2025-1-RO01-KA121-VET-000319082

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

Beneficiary: "Gheorghe Vranceanu" National College, Bacau

Sending organization: "Ferdinand I" National College, Bacau

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: Ilie Mihnea-Lucas

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

Introducere personală (site)

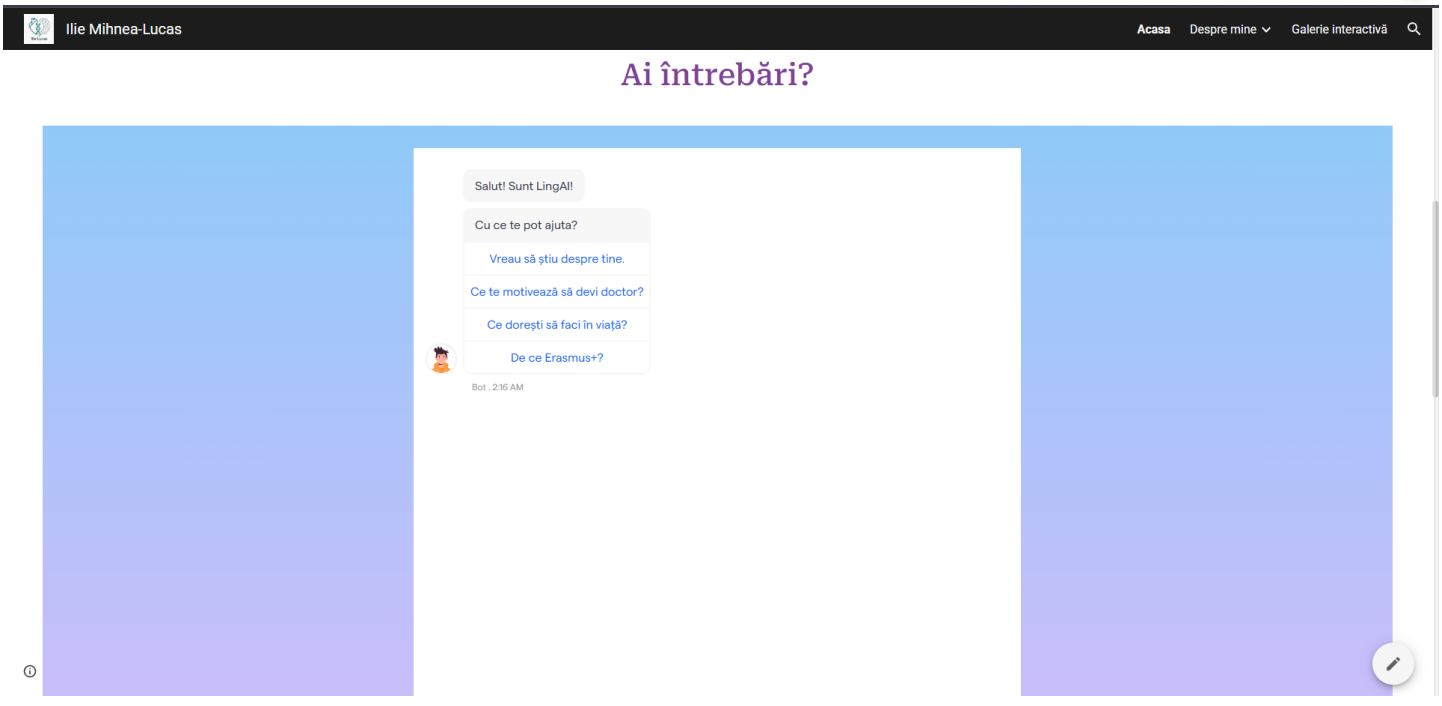


De ce îmi doresc să particip?

Participarea la acest proiect Erasmus+ îmi va da oportunitatea de a mă dezvolta profesional și personal într-un cadru global ce încurajează empatia, munca în echipă și schimbul de idei.

Pe lângă explorarea diferitelor culturi și exprimarea unei dorințe de a avea un impact pozitiv, sper să îmi îmbunătățesc abilitățile practice si profesionale in cadrul realizării de site-uri web si comerț electronic. Vreau să imi transform cunoștințele teoretice în rezultate precise.





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.

