (int i = 0; i < MAXSIZE; i++) {
        cout << "Student " <<                xam 1: "; cin >> exam1[i];
        cout << "Exam 2: "; c   float *exam2   ;
        cout << "Exam 3: "; c   exam2          ;
        avg[i] = (exam1[i] + exam2[i] + exam3[i]) / 3;
    }

    sort_crescator(exam1, MAXSIZE);
    sort_crescator(exam2, MAXSIZE);
    sort_crescator(exam3, MAXSIZE);

    float lowest = exam1[0], highest = exam1[MAXSIZE - 1];
    if (exam2[0] < lowest) lowest = exam2[0];
    if (exam3[0] < lowest) lowest = exam3[0];
    if (exam2[MAXSIZE - 1] > highest) highest = exam2[MAXSIZE - 1];
    if (exam3[MAXSIZE - 1] > highest) highest = exam3[MAXSIZE - 1];

    cout << "\nCea mai mica nota: " << lowest << "\nCea mai mare nota: " << highest << "\n\n";

    for (int i = 0; i < MAXSIZE; i++)
        cout << "Exam1: " << exam1[i] << " Exam2: " << exam2[i]
             << " Exam3: " << exam3[i] << " Avg: " << avg[i] << '\n';

    delete[] exam1; delete[] exam2; delete[] exam3; delete[] avg;
}
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

2.

```cpp
#include <iostream>
using namespace std;

int main() {
    int ARRAYSIZE = 0;
    int tmpN = 0;
    int n = 0;

    float* exam1;
    float* exam2;
    float* temp1;
    float* temp2;

    cout << "How many students would you like to enter? ";
    cin >> ARRAYSIZE;

    exam1 = new float[ARRAYSIZE];
    exam2 = new float[ARRAYSIZE];

    do {
        cout << "Enter exam 1 result for student " << n + 1 << ": ";
        cin >> exam1[n];
        cout << "Enter exam 2 result for student " << n + 1 << ": ";
        cin >> exam2[n];
        cout << "Student saved!" << endl;
        n++;

        if (n >= ARRAYSIZE) {
            cout << "Out of Space!!! Would you like to enter more students?" << endl;
            cout << "Please enter how many more (0 to Exit): ";
            cin >> tmpN;

            if (tmpN > 0) {
                ARRAYSIZE = ARRAYSIZE + tmpN;
                temp1 = new float[ARRAYSIZE];
                temp2 = new float[ARRAYSIZE];

                   for (int i = 0; i < n; i++) {
                    temp1[i] = exam1[i];
                    temp2[i] = exam2[i];
                }

                delete[] exam1;
                delete[] exam2;

                exam1 = temp1;
                exam2 = temp2;
            }
        }

    } while (n < ARRAYSIZE);

    cout << "\n--- Student Results ---\n";
    for (int i = 0; i < n; i++) {
        float avg = (exam1[i] + exam2[i]) / 2;
        cout << "Student " << i + 1 << endl;
        cout << "Exam1: " << exam1[i] << endl;
        cout << "Exam2: " << exam2[i] << endl;
        cout << "Average = " << avg << endl;
    }
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
} while (n < ARRAYSIZE);

cout << "\n--- Student Results ---\n";
for (int i = 0; i < n; i++) {
    float avg = (exam1[i] + exam2[i]) / 2;
    cout << "Student " << i + 1 << endl;
    cout << "Exam1: " << exam1[i] << endl;
    cout << "Exam2: " << exam2[i] << endl;
    cout << "Average = " << avg << endl;
}

delete[] exam1;
delete[] exam2;

return 0;
```

3.

```
#include <iostream>
#include <string>
using namespace std;

class Car {
private:
    string model;
    string brand;
    int hp;
    int year;

public:
    void read() {
        cout << "Enter the car brand: ";
        getline(cin, brand);
        cout << "Enter the car model: ";
        getline(cin, model);
        cout << "Enter the car HP: ";
        cin >> hp;
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

