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, Roman

MOBILITY PORTFOLIO

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

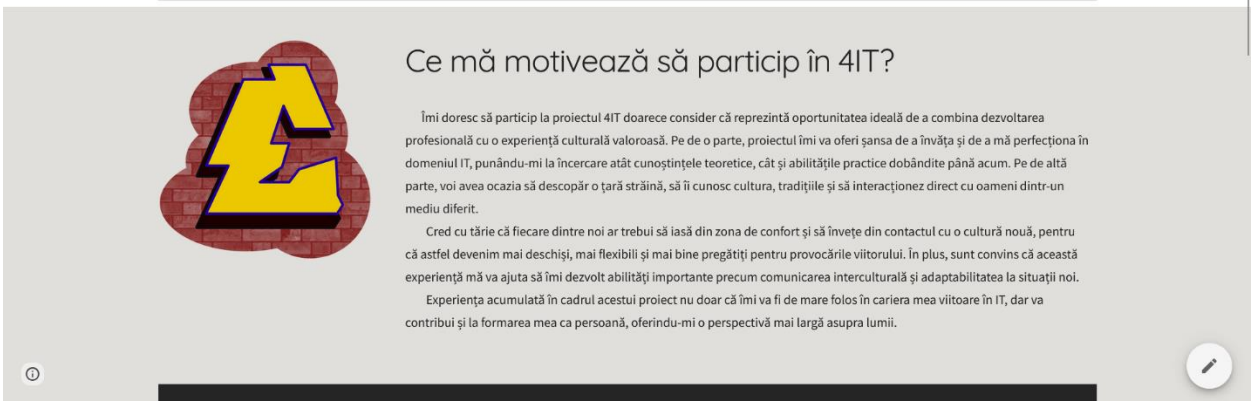
Participant: Balașcă Eduard-Gabriel

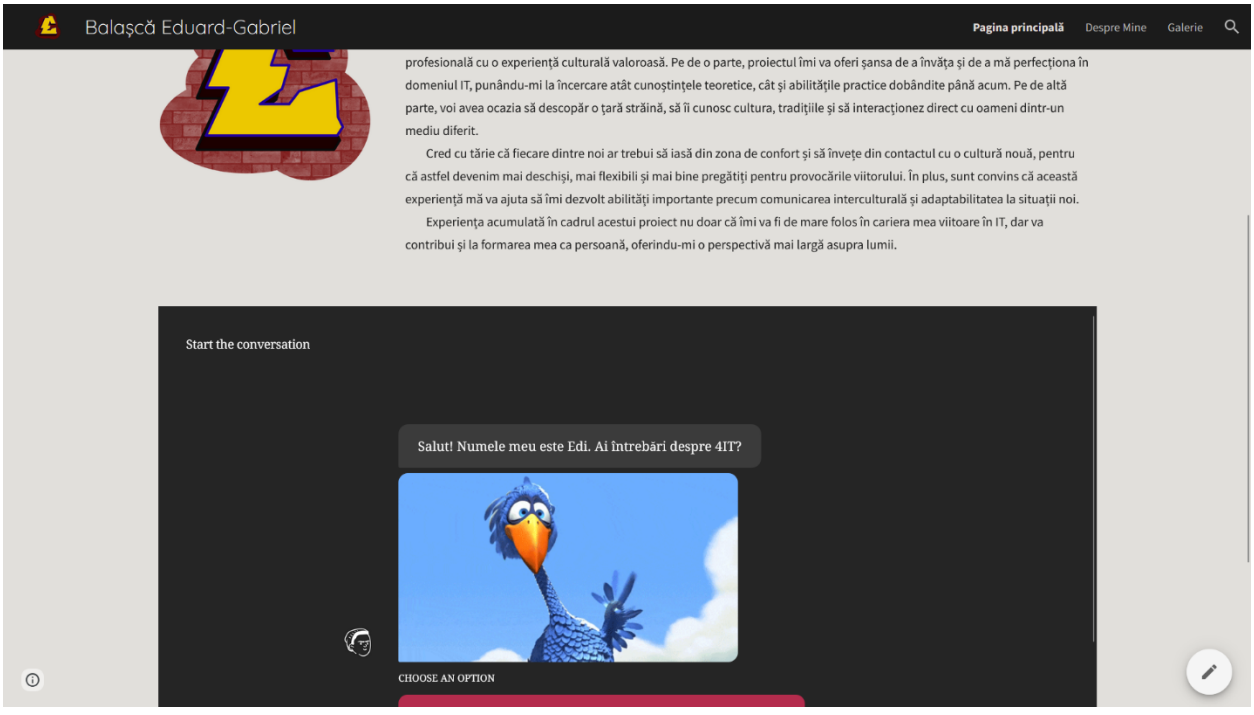
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2025-1-RO01-KA121-VET-000319082