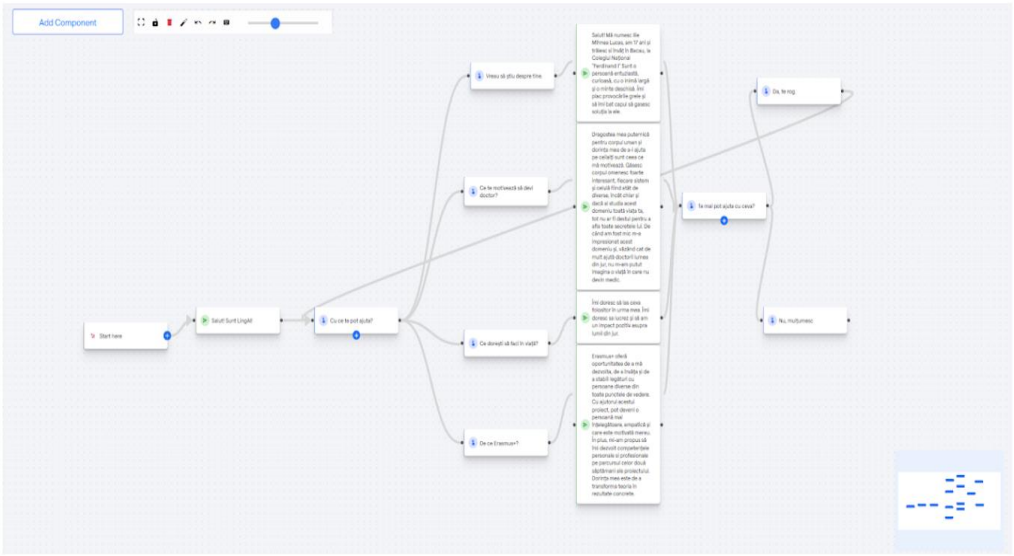
Co-funded by the Erasmus+ Programme of the European Union

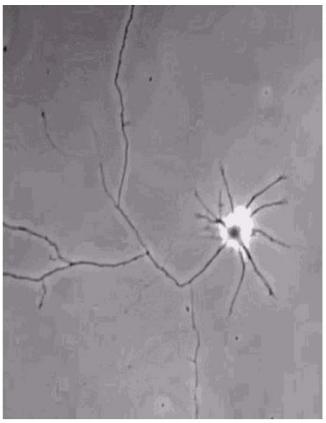
Erasmus+

2025-1-RO01-KA121-VET-000319082

Chatbot

Daca dintr-un motiv nu funcționează chatbot-ul, apăsați pe butonul din stânga.
Vă va trimite la o pagină noua dedicată acestuia.





Ilie Mihnea-Lucas

Acasa Despre mine Galerie interactivă

Aspirațiile mele



Chemarea mea către medicină

Dorința mea cea mai mare este să devin medic, întrucât consider că acest lucru îmi va permite să combin dragostea mea pentru științe cu dorința de a ajuta pe ceilalți și de a lăsa ceva în urma mea. Am fost întotdeauna fascinat de modul în care funcționează corpul uman, de complexitatea fiecărei celule și de metoda perfectă în care totul este sincronizat. Am învățat că studiul biologiei și al chimiei înseamnă mai mult decât simple materii academice: este vorba despre un set de cunoștințe vaste care poate fi înțeles și utilizat pentru a le reda sănătatea și speranța celor din jur.

Pentru mine, medicina înseamnă mai mult să fii aproape de oameni decât să te ocupi de proceduri și tratamente. Pe lângă furnizarea de tratamente medicale, vreau să fiu persoana la care pacienții pot apela pentru curaj și încredere în momentele cele mai dificile. A fi medic implică sacrificiu, angajament și responsabilitate, dar aduce și un mare sentiment de împlinire, deoarece știi că ai îmbunătățit viața oamenilor prin profesia ta.

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.

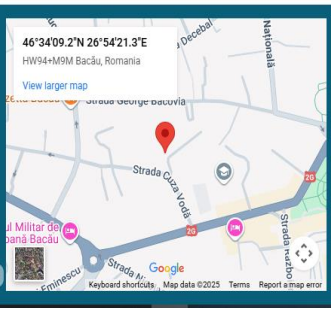
Co-funded by the Erasmus+ Programme of the European Union

2025-1-RO01-KA121-VET-000319082

Dorința de a împărtăși

Mereu mi-am dorit să las ceva în urmă pe lume, să fac o diferență. Această convingere mi-a fost îndemnată de mama, care mereu mi-a spus că cei mai importanți oameni de pe lume sunt fiecare dintre noi are un dar de oferit lumii. Ea este cea care m-a învățat că una dintre cele mai frumoase modalități de a face asta este să împărtășim cunoștințele pe care le dobândim. De aceea, voi depune tot efortul pentru a lăsa acel ceva, pe lume.





Strada George Bacovia nr. 45.
Colegiul Național "Ferdinand I"

Pentru informații, vă rog să contactați:
Telefon: +40-736-192-526
Instagram: @mihneaalucas
lucas_ilie1@icloud.com

Ilie Mihnea-Lucas

Acasa | Despre mine | Galerie interactivă

Pasiunile mele



Biologia și chimia

Biologia este harta corpului uman. Ea mă învață despre "arhitectura" organelor, despre modul de funcționare al celulelor și despre complexitatea sistemelor care ne mențin în viață. Fără această fundație, anatomia, fiziologia și chiar boala ar fi doar o provocare nerezolvabilă. Biologia mă pregătește să înțeleg „de ce" și „cum" funcționează organismul sănătos, pentru ca mai apoi să pot identifica cu precizie ce anume s-a abătut de la această funcționare perfectă.