```
        cout << "Enter the car year: ";
        cin >> year;
        cin.ignore();
    }

    void output() {
        cout << "\n--- Car specifications ---\n";
        cout << "Brand: " << brand << "\n";
        cout << "Model: " << model << "\n";
        cout << "HP: " << hp << "\n";
        cout << "Year: " << year << "\n";
    }
};

int main() {
    int ARRAYSIZE, n = 0, tmpN;
    Car* cars;
    Car* temp;

    cout << "Enter how many cars you want to add: ";
    cin >> ARRAYSIZE;
    cin.ignore();

    cars = new Car[ARRAYSIZE];

    do {
        cout << "\nCar number " << n + 1 << ":\n";
        cars[n].read();
        cout << "Car saved!\n";
        n++;

        if (n >= ARRAYSIZE) {
            cout << "\nOut of space! Would you like to enter more cars?\n";
            cout << "Enter how many more (0 to exit): ";
            cin >> tmpN;
            cin.ignore();

            if (tmpN > 0) {
                ARRAYSIZE += tmpN;
                temp = new Car[ARRAYSIZE];

                // Copiem mașinile existente
                for (int i = 0; i < n; i++) {
                    temp[i] = cars[i];
                }

                delete[] cars;
                cars = temp;
            }
        }

    } while (n < ARRAYSIZE);

    cout << "\n=== List of all cars entered ===\n";
    for (int i = 0; i < n; i++) {
        cout << "\nCar number " << i + 1 << " details:\n";
        cars[i].output();
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4.

```cpp
#include <iostream>
#include <string>
using namespace std;

class Car {
private:
    string model;
    string brand;
    int hp;
    int year;

public:
    void read() {
        cout << "Enter the car brand: ";
        getline(cin, brand);
        cout << "Enter the car model: ";
        getline(cin, model);
        cout << "Enter the car HP: ";
        cin >> hp;
        cout << "Enter the car year: ";
        cin >> year;
        cin.ignore();
    }

    void output() {
        cout << "\n--- Car specifications ---\n";
        cout << "Brand: " << brand << "\n";
        cout << "Model: " << model << "\n";
        cout << "HP: " << hp << "\n";
        cout << "Year: " << year << "\n";
    }
};

int main() {
    Car cars[100];
    int number_cars;

    cout << "Enter the number of cars: ";
    cin >> number_cars;
    cin.ignore();

    for (int i = 0; i < number_cars; i++) {
        cout << "\nCar number " << i + 1 << ":\n";
        cars[i].read();
    }

    for (int i = 0; i < number_cars; i++) {
        cout << "\nCar number " << i + 1 << " details:\n";
        cars[i].output();
    }

    return 0;
}

```

5.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

```cpp
#include <iostream>
using namespace std;

class patient {
private:
    string name;
    int age;
    string phone;
    string medicine;

public:
    void setName(string n) { name = n; }
    void setAge(int a) { age = a; }
    void setPhone(string p) { phone = p; }
    void setMedicine(string m) { medicine = m; }

    string getName() { return name; }
    int getAge() { return age; }
    string getPhone() { return phone; }
    string getMedicine() { return medicine; }
};

// Funcții auxiliare
void addPatient(patient *cl, int index);
patient *growArray(patient *cl, int arraySize, int index);
void printArray(patient *clinic, int index);
void searchByName(patient *clinic, int index, string name);
void searchByMedicine(patient *clinic, int index, string medicine);

int main() {
    int ArraySize = 1;
    int index = 0;
    patient *clinic;
    clinic = new patient[ArraySize];
    int userChoice = 1;

    while (userChoice > 0) {
        cout << "1. Add a new user" << endl;
        cout << "2. Edit a User" << endl;
        cout << "3. Delete a User" << endl;
        cout << "4. Search by Name" << endl;
        cout << "5. Search by Medicine" << endl;
        cout << "6. Print The Array" << endl;
        cout << "0. Exit" << endl;
        cout << "What would you like to do? ";
        cin >> userChoice;
        cout << "------------------------------" << endl;
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