1) Site



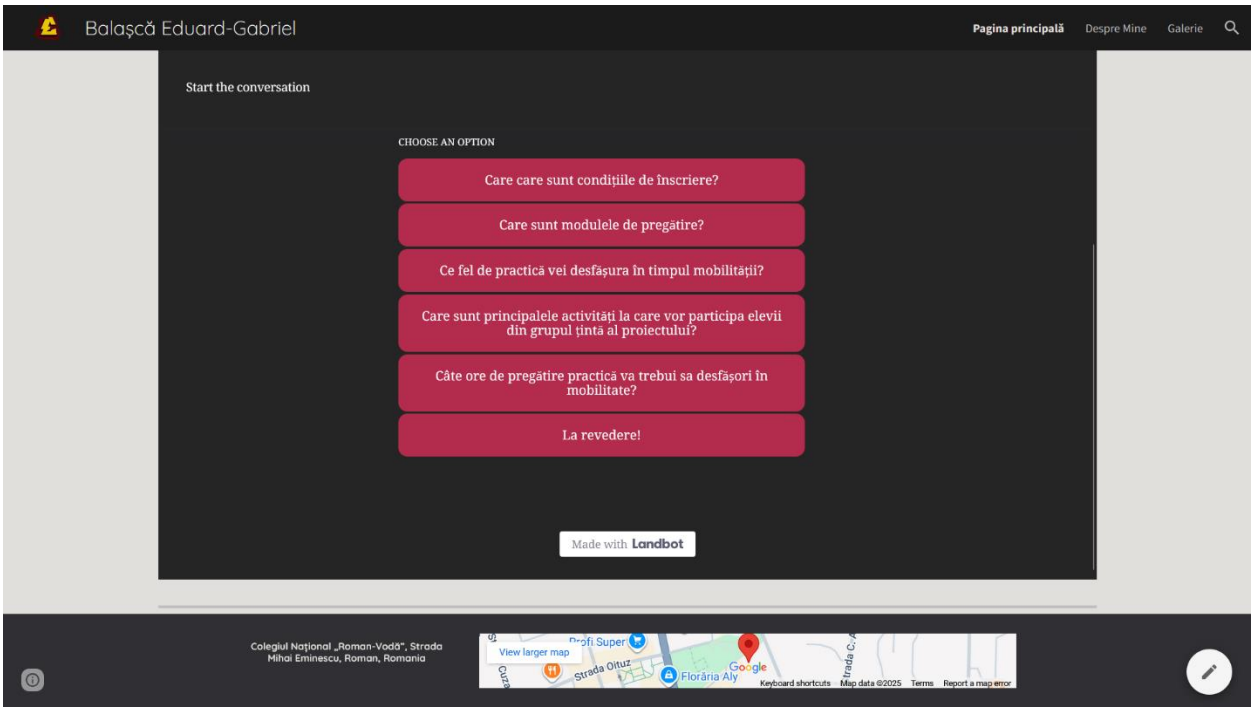


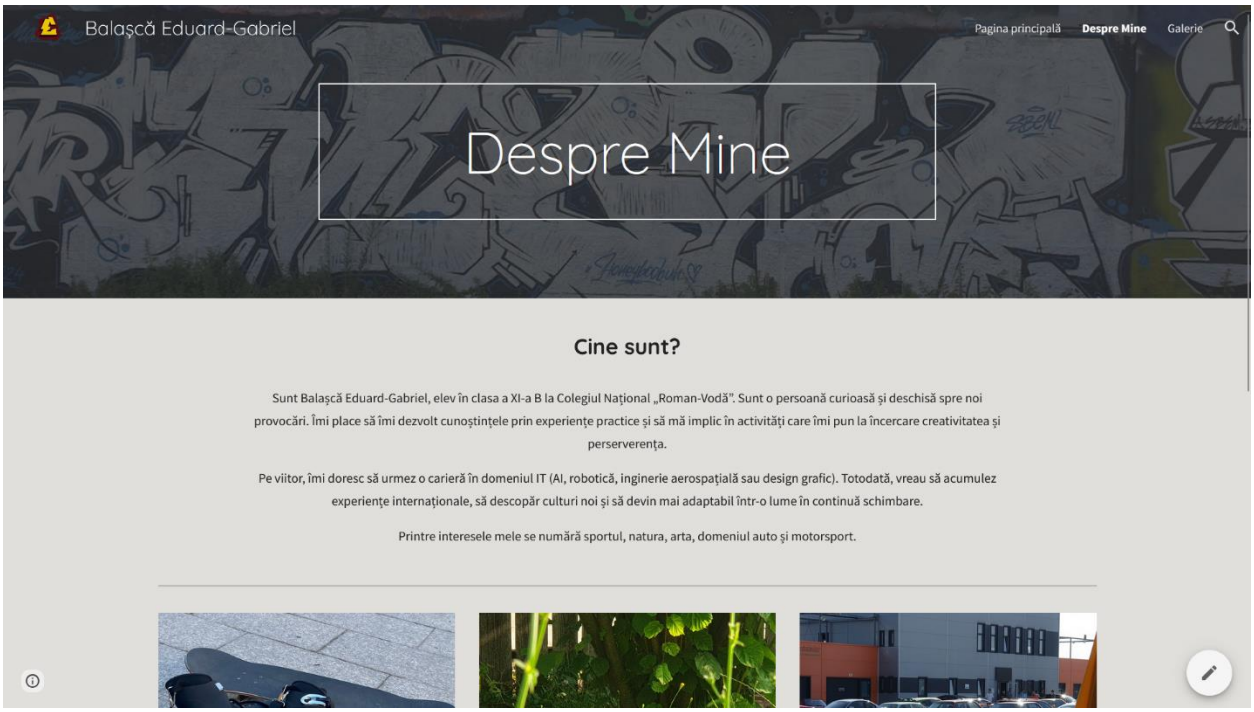


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