Chimia, la rândul ei, este limbajul acestor procese. Ea îmi explică schimburile energetice din celule și modul în care moleculele interacționează pentru a menține echilibrul. Ea este știința care stă la baza dezechilibrelor care stau la originea bolilor.

Aceste două domenii stau la baza pasiunilor mele.

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



Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Inginerie Chimică și Protecția Mediului "Cristofor Simionescu"

Societatea Profesorilor de Științe Iași

Concursul online de fizică, chimie și biologie „ZIRCON"

Se acordă

Premiul II

elevului/elevei **MIHNEA-LUCAS ILIE**

din **clasa a X-a** de la

COLEGIUL NATIONAL "FERDINAND I" BACAU

pentru rezultatul obținut la secțiunea **CHIMIE - BIOLOGIE**

la a cincea ediție a Concursului "ZIRCON" online, 15-16 martie 2025

Cadre didactice îndrumătoare:

prof. LUCIANA TUFESCU (CHIMIE) și prof. SIMONA MOCANU (BIOLOGIE)

Decan FICPMCS Prof. univ. dr. ing. Teodor Mihăilescu

Președinte SPSI Prof. dr. Elena-Diana Carasevici

Nr. ICPM/189/1345-CB/2025

INGENIUM European University

Iași 2025



INSPECTORATUL ȘCOLAR JUDEȚEAN BACĂU

MINISTERUL EDUCAȚIEI

DIPLOMĂ

Se acordă PREMIUL al II-lea,

elevului / ~~elevei~~ ILIE MIHNEA LUCAS, din clasa a IX-a,

de la unitatea de învățământ COLEGIUL NAȚIONAL „FERDINAND I" BACĂU

pregătit(ă) de doamna / ~~domnul~~ PROF. MOCANU SIMONA-IULIANA

pentru rezultatele obținute la etapa județeană a

OLIMPIADA DE BIOLOGIE

INSPECTOR ȘCOLAR GENERAL Prof. Egarmin Ana-Maria

INSPECTOR ȘCOLAR PROF. RUSU GEORGETA-IRINA

Nr. 3331/13 din 21.04.2024

Robotica

Am fost impresionat de energia și spiritul de echipă, de modul în care oamenii se strâng în jurul unui obiectiv tehnic complex și lucrează împreună pentru un rezultat concret. Fiecare departament: mecanică, programare, PR este esențial pentru "sănătatea" și succesul întregii echipe.

Aici am înțeles că un proiect tehnic, oricât de complex, trăiește prin povestea pe care o spune. Am avut șansa să fiu acel drum între lumea tehnică a robotului și public, părinți, sponsori și comunitate. Am învățat să "traduc" termeni complicați în mesaje clare și captivante, să motivez echipa și să construim împreună imaginea proiectului. Pentru mine, aceasta a fost o lecție practică de comunicare și empatie, abilități pe care le consider semnificative și în viitoarea mea carieră.