```
        if (userChoice == 1) {
            cout << "Add a new Patient" << endl << "------------------" << endl;
            if (index >= ArraySize) {
                ArraySize = ArraySize + 1;
                clinic = growArray(clinic, ArraySize, index);
            }
            addPatient(clinic, index);
            index++;
        }
        else if (userChoice == 2) {
            cout << "Edit a Patient (not implemented yet)" << endl;
        }
        else if (userChoice == 3) {
            cout << "Delete a Patient (not implemented yet)" << endl;
        }
        else if (userChoice == 4) {
            cout << "Search By Name" << endl << "------------------" << endl;
            string name;
            cout << "Enter name to search: ";
            cin >> name;
            searchByName(clinic, index, name);
        }
        else if (userChoice == 5) {
            cout << "Search By Medicine" << endl << "------------------" << endl;
            string med;
            cout << "Enter medicine to search: ";
            cin >> med;
            searchByMedicine(clinic, index, med);
        }
        else if (userChoice == 6) {
            printArray(clinic, index);
        }
        else if (userChoice == 0) {
            cout << "Thank you for using the program!" << endl;
        }
        else {
            cout << "Wrong Choice! Try again" << endl;
        }
    }

    delete[] clinic;
    return 0;
}

patient *growArray(patient *clinic, int ArraySize, int index) {
    patient *tmp;
    tmp = new patient[ArraySize];
    for (int i = 0; i < index; i++) {
        tmp[i].setAge(clinic[i].getAge());
        tmp[i].setMedicine(clinic[i].getMedicine());
        tmp[i].setName(clinic[i].getName());
        tmp[i].setPhone(clinic[i].getPhone());
    }
    delete[] clinic;
    return tmp;
}
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

```cpp

void addPatient(patient *cl, int index) {
    string name, phone, med;
    int age;

    cout << "Name: ";
    cin >> name;
    cl[index].setName(name);
    cout << "Age: ";
    cin >> age;
    cl[index].setAge(age);
    cout << "Phone: ";
    cin >> phone;
    cl[index].setPhone(phone);
    cout << "Medicine Given: ";
    cin >> med;
    cl[index].setMedicine(med);
    cout << "Patient Successfully Added!" << endl;
}

void printArray(patient *clinic, int index) {
    cout << "PRINTING ALL DATA" << endl << "-----------------" << endl;
    for (int i = 0; i < index; i++) {
        cout << "----- Patient " << i + 1 << " -----" << endl;
        cout << "Name: " << clinic[i].getName() << endl;
        cout << "Age: " << clinic[i].getAge() << endl;
        cout << "Phone: " << clinic[i].getPhone() << endl;
        cout << "Medicine: " << clinic[i].getMedicine() << endl;
    }
}

/*-----------------------------------
    SEARCH FUNCTIONS
-----------------------------------*/

void searchByName(patient *clinic, int index, string name) {
    bool found = false;
    for (int i = 0; i < index; i++) {
        if (clinic[i].getName() == name) {
            cout << "Patient Found!" << endl;
            cout << "Name: " << clinic[i].getName() << endl;
            cout << "Age: " << clinic[i].getAge() << endl;
            cout << "Phone: " << clinic[i].getPhone() << endl;
            cout << "Medicine: " << clinic[i].getMedicine() << endl;
            found = true;
        }
    }
    if (!found)
        cout << "No patient found with the name \"" << name << "\"." << endl;
}

```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

```cpp
void searchByName(patient *clinic, int index, string name) {
    bool found = false;
    for (int i = 0; i < index; i++) {
        if (clinic[i].getName() == name) {
            cout << "Patient Found!" << endl;
            cout << "Name: " << clinic[i].getName() << endl;
            cout << "Age: " << clinic[i].getAge() << endl;
            cout << "Phone: " << clinic[i].getPhone() << endl;
            cout << "Medicine: " << clinic[i].getMedicine() << endl;
            found = true;
        }
    }
    if (!found)
        cout << "No patient found with the name \"" << name << "\"." << endl;
}

void searchByMedicine(patient *clinic, int index, string medicine) {
    bool found = false;
    for (int i = 0; i < index; i++) {
        if (clinic[i].getMedicine() == medicine) {
            cout << "Patient Found!" << endl;
            cout << "Name: " << clinic[i].getName() << endl;
            cout << "Age: " << clinic[i].getAge() << endl;
            cout << "Phone: " << clinic[i].getPhone() << endl;
            cout << "Medicine: " << clinic[i].getMedicine() << endl;
            found = true;
        }
    }
    if (!found)
        cout << "No patient found taking \"" << medicine << "\"." << endl;
}
```