2025-1-RO01-KA121-VET-000319082



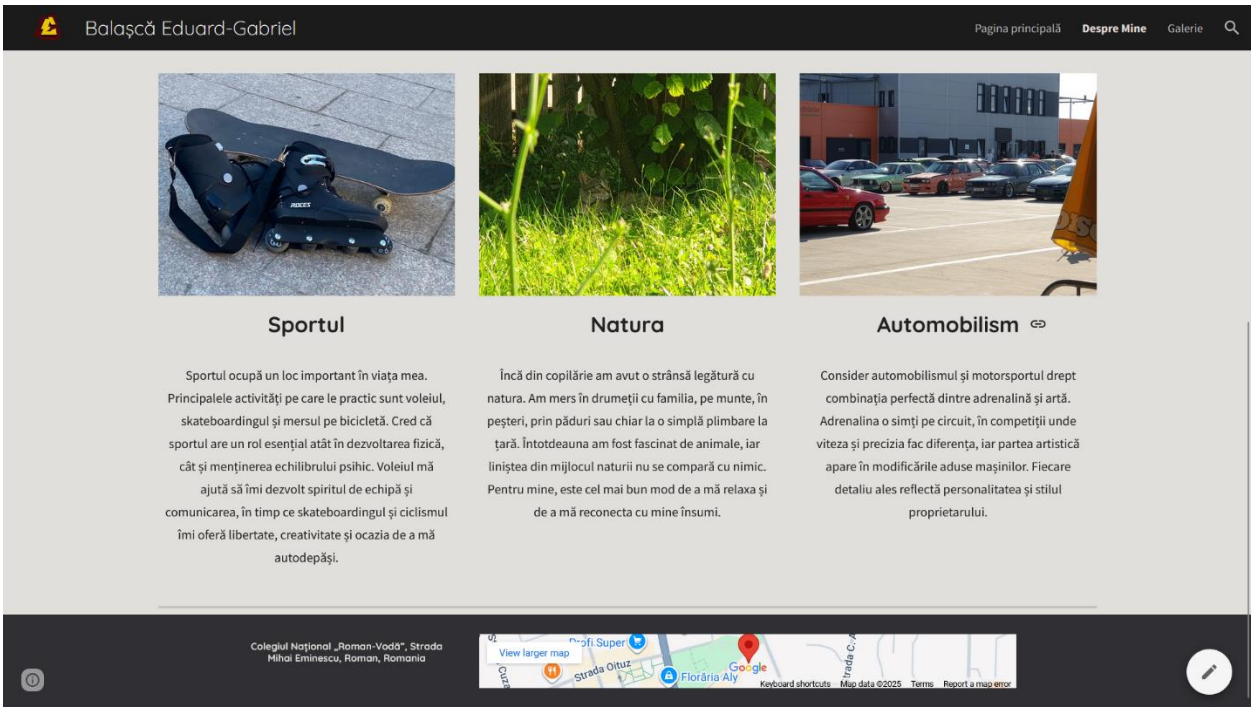


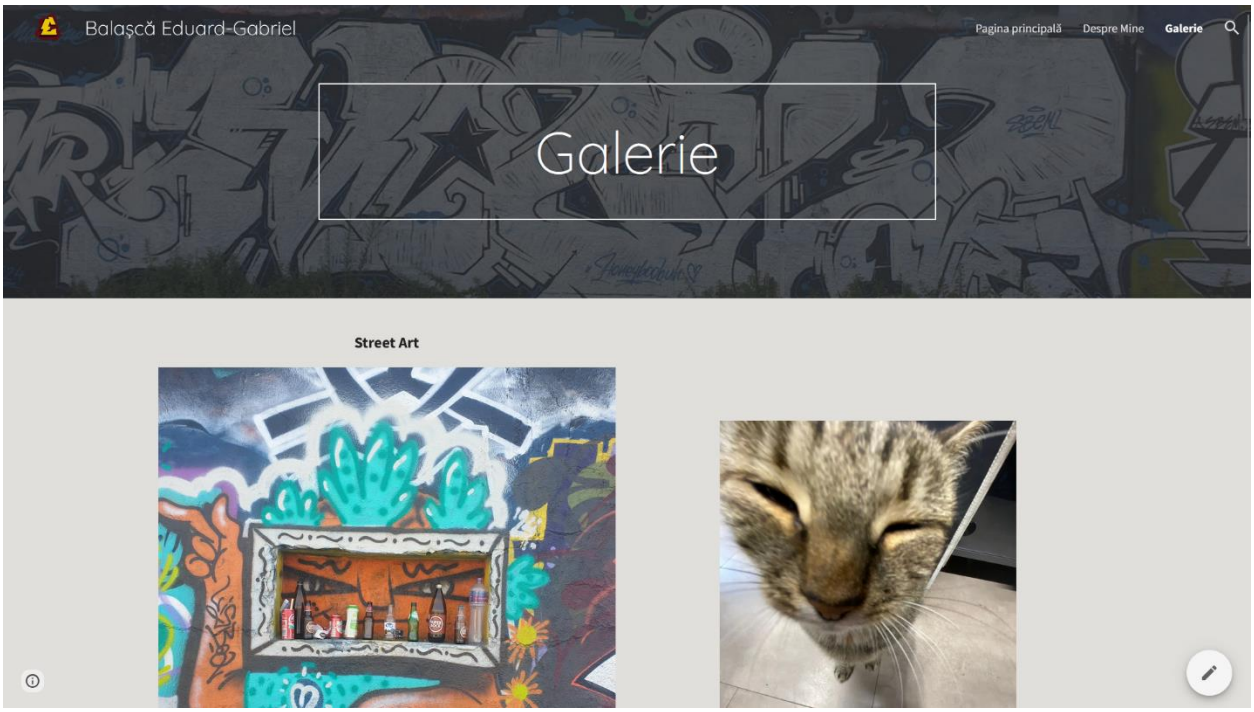
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