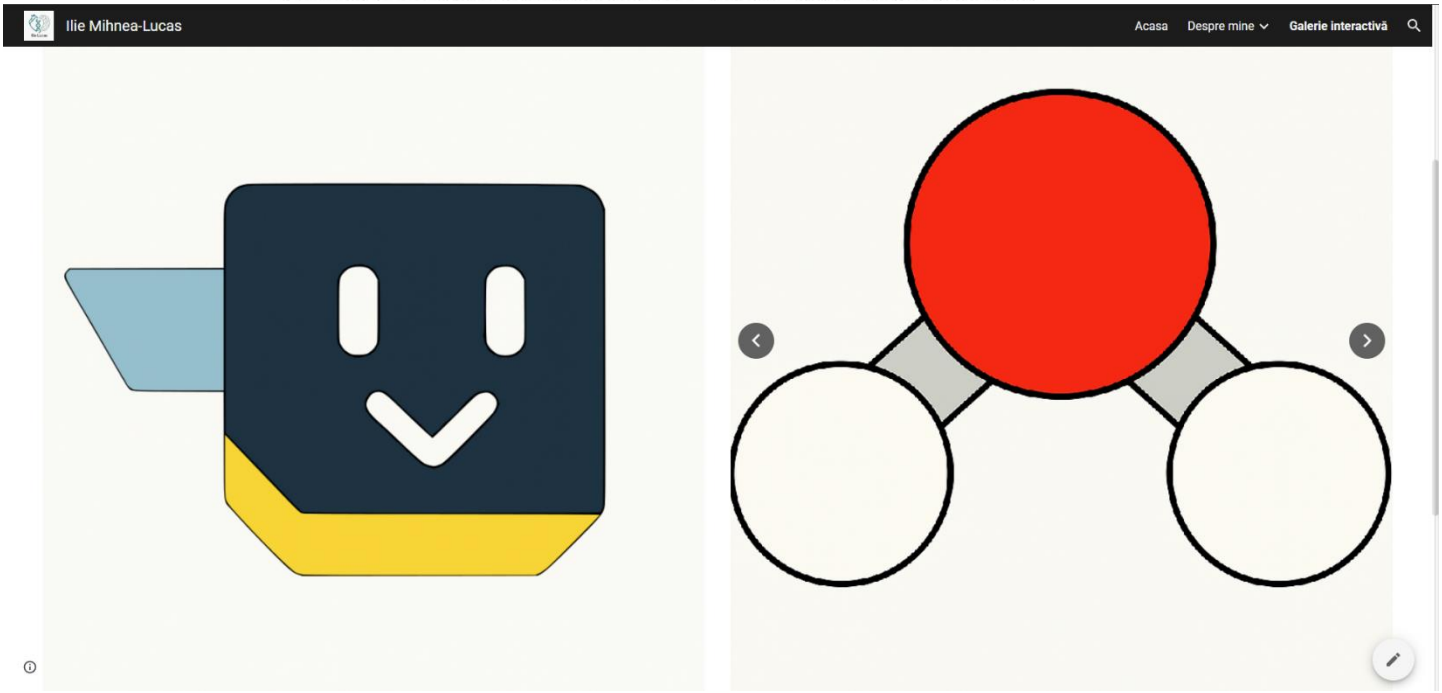


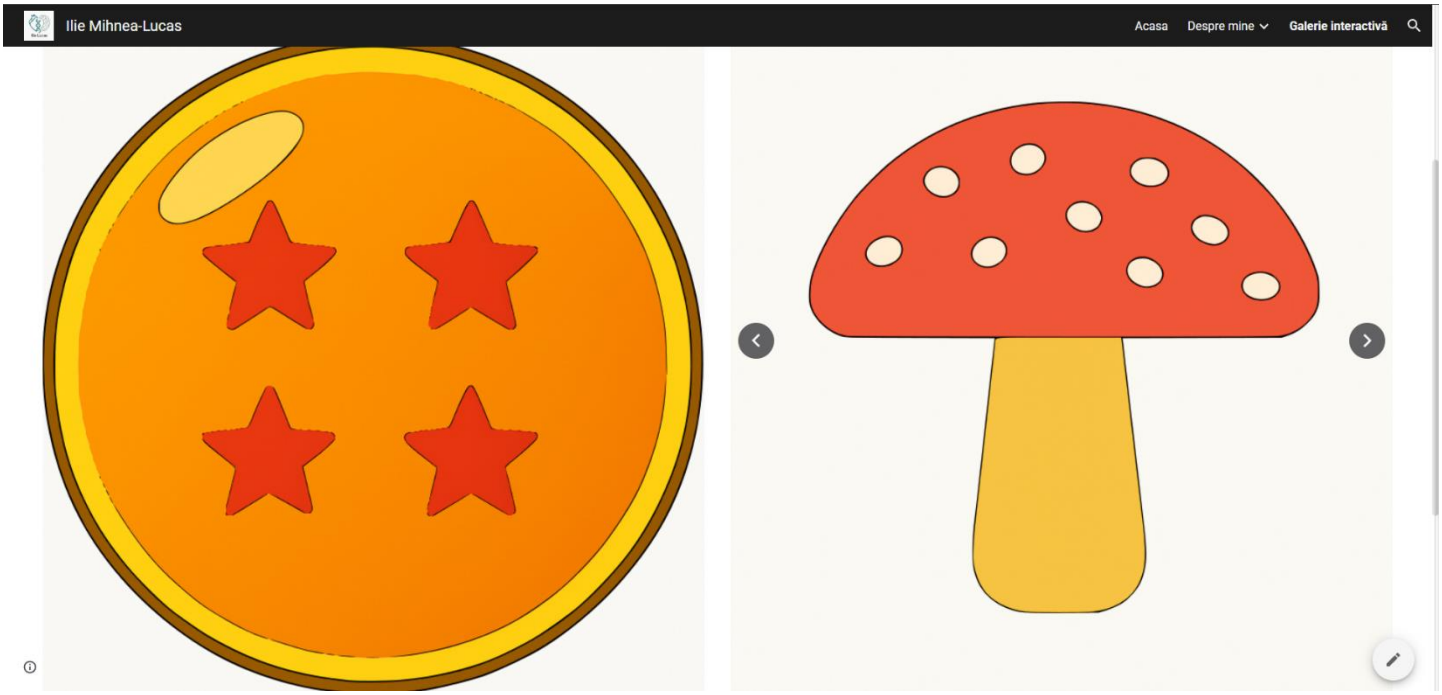


DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

Pregătire culturală - Realitate Virtuală/Augmentată (VR/AR) Educațională



Provocare: Managerul sitului „Mormântul Regilor”

Bani: 50 **Sustenabilitate:** 10 **Valoare Culturală:** 10

1. Accesul la situri

„Aici, sub soare, sunt pașii celor care au trăit înaintea noastră. Dacă deschizi porțile fără măsură, locul se va uza; dacă le închizi cu prea multă asprime, vei tăia întâlnirea oamenilor cu trecutul.”