6.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

```arduino
const int trigPin = 9;
const int echoPin = 10;
const int pinLed = 13;

float duration, distance;

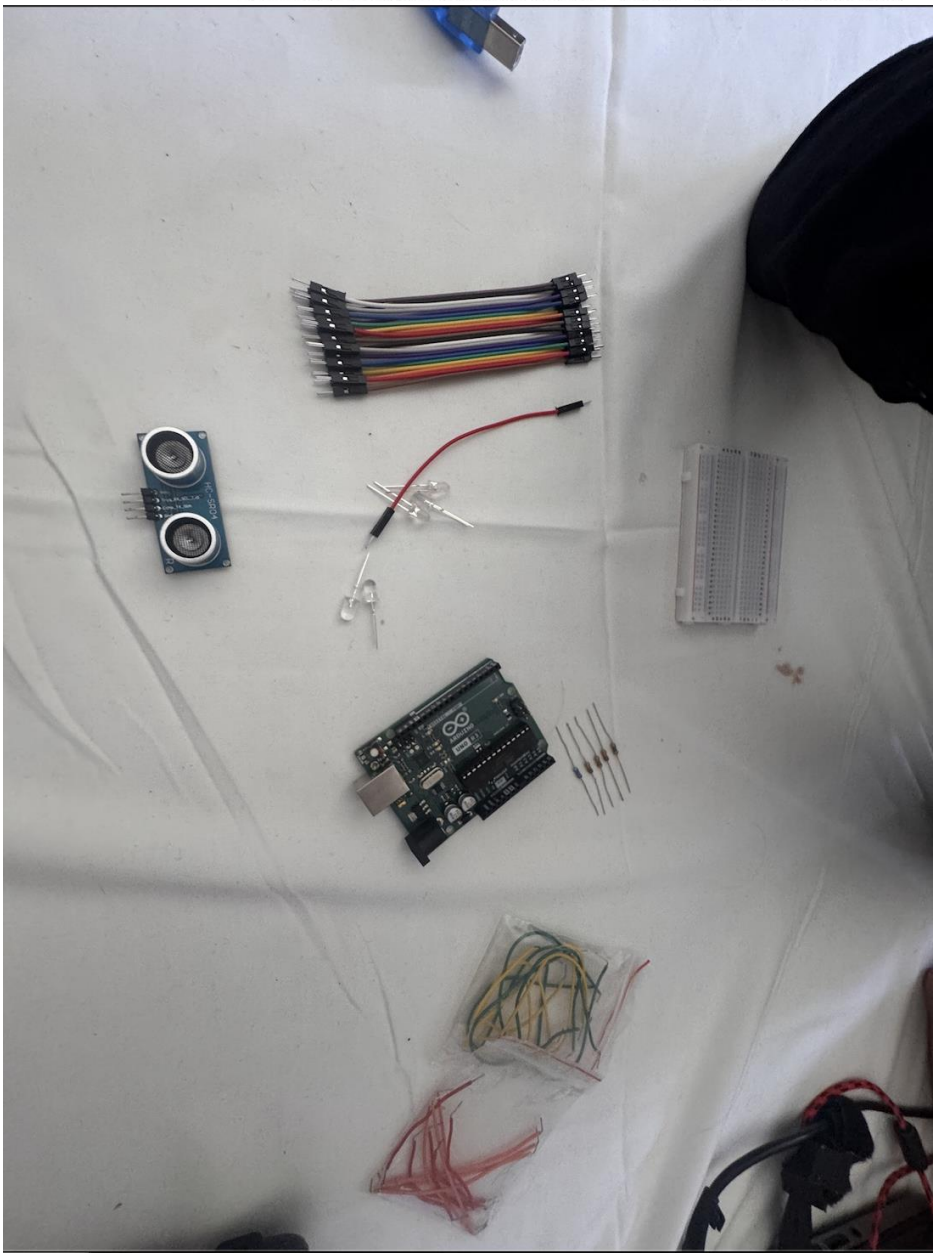
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(pinLed, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
  if(distance < 30){
    digitalWrite([pinLed,HIGH);
  }
  else{
    digitalWrite(pinLed, LOW);
  }
}
```

program **Arduino**, care folosește un **senzor ultrasonic** (de tip HC-SR04) pentru a măsura distanța față de un obiect și aprinde un **LED** când obiectul este la o distanta mai mica de 30 cm față de obiect.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



7.