Erasmus+

2025-1-RO01-KA121-VET-000319082



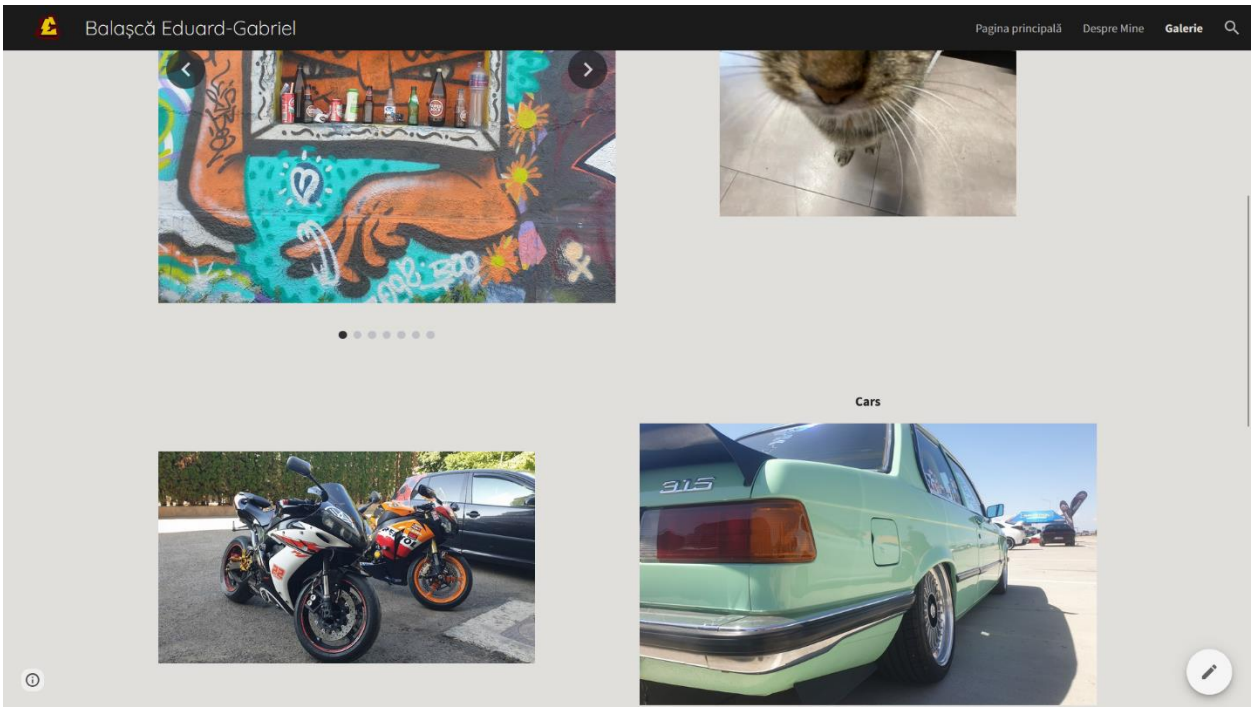


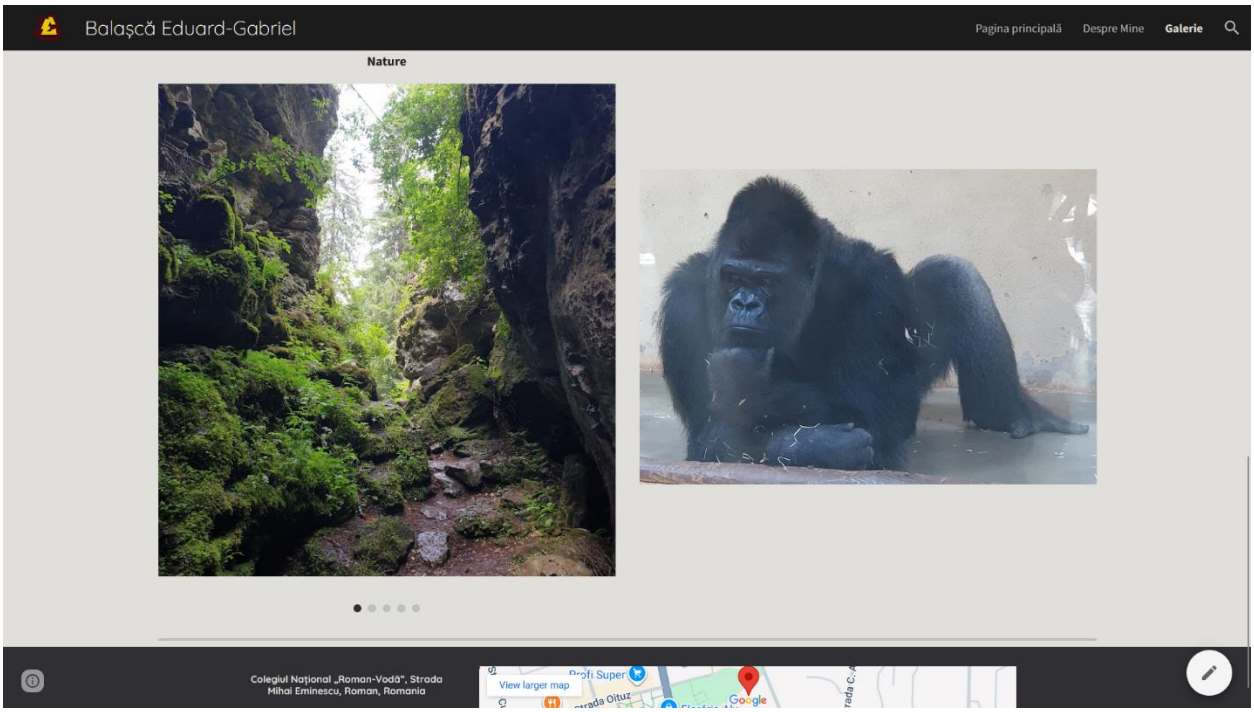
DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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.

2) 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.

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

2025-1-RO01-KA121-VET-000319082

Despre

- Localizare: Insulă în estul Mării Mediterane, aproape de Turcia și Siria
- Capitală: Nicosia
- Limbi oficiale: Greacă și turcă
- Monedă: Euro (€)
- Independență: 16 august 1960, față de Regatul Unit
- Guvernare: Republică democratică prezidențială
- A fost sub stăpânirea asirienilor, egiptenilor, persanilor, romanilor, bizantinilor, cruciaților, veneților și otomanilor
- În 1878 a trecut sub administrație britanică
- După lupte între ciprioții greci și turci, în 1960 devine independentă
- În 1974, insula este divizată: partea de sud (Republica Cipru) și partea de nord (ocupată de Turcia)
- Din 2004, Republica Cipru este membră a Uniunii Europene