Sfat Zenon: „Limitează cu blândețe — permite vizitarea, nu consumul.”

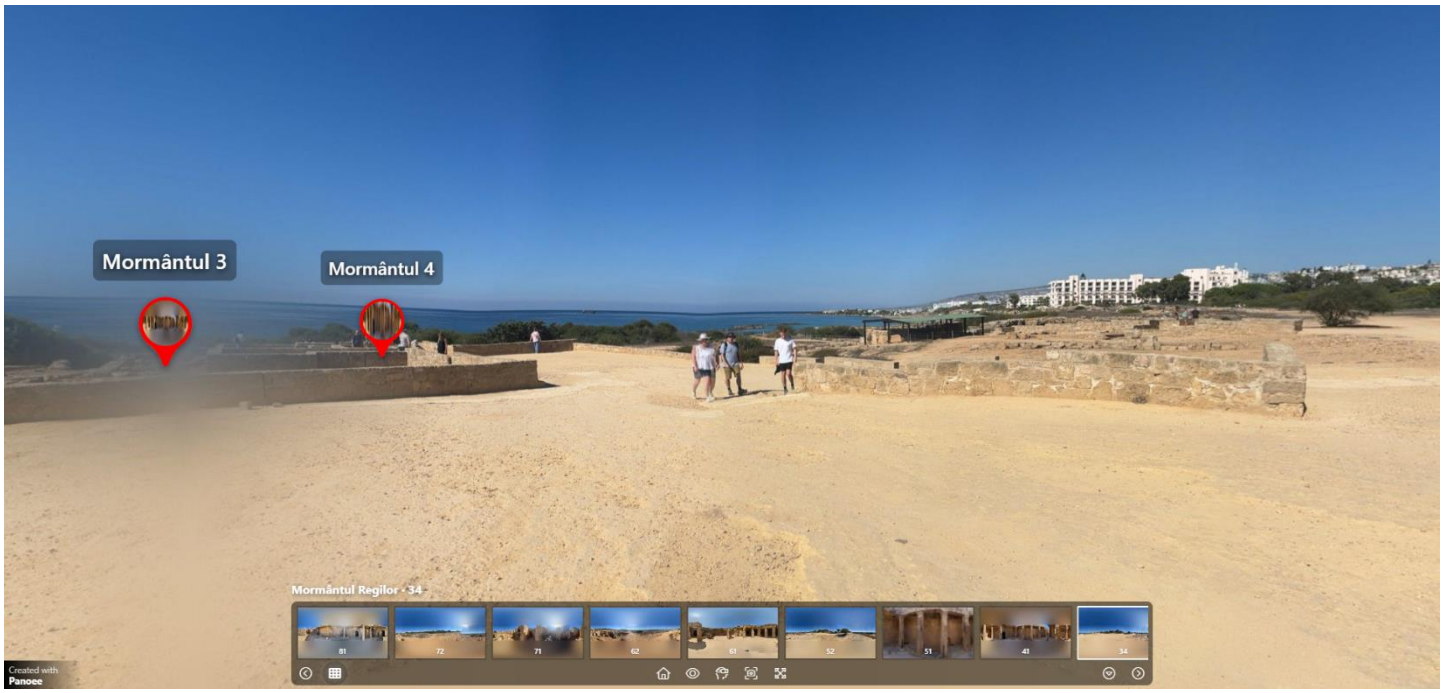
1A — Acces liber, bilete ieftine, fără limitări
Ce se întâmplă: Aflux mare, venit imediat, dar eroziune și deteriorare accelerată.

1B — Acces controlat: tururi ghidate, bilete limitate
Ce se întâmplă: Mai puțin trafic fizic, experiență de calitate, alternative VR pentru cei care rămân afară.