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

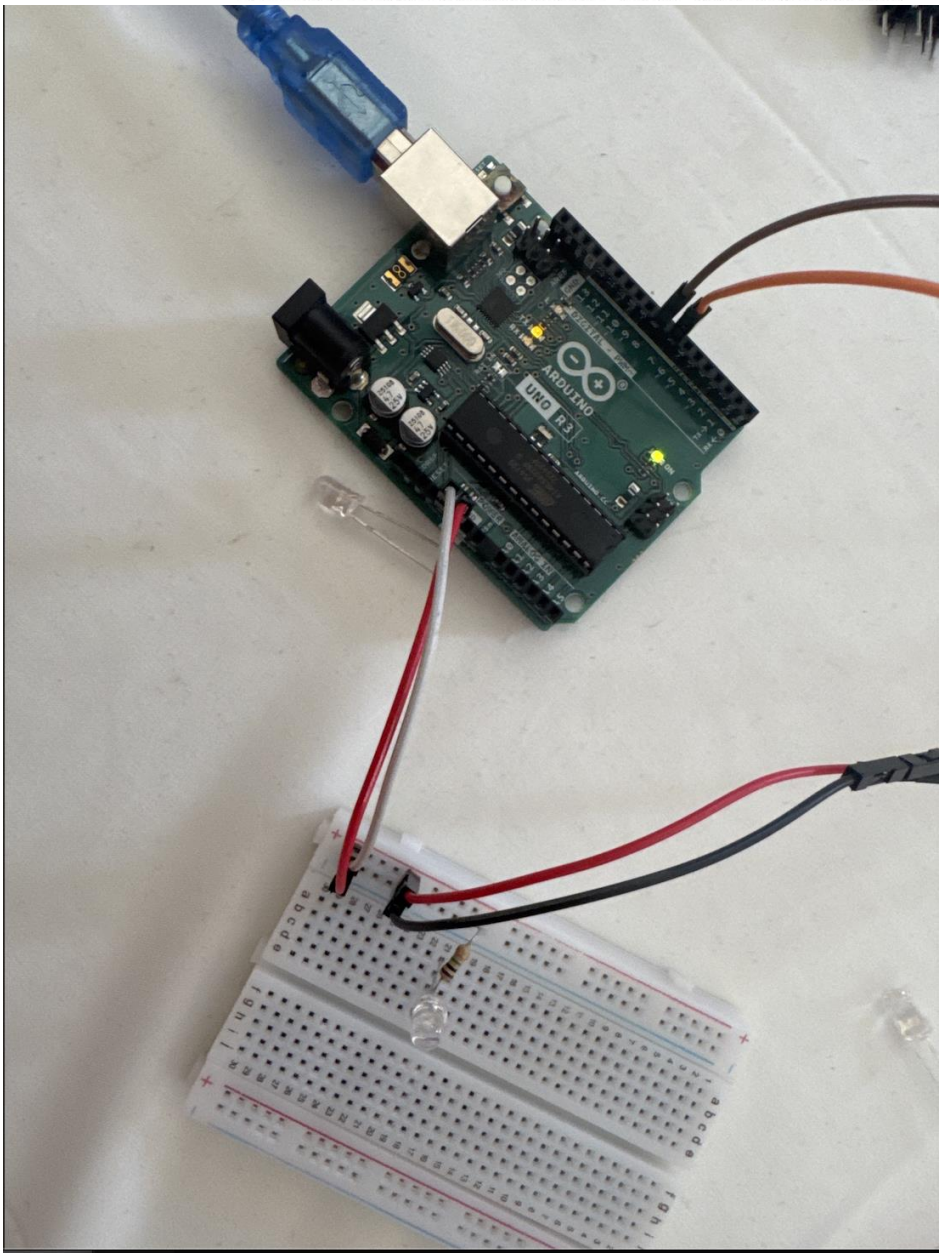
Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

```
1
2   // the setup function runs once when you press reset or power the board
3   void setup() {
4     // initialize digital pin LED_BUILTIN as an output.
5     pinMode(LED_BUILTIN, OUTPUT);
6   }
7
8   // the loop function runs over and over again forever
9   void loop() {
10    digitalWrite(LED_BUILTIN, HIGH);  //short blink
11    delay(300);
12    digitalWrite(LED_BUILTIN, LOW);
13    delay(100);
14    digitalWrite(LED_BUILTIN, HIGH);  //short blink
15    delay(300);
16    digitalWrite(LED_BUILTIN, LOW);
17    delay(100);
18    digitalWrite(LED_BUILTIN, HIGH);  //short blink
19    delay(300);
20    digitalWrite(LED_BUILTIN, LOW);
21    delay(100);
22    digitalWrite(LED_BUILTIN, HIGH);  //long blink
23    delay(600);
24    digitalWrite(LED_BUILTIN, LOW);
25    delay(100);
26    digitalWrite(LED_BUILTIN, HIGH);  //long blink
27    delay(600);
28    digitalWrite(LED_BUILTIN, LOW);
29    delay(100);
30    digitalWrite(LED_BUILTIN, HIGH);  //long blink
31    delay(600);
32    digitalWrite(LED_BUILTIN, LOW);
33    delay(100);
34    digitalWrite(LED_BUILTIN, HIGH);  //short blink
35    delay(300);
36    digitalWrite(LED_BUILTIN, LOW);
37    delay(100);
38    digitalWrite(LED_BUILTIN, HIGH);  //short blink
39    delay(300);
40    digitalWrite(LED_BUILTIN, LOW);
41    delay(100);
42    digitalWrite(LED_BUILTIN, HIGH);  //short blink
43    delay(300);
44    digitalWrite(LED_BUILTIN, LOW);
45    delay(100);
46
47  }
48
```