1960

Stema țării

https://commons.wikimedia.org/wiki/File:Cyprus_location_map.svg



Stăpânirea veneţiană

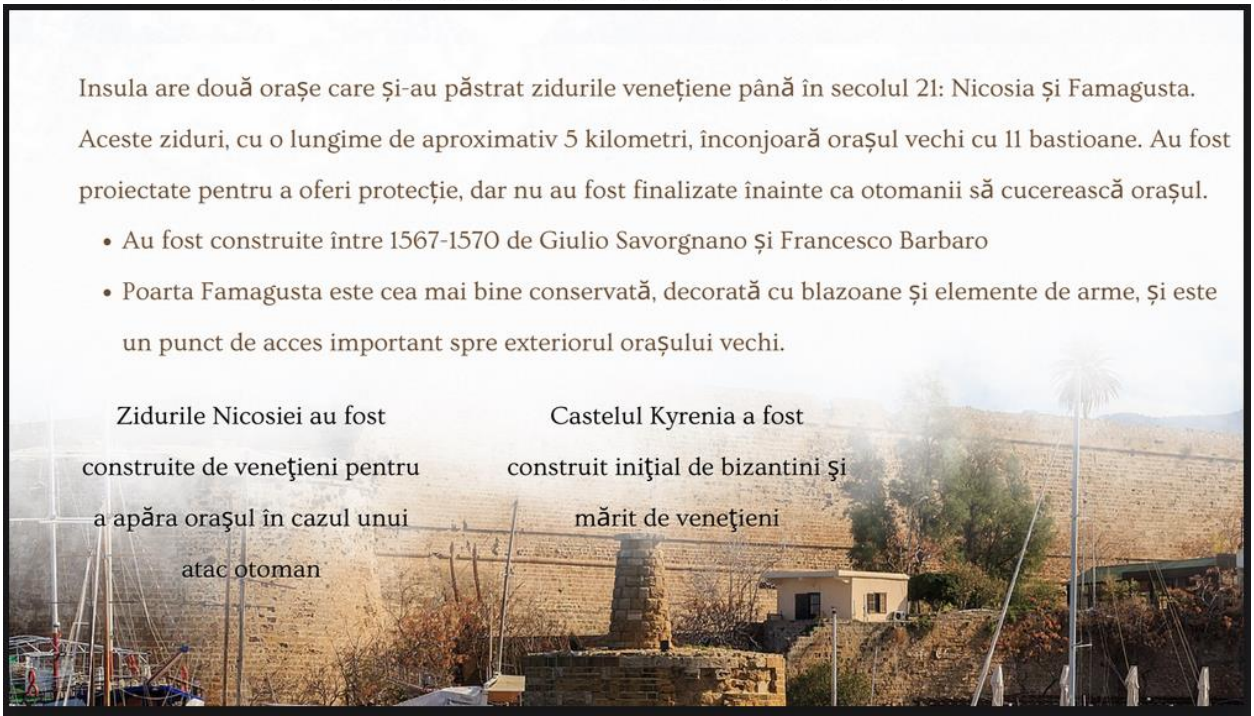
Stăpânirea veneţiană asupra Ciprului a durat din 1489 până în 1571, perioadă cunoscută sub numele de Venetokratia. După moartea lui Iacob al II-lea în 1473, ultimul rege din dinastia Lusignan, Republica Veneţia a preluat controlul insulei, în timp ce văduva veneţiană a regelui, regina Caterina Cornaro, a domnit ca o figură simbolică. Veneţia a anexat formal Regatul Ciprului în 1489, după abdicarea Caterinei.

Veneţienii au fortificat Nicosia construind zidurile oraşului şi au folosit-o ca un important centru comercial. Pe parcursul stăpânirii veneţiene, Imperiul Otoman a efectuat frecvente raiduri asupra Ciprului. În 1539, otomanii au distrus Limassol, iar de teama celor mai rele, veneţienii au fortificat şi Famagusta şi Kyrenia.

https://mycypruslife.com/ro/famagusta-vazuta-de-pe-ziduri/

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





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





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



Date biografice:

- Nume real: Mihail Christodoulou Mouskos
- Născut la 13 august 1913 în satul Panagia, districtul Paphos, Cipru.
- A studiat teologia la Atena și la Universitatea din Boston, SUA.
- În 1948 a devenit Arhiepiscop al Ciprului, sub numele de Makarios al III-lea.

Rol istoric:

- A fost liderul mișcării pentru independența Ciprului față de Marea Britanie.
- În anii '50, a sprijinit mișcarea EOKA, care lupta pentru eliberarea insulei.
- După negocierile cu Marea Britanie, Grecia și Turcia, Cipru a devenit stat independent în 1960.
- Makarios a devenit primul președinte al Republicii Cipru (1960–1977).

https://upload.wikimedia.org/wikipedia/commons/5/5e/Makarios_III_and_Robert_F_Wagner_NYWTS_cropped.jpg



Evenimente importante:

- A încercat să mențină pacea între ciprioții greci și turci.
- În 1974, a avut loc o lovitură de stat susținută de Grecia, urmată de invazia turcă din nordul insulei.
- Makarios a fost forțat să plece temporar, dar s-a întors ulterior ca președinte.

Moștenire:

- Considerat părintele fondator al statului cipriot.
- A fost o figură respectată pentru eforturile sale de a păstra unitatea națională.
- Bustul și statuia sa se află la Mănăstirea Kykkos, locul unde a fost călugăr în tinerețe.

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





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

1. Enciclopedia Britannica – https://www.britannica.com/place/Cyprus
2. Visit Cyprus – https://www.visitcyprus.com/
3. Wikipedia – https://ro.wikipedia.org/wiki/Cipru
4. UNESCO World Heritage – https://whc.unesco.org/
5. Wikimedia - https://commons.wikimedia.org/wiki/File:Cyprus_location_map.svg



Bibliografie

2025-1-RO01-KA121-VET-000319082

Mulțumesc

proiect realizat de Balașcă Eduard-Gabriel

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.

3) OOP (Object Oriented Programming)

```cpp
#include <iostream>
#include <climits>
using namespace std;
int main ()
{
    float *exam1, *exam2, *exam3, *avgrade;
    int maxexam=3;
    exam1= new int[maxexam];
    exam2= new int[maxexam];
    exam3= new int[maxexam];
    avgrade= new int[maxexam];
    int maxiex1, maxiex2, maxiex3, miniex1, miniex2, miniex3, maxi, mini;
    maxiex1=maxiex2=maxiex3-maxi=INT_MIN; miniex1=miniex2=miniex3=mini=INT_MAX;
    for(int i=0; i<maxexam; i++)
    {
        cout << "Enter grade for exam 1, student " << (i+1) << ": ";
        cin >> exam1[i];
        if(exam1[i]>maxiex1) maxiex1=exam1[i];
        if(exam1[i]<miniex1) miniex1=exam1[i];
        cout << "Enter grade for exam 2, student " << (i+1) << ": ";
        cin >> exam2[i];
        if(exam2[i]>maxiex2) maxiex2=exam2[i];
        if(exam2[i]<miniex2) miniex2=exam2[i];
        cout << "Enter grade for exam 3, student " << (i+1) << ": ";
```