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

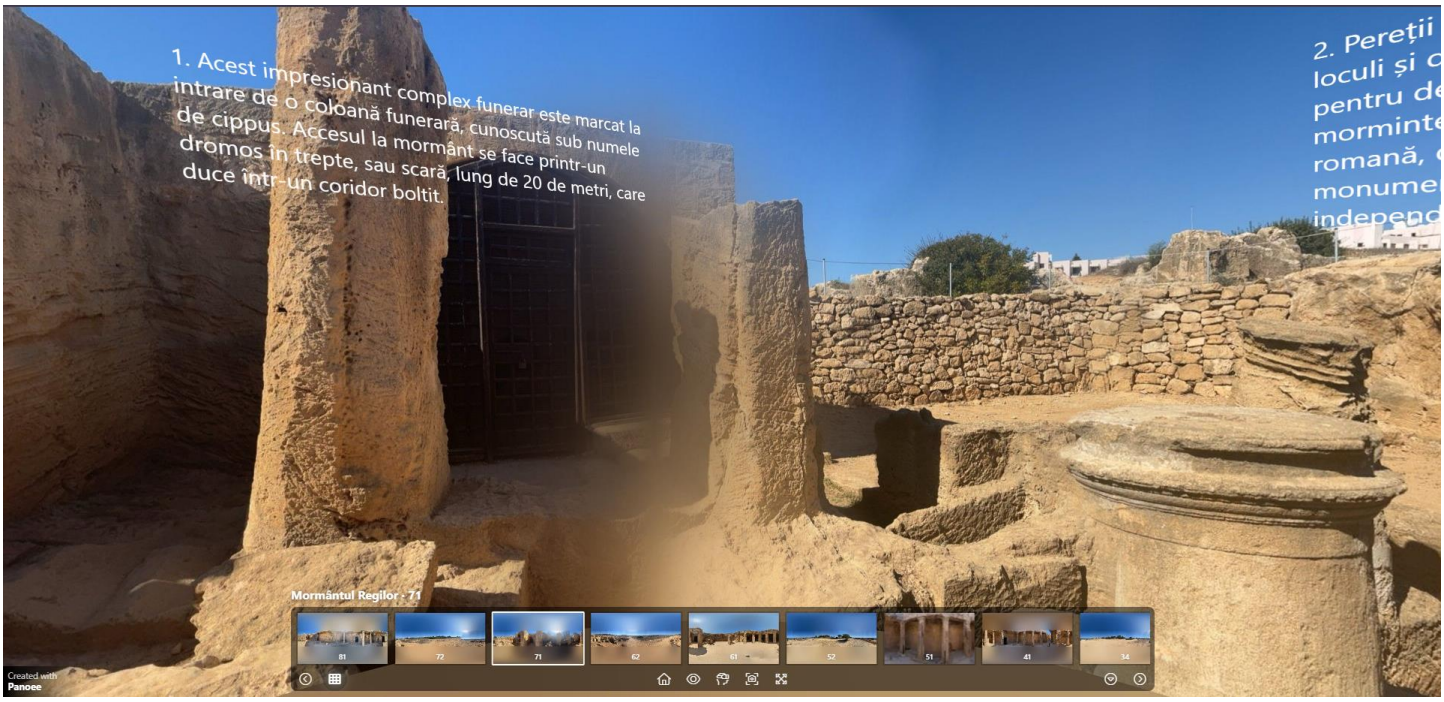




DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

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

2025-1-RO01-KA121-VET-000319082





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

Stagiu practic - Limbaj C++

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    struct Student {
        int *exam1;
        int *exam2;
        int n1, n2;
        float avg1, avg2, totalAvg;
    };

    int size = 2;
    int tmpStudents = 0;
    int numStudents = 0;
    Student *students;
    Student *tempStudents;

    students = new Student[size];

    do {
        cout << "Enter Exam results for Student " << numStudents + 1 << endl;

        students[numStudents].n1 = 0;
        students[numStudents].n2 = 0;
        students[numStudents].avg1 = 0;
        students[numStudents].avg2 = 0;
        students[numStudents].totalAvg = 0;

        int size1 = 2, size2 = 2;
        int tmpN1 = 0, tmpN2 = 0;
        int *temp1, *temp2;

        students[numStudents].exam1 = new int[size1];
        students[numStudents].exam2 = new int[size2];

        // Exam 1
        do {
            cout << "Exam 1: " << students[numStudents].n1 + 1 << " --> ";
            cin >> students[numStudents].exam1[students[numStudents].n1];
            cout << "Number saved!" << endl;

            students[numStudents].n1++;

            if (students[numStudents].n1 >= size1) {
                cout << "Out of space! Would you like to enter more? Please enter how many (Or press 0 to exit): ";
                cin >> tmpN1;

                if (tmpN1 > 0) {
                    size1 = size1 + tmpN1;
                    temp1 = new int[size1];
```

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.

```
                for (int i = 0; i < students[numStudents].n1; i++) {
                    temp1[i] = students[numStudents].exam1[i];
                }

                delete[] students[numStudents].exam1;
                students[numStudents].exam1 = temp1;
            } else {
                break;
            }
        }
    } while (students[numStudents].n1 < size1);

    cout << "Exam results 1:" << endl;
    for (int i = 0; i < students[numStudents].n1; i++) {
        cout << students[numStudents].exam1[i] << endl;
        students[numStudents].avg1 += students[numStudents].exam1[i];
    }
    students[numStudents].avg1 /= students[numStudents].n1;
    cout << "Average for Exam 1: " << students[numStudents].avg1 << endl;