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082



Program care simuleaza codul morse cu ajutorul led-ului implementat in placa arduino.

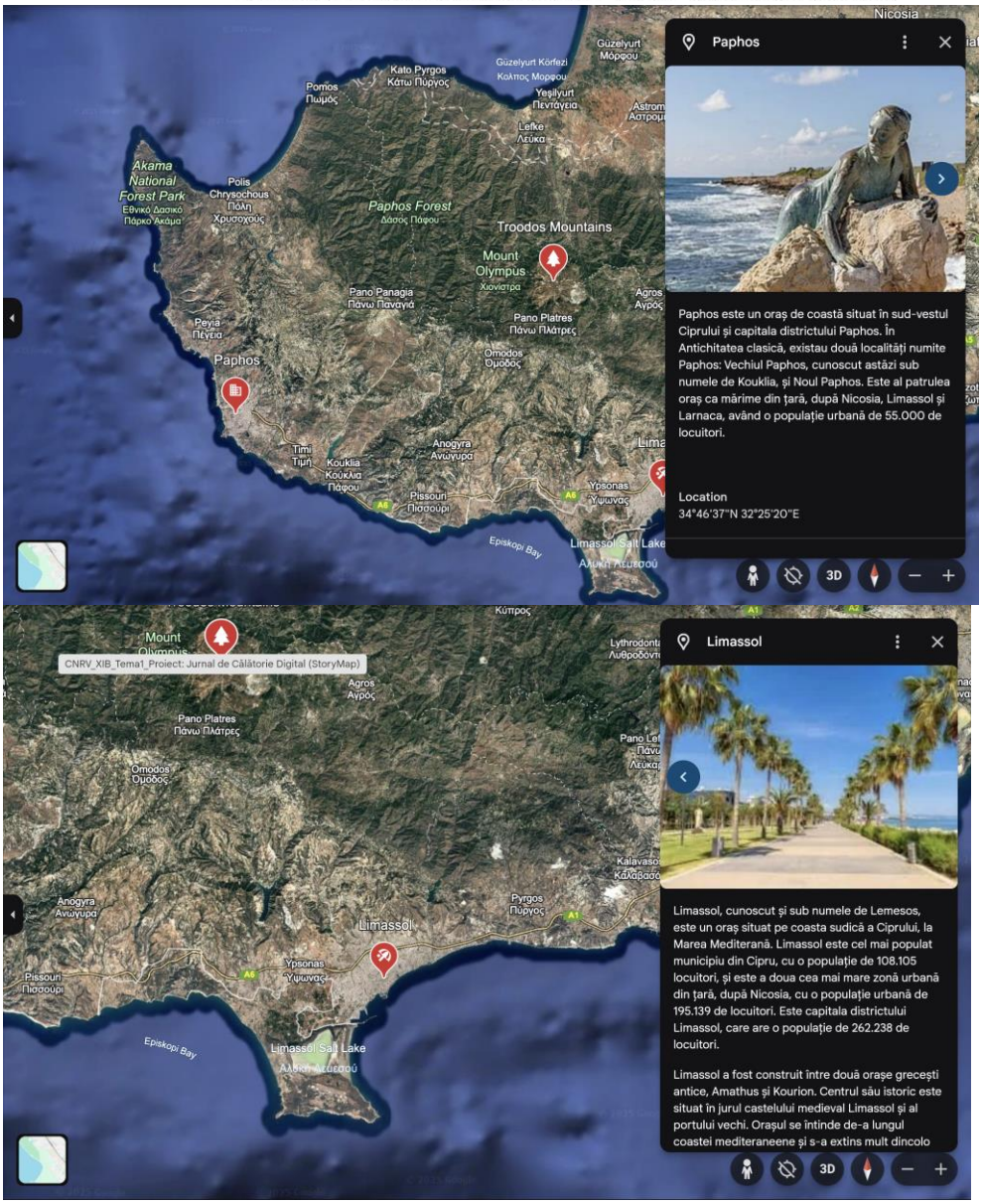
Jurnal de călătorie:

DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082

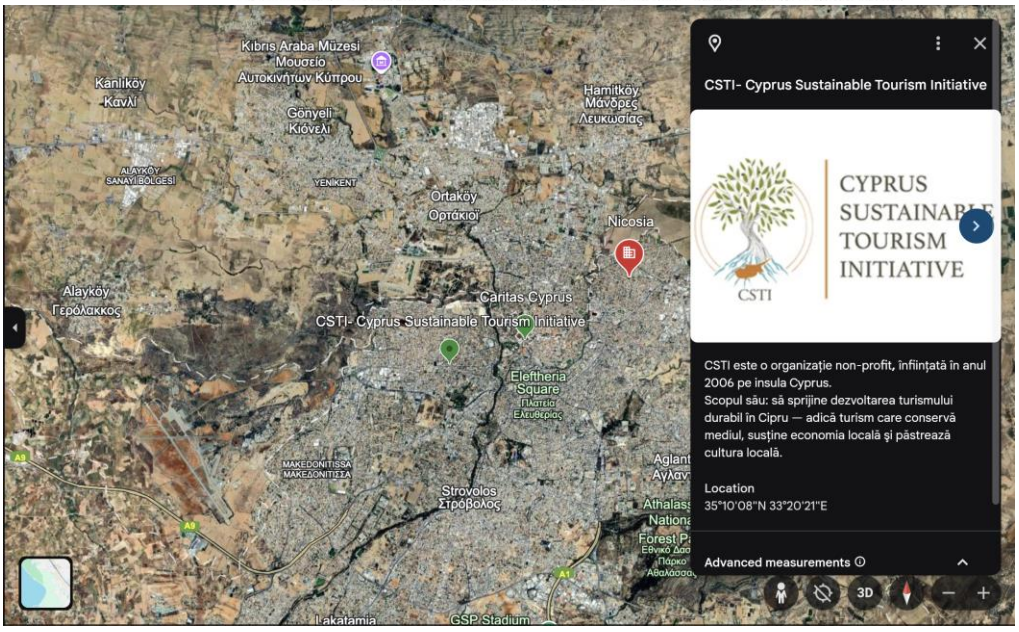


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Erasmus+

2025-1-RO01-KA121-VET-000319082





DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.