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.

```
        cout << "Enter grade for exam 3, student " << (i+1) << ": ";
        cin >> exam3[i];
        if(exam3[i]>maxiex3) maxiex3=exam3[i];
        if(exam3[i]<miniex3) miniex3=exam3[i];
        avgrade[i]=(exam1[i]+exam2[i]+exam3[i])/3;
    }

    if(maxiex1>maxiex2) maxi=maxiex1;
    else maxi=maxiex2;
    if(maxiex3>maxi) maxi=maxiex3;
    if(miniex1<miniex2) mini=miniex1;
    else mini=miniex2;
    if(miniex3<mini) mini=miniex3;
    for(int i=0; i<maxexam; i++)
    {

    }
    cout<<maxiex1<<' '<<miniex1;
    cout<<maxiex2<<' '<<miniex2;
    cout<<maxiex3<<' '<<miniex3;
    return 0;
}
```

```
int main()
{
    int ARRAYSIZE;
    int *exam1, *exam2;
    int *temp1, *temp2;
    int  n = 0, tmpN=0;
    cin >> ARRAYSIZE;
    exam1 = new int[ARRAYSIZE];
    exam2 = new int[ARRAYSIZE];
    cout<<"How many numbers would you like to enter?:";
    do{
        cout<<"Enter number "<<n+1<<" ->";
        cin >> examen1[n]>> examen2[n];
        cout<<"Number saved!"<<endl;
        n++;
        if (n >= ARRAYSIZE)
        {
            cout<<"Out of Space!!!! would you like to enter more? Please enter how many(0 to Exit): ";
            if (tmpN > 0)
            {
                ARRAYSIZE = ARRAYSIZE + tmpN;
                temp1 = new int[ARRAYSIZE];
                temp2 = new int[ARRAYSIZE];
                for (int i=0; i<n; i++)
```

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

```
                    temp2 = new int[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<<"Now printing the numbers you inputed"<<endl;
        for (int i=0; i<ARRAYSIZE;i++){

            cout<<exam1[i]<<endl<<exam2<<endl<<((float)exam1[i]+exam2[i])/2<<endl<<endl;
        }
        delete [] exam1;
        delete [] exam2;
```

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

/*
EXERCISE
write a program which takes input from the user
the information for cars.
you need to save the following information:
BRAND
MODEL
YEAR
HP
When you are done reading print your information on the screen. Create a growing array. Print all of the information inputted.
*/

class Cars{
private:
    string model;
    string brand;
    string year;
    float hp;
public:
    void setModel(string m){
        model=m;}
    void setBrand(string b){
        brand=b;}
    void setYear(string y){
        year=y;}
    void setHp(float h){
        hp=h;}

    string getModel(){return model;}
    string getBrand(){return brand;}
    string getYear(){return year;}
    float getHp(){return 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

```cpp
int main()
{   string model[20], brand[20], year[20];
    float hp[20];
    int i;
    int* temp, nr;
    Cars test;

    temp = new int();

    cout<<"Welcome. How many cars would you like to save? ";
    cin>>nr;


    for (i=1;i<=nr;i++){
    cout<<"Please input the name of the car model:";
    cin>>model[i];
    cout<<"Please input the the name of the car brand:";
    cin>>brand[i];
    cout<<"Please input the year of manufacturing:";
    cin>>year[i];
    cout<<"Please input the horsepower of the car:";
    cin>>hp[i];
    cout<<"Car saved!";
    test.setModel(model[i]);
    test.setBrand(brand[i]);
    test.setYear(year[i]);
    test.setHp(hp[i]);

        if (i==nr) {
            cout<<"================================";
            cout<<"You have run out of space. How many cars would you like to add? (0 to exit) ";
            cin>>*temp;
        }
    }

    nr = nr + *temp;
    nr = nr + *temp;
    cout<<"================================";
    for (i=1;i<=nr; i++)
    {cout<<"The car you inputted is "<<test.getBrand()<<" "<<test.getModel()<<" manufactured in "<<test.getYear()<<". It has "<<test.getHp()<<" hors

    delete [] temp;
    return 0;
}
```

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>
#include <string>
using namespace std;
/*Homework write a program which is going to use a class to create objects in order to save information about cars.
the information you save will be the same as the first exercise above
Brand
Model
Year
HP
in your program you must create an array of objects that is growing dynamically (like exercise Growing Arrays).
You will ask the user how many cars they want to save and when the array is full you will dynamically grow the array.
Your program at the end must print the information of every car that was saved from the user.*/
class Car {
    public:
        string brand;
        string model;
        int year;
        int hp;
};

int main()
{
    int n;
    cout << "How many cars would you like to enter?: ";
    cin >> n;

    Car cars[n];

    for (int i = 0; i < n; i++) {
        cout << "Enter details for car " << (i + 1) << ":\n";
        cout << "Brand: ";
        cin >> cars[i].brand;
        cout << "Model: ";
        cin >> cars[i].model;
        cout << "Year: ";
        cin >> cars[i].year;
        cout << "HP: ";

        cout << "\nCars informations:\n";
        for (int i = 0; i < n; i++) {
            cout << "Car " << (i + 1) << ":\n";
            cout << "Brand: " << cars[i].brand << "\n";
            cout << "Model: " << cars[i].model << "\n";
            cout << "Year: " << cars[i].year << "\n";
            cout << "HP: " << cars[i].hp << "\n\n";
        }

        return 0;
    }
```

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

pacienti.cpp

```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;
```

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

```
45          cout << "What would you like to do? ";
46          cin >> userChoice;
47          cout << "------------------------------" << endl;
48
49          if (userChoice == 1) {
50              cout << "Add a new Patient" << endl << "-----------------" << endl;
51              if (index >= ArraySize) {
52                  ArraySize = ArraySize + 1;
53                  clinic = growArray(clinic, ArraySize, index);
54              }
55              addPatient(clinic, index);
56              index++;
57          }
58          else if (userChoice == 2) {
59              cout << "Edit a Patient (not implemented yet)" << endl;
60          }
61          else if (userChoice == 3) {
62              cout << "Delete a Patient (not implemented yet)" << endl;
63          }
64          else if (userChoice == 4) {
65              cout << "Search By Name" << endl << "-----------------" << endl;
66              string name;
67              cout << "Enter name to search: ";
68              cin >> name;
69              searchByName(clinic, index, name);
70          }
71          else if (userChoice == 5) {
72              cout << "Search By Medicine" << endl << "------------------" << endl;
73              string med;
74              cout << "Enter medicine to search: ";
75              cin >> med;
76              searchByMedicine(clinic, index, med);
77          }
78          else if (userChoice == 6) {
79              printArray(clinic, index);
80          }
81          else if (userChoice == 0) {
```

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 << "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;
}

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);
```

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.