    // Exam 2
    do {
        cout << "Exam 2: " << students[numStudents].n2 + 1 << " --> ";
        cin >> students[numStudents].exam2[students[numStudents].n2];
        cout << "Number saved!" << endl;

        students[numStudents].n2++;

        if (students[numStudents].n2 >= size2) {
            cout << "Out of space! Would you like to enter more? Please enter how many (Or press 0 to exit): ";
            cin >> tmpN2;

            if (tmpN2 > 0) {
                size2 = size2 + tmpN2;
                temp2 = new int[size2];

                for (int i = 0; i < students[numStudents].n2; i++) {
                    temp2[i] = students[numStudents].exam2[i];
                }

                delete[] students[numStudents].exam2;
                students[numStudents].exam2 = temp2;
            } else {
                break;
            }
        }
    } while (students[numStudents].n2 < size2);

    cout << "Exam results 2:" << endl;
    for (int i = 0; i < students[numStudents].n2; i++) {
        cout << students[numStudents].exam2[i] << 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.

```
            students[numStudents].avg2 += students[numStudents].exam2[i];
        }
        students[numStudents].avg2 /= students[numStudents].n2;
        cout << "Average for Exam 2: " << students[numStudents].avg2 << endl;

        students[numStudents].totalAvg = (students[numStudents].avg1 + students[numStudents].avg2) / 2;
        cout << "Overall average for this student: " << students[numStudents].totalAvg << endl;

        numStudents++;

        if (numStudents >= size) {
            cout << "Out of student slots! Would you like to add more students? Please enter how many (Or press 0 to exit): ";
            cin >> tmpStudents;

            if (tmpStudents > 0) {
                size = size + tmpStudents;
                tempStudents = new Student[size];

                for (int i = 0; i < numStudents; i++) {
                    tempStudents[i] = students[i];
                }

                delete[] students;
                students = tempStudents;
            } else {
                break;
            }
        }

    } while (numStudents < size);

    cout << "\n===== STUDENT SUMMARY =====\n";
    for (int s = 0; s < numStudents; s++) {
        cout << "Student " << s + 1
             << " | Exam 1 avg: " << students[s].avg1
             << " | Exam 2 avg: " << students[s].avg2
             << " | Total avg: " << students[s].totalAvg << endl;

        delete[] students[s].exam1;
        delete[] students[s].exam2;
    }

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

Am uitlizat pointeri si array-uri dinamice pentru a crea un program ce adaugă, în timpul execuției, poziții noi array-ului si afișează mediile celor două examene + media totală a examenelor a fiecărui elev introdus în array-ul respectiv.

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 <bits/stdc++.h>
using namespace std;

int main()
{
    int maxstudents = 0;
    int *exam1;
    int *exam2;
    float *avg;
    int *temp1;
    int *temp2;
    float *tempavg;
    int n = 0, tmpN = 0;

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

    exam1 = new int[maxstudents];
    exam2 = new int[maxstudents];
    avg = new float[maxstudents];

    do
    {
        cout << "Student " << n + 1 << ":" << endl;
        cout << "exam 1: ";
        cin >> exam1[n];
        cout << "exam 2: ";
        cin >> exam2[n];

        avg[n] = (exam1[n] + exam2[n]) / 2.0;

        n++;

        if (n >= maxstudents)
        {
            cout << "Out of space! Would you like to enter more students? (enter number to add, 0 to exit): ";
            cin >> tmpN;
            if (tmpN > 0)
            {
                maxstudents = maxstudents + tmpN;

                temp1 = new int[maxstudents];
                temp2 = new int[maxstudents];
                tempavg = new float[maxstudents];

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

                delete[] exam1;
                delete[] exam2;
                delete[] avg;

                exam1 = temp1;
                exam2 = temp2;
                avg = tempavg;
            }
        }

    } while (n < maxstudents);

    cout << "\nNow printing student results...\n";
    for (int i = 0; i < n; i++)
    {
        cout << "-----------------------------------" << endl;
        cout << "Student " << i + 1 << ":" << endl;
        cout << "Exam 1: " << exam1[i] << endl;
        cout << "Exam 2: " << exam2[i] << endl;
        cout << "Average: " << avg[i] << endl;
    }

    delete[] exam1;
    delete[] exam2;
    delete[] avg;

    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>
using namespace std;

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

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

    string getBrand(){return brand;}
    string getModel(){return model;}
    int getYear(){return year;}
    int getHp(){return hp;}
};
int main(){

    cars someCars[3];
    string brand;
    string model;
    int year;
    int hp;
 cout << "****************" << endl;
    for (int i=0; i<3; i++){
        cout << "Brand: "; cin >> brand;
        someCars[i].setBrand(brand);
        cout << "Model: "; cin >> model;
        someCars[i].setModel(model);
        cout << "Year: "; cin >> year;
        someCars[i].setYear(year);
        cout << "Hp: "; cin >> hp;
        someCars[i].setHp(hp);
        cout << "---------" << endl;
    }
    cout << endl;
    for (int i=0; i<3; i++){
        cout << "Brand: " << someCars[i].getBrand() << endl;
        cout << "Model: " << someCars[i].getModel() << endl;
        cout << "Year: " << someCars[i].getYear() << endl;
        cout << "Hp: " << someCars[i].getHp() << endl;
        cout << "----------" << endl;
    } 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