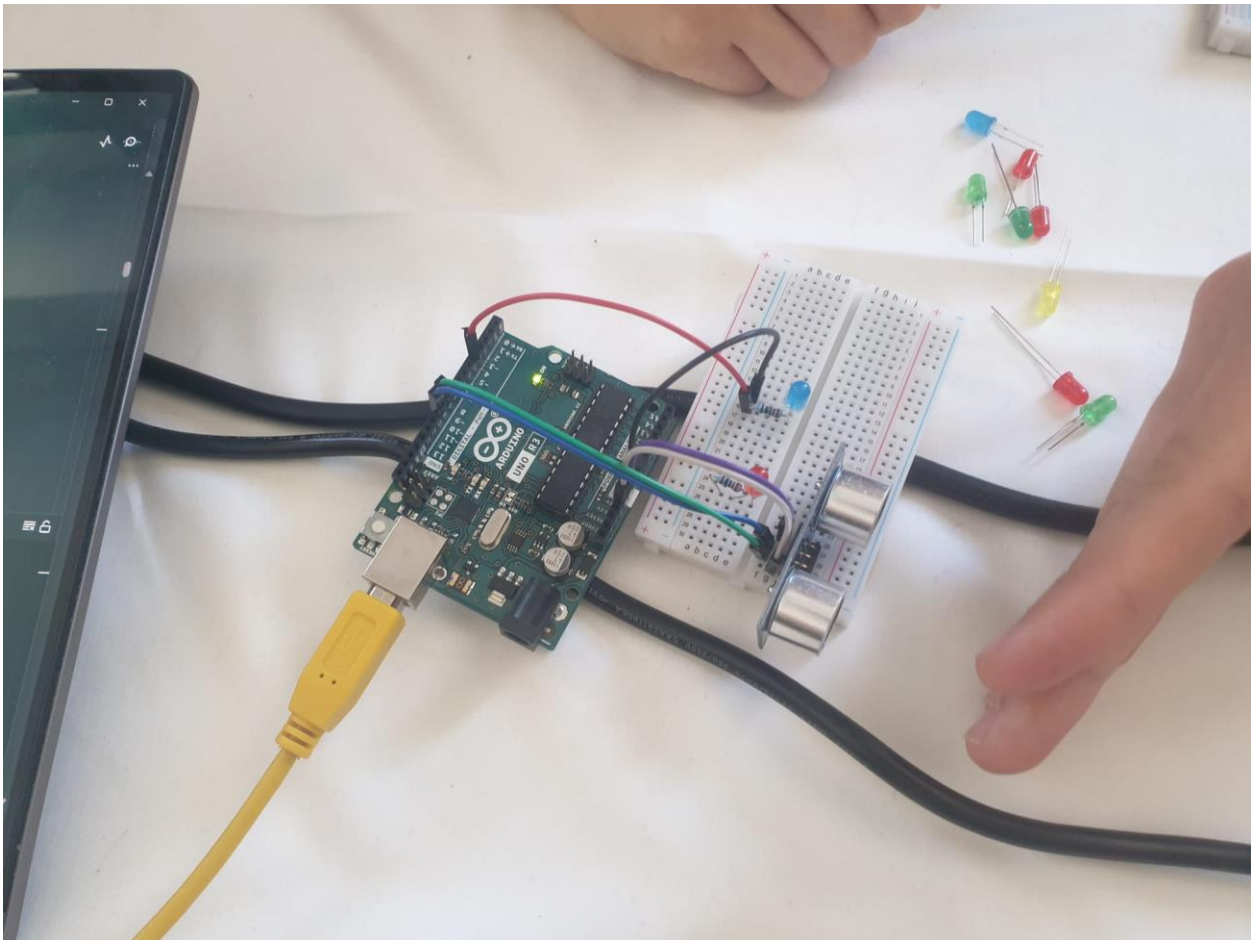
Co-funded by the Erasmus+ Programme of the European Union
Erasmus+
2025-1-RO01-KA121-VET-000319082

```
122     cout << "Patient Successfully Added!" << endl;
123 }
124
125 void printArray(patient *clinic, int index) {
126     cout << "PRINTING ALL DATA" << endl << "------------------" << endl;
127     for (int i = 0; i < index; i++) {
128         cout << "----- Patient " << i + 1 << " -----" << endl;
129         cout << "Name: " << clinic[i].getName() << endl;
130         cout << "Age: " << clinic[i].getAge() << endl;
131         cout << "Phone: " << clinic[i].getPhone() << endl;
132         cout << "Medicine: " << clinic[i].getMedicine() << endl;
133     }
134 }
135
136 /*-----------------------------------
137     SEARCH FUNCTIONS
138 -----------------------------------*/
139
140 void searchByName(patient *clinic, int index, string name) {
141     bool found = false;
142     for (int i = 0; i < index; i++) {
142     for (int i = 0; i < index; i++) {
143         if (clinic[i].getName() == name) {
144             cout << "Patient Found!" << endl;
145             cout << "Name: " << clinic[i].getName() << endl;
146             cout << "Age: " << clinic[i].getAge() << endl;
147             cout << "Phone: " << clinic[i].getPhone() << endl;
148             cout << "Medicine: " << clinic[i].getMedicine() << endl;
149             found = true;
150         }
151     }
152     if (!found)
153         cout << "No patient found with the name \"" << name << "\"." << endl;
154 }
155
156 void searchByMedicine(patient *clinic, int index, string medicine) {
157     bool found = false;
158     for (int i = 0; i < index; i++) {
159         if (clinic[i].getMedicine() == medicine) {
160             cout << "Patient Found!" << endl;
161             cout << "Name: " << clinic[i].getName() << endl;
162             cout << "Age: " << clinic[i].getAge() << endl;
163             cout << "Phone: " << clinic[i].getPhone() << 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.

```
            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;
}
```

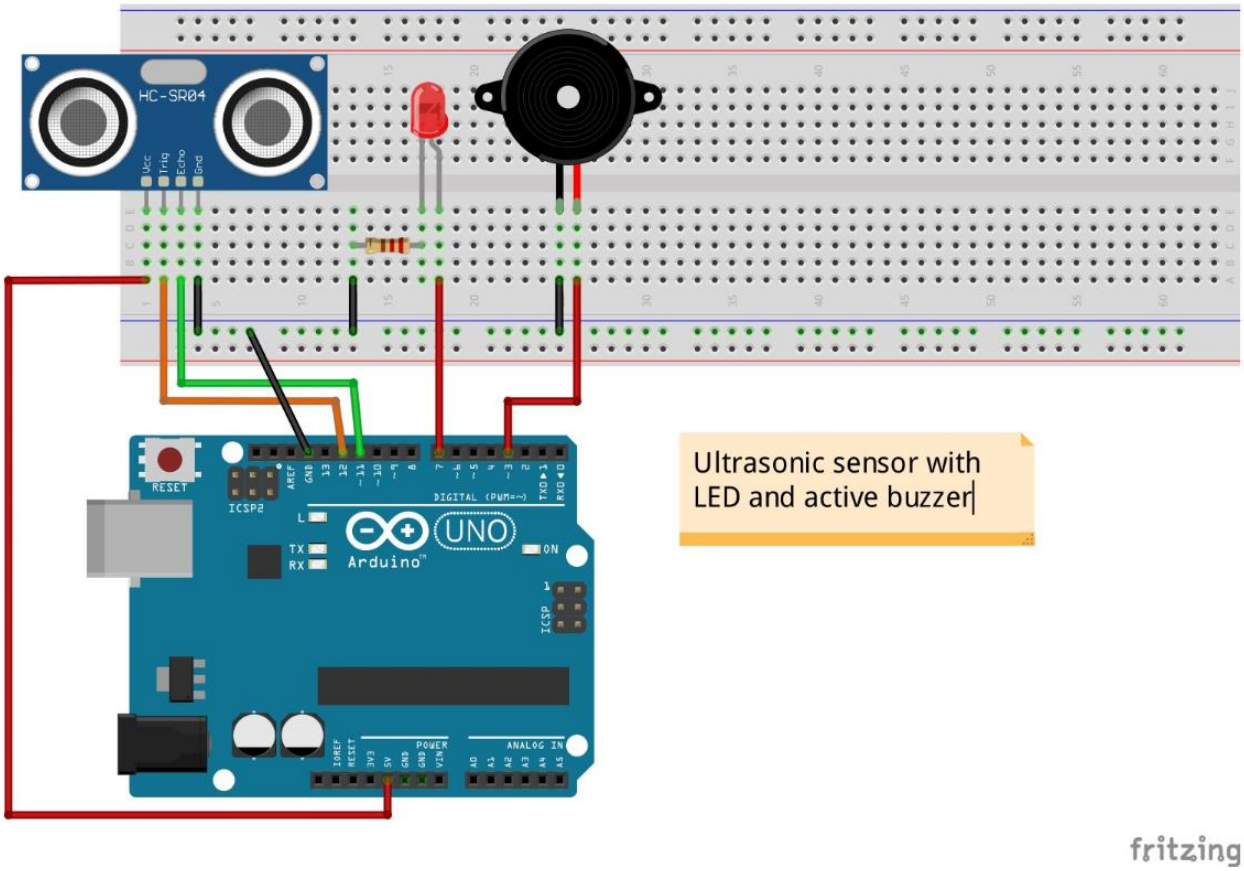
4) Arduino



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



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

```ino
//include library by using Sketch->Include Library->Add .Zip library-> find "SR04.zip"
#include "SR04.h"

// assign pins for ultrasonic sensor HC-SR04
// trigger pin to #12; echo pin to #11
#define TRIG_PIN 12
#define ECHO_PIN 11

// we create an instance of the "SR04" object type (this is from the library file we include)
// the argument for creating the instance is the trigger pin and echo pin we specified.
SR04 sr04 = SR04(ECHO_PIN, TRIG_PIN);

// variable a is to store measured distance from ultrasonic sensor
long a;

// connect led pin to #7
int ledpin = 7;
// connect active buzzer pin to #3
int buzzerpin = 3;

//set a distance threshold value
long WarningDistance = 10; // threshold value is 10cm. if an object distance<WarningDistance, we let LED and buzzer on

void setup() {

  //initialize serial monitor
  Serial.begin(9600);
  //set ledpin and buzzerpin as OUTPUT (we send commands to them)
  pinMode(ledpin, OUTPUT);
  pinMode(buzzerpin, OUTPUT);

  delay(1000);
}

void loop() {
  //
  a = sr04.Distance();
  Serial.print(a);
  Serial.println("cm");

  // LED and buzzer go on when measured distance < our threshold value WarningDistance
  if (a <= WarningDistance) {
    digitalWrite(ledpin, HIGH);
    digitalWrite(buzzerpin, HIGH);
  }
  // if distance>threshold, we turn off LED and Buzzer
  else {
    digitalWrite(ledpin, LOW);
    digitalWrite(buzzerpin, LOW);
  }
  delay(300);
}

```

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
// Definire pini
const int trigPin = 9;
const int echoPin = 10;
const int ledPin = 11;
const int buzzerPin = 12;

// Variabile pentru măsurarea distanței
long duration;
int distance;

// Definire note muzicale
int melody[] = {
  262, 294, 330, 349, 392, 440, 494, 523 // Do, Re, Mi, Fa, Sol, La, Si, Do (octava 4)
};

// Definire durata notelor
int noteDurations[] = {
  400, 400, 400, 400, 400, 400, 400, 800 // Durata fiecărei note (în milisecunde)
};

void setup() {
  // Setează pinii
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  Serial.begin(9600);
}

void loop() {
  // Trimite un impuls pe pinul Trig
```

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

```
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Citește durata semnalului de la pinul Echo
  duration = pulseIn(echoPin, HIGH);

  // Calculează distanța
  distance = duration * 0.034 / 2; // Formula pentru distanță (în cm)

  // Afișează distanța pe Serial Monitor
  Serial.print("Distanta: ");
  Serial.print(distance);
  Serial.println(" cm");

  // Dacă distanța este sub 10 cm, activează alarma
  if (distance < 10) {
    digitalWrite(ledPin, HIGH); // Aprinde LED-ul

    // Redă melodia
    for (int i = 0; i < 8; i++) {
      tone(buzzerPin, melody[i]);          // Redă fiecare notă
      delay(noteDurations[i]);             // Așteaptă durata notei
      noTone(buzzerPin);                   // Oprește sunetul
      delay(50);                           // Pauză între note
    }
  } else {
    digitalWrite(ledPin, LOW);  // Stinge LED-ul
    noTone(buzzerPin);          // Oprește sunetul buzzer-ului
  }

    delay(100); // Pauză între măsurători
  }
```

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.