```cpp
#include <bits/stdc++.h>
using namespace std;

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

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

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

};

int main(){
    int size = 0;
    int n=0, tmpN=0;
    string brand;
    string model;
    int year;
    int hp;

    cout << "How many cars would you like to enter?: ";
    cin >> size;

    cars* someCars = new cars[size];
    cars* temp;

    do{
        cout << "Car number: " << n+1 << endl;

        cout << "Brand: "; cin >> brand;
        someCars[n].setBrand(brand);

        cout << "Model: "; cin >> model;
        someCars[n].setModel(model);

        cout << "Year: "; cin >> year;
        someCars[n].setYear(year);
```

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.

```
49
50          cout << "Hp: "; cin >> hp;
51          someCars[n].setHp(hp);
52
53          cout << "Car saved successfully!" << endl;
54          n++;
55
56          if (n>=size){
57              cout << "Out of space! More cars? (0 to exit)";
58              cin >> tmpN;
59
60              if (tmpN > 0){
61                  size = size + tmpN;
62                  temp = new cars[size];
63                  for (int i = 0; i<n; i++) {
64                      temp[i] = someCars[i];
65                  }
66
67                  delete [] someCars;
68                  someCars = temp;
69              }
70          }
71      }while (n<size);
72
73      cout << endl;
74      cout << "------------Results------------";
75      cout << endl;
76      for (int i=0; i<n; i++){
77          cout << "Brand: " << someCars[i].getBrand() << endl;
78          cout << "Model: " << someCars[i].getModel() << endl;
79          cout << "Year: " << someCars[i].getYear() << endl;
80          cout << "Hp: " << someCars[i].getHp() << endl;
81          cout << "----------" << endl;
82      }
83
84      return 0;
85  }
86
```

Am uitlizat clase si obiecte pentru a crea un program ce afișeaza diferite informații legate de acestea (brand, model, an, cai putere).

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
#include <iostream>
using namespace std;

int main(int argc, char** argv)
{
    int side1=0, side2=0;


    cout << "You have entered: " << argc << " arguments: " << "\n";

    for(int i=0; i < argc; i++){
        cout << argv[i] << "\n";
    }

    if(argc < 3){
        cout << "You have not entered the correct amount of arguments! Enter more: \n";
        cout << "Side 1: ";
        cin >> side1;
        cout << "Side 2: ";
        cin >> side2;
        cout << "The perimeter is:" << side1*2 + side2*2;
    }
    else{
            side1 = stoi(argv[1]);
            side2 = stoi(argv[2]);
        cout << "The perimeter is: " << (side1 + side2)*2;
    }


    return 0;
}
```

Am uitlizat parametrii liniei de comandă pentru a afișa perimetrul unui dreptunghi.

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
#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;}
};

void addPatient(patient *cl,int index);
patient *growArray(patient *cl,int arraySize,int index);
void printArray(patient *clinic, int index);

// NEW PROTOTYPES:
int searchByName(patient *clinic, int index);
void editPatient(patient *clinic,int index);
void deletePatient(patient *clinic,int &index,int pos);
void searchByMedicine(patient *clinic, int index);

//-----------------------------------------------
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

2025-1-RO01-KA121-VET-000319082

```
49          cout<<"What would you like to do? ";
50          cin>>userChoice;
51          cout<<"--------------------------------"<<endl;
52
53          if (userChoice == 1){
54              if (index>=ArraySize){
55                  ArraySize++;
56                  clinic=growArray(clinic,ArraySize,index);
57              }
58              addPatient(clinic,index);
59              index++;
60
61          }else if(userChoice==2){
62              int pos=searchByName(clinic,index);
63              if(pos!=-1) editPatient(clinic,pos);
64
65          }else if(userChoice==3){
66              int pos=searchByName(clinic,index);
67              if(pos!=-1) deletePatient(clinic,index,pos);
68
69          }else if (userChoice==4){
70              searchByName(clinic,index);
71
72          }else if(userChoice==5){
73              searchByMedicine(clinic,index);
74
75          }else if(userChoice==6){
76              printArray(clinic,index);
77
78          }
79      }
80      return 0;
81  }
82
83  //-----------------------------------------------
84  patient *growArray(patient *clinic,int ArraySize, int index){
85      patient *tmp;
86      tmp=new patient[ArraySize];
87      for (int i=0;i<index;i++){
88          tmp[i]=clinic[i];
89      }
90      delete [] clinic;
91      return tmp;
92  }
93
94  //-----------------------------------------------
95  void addPatient(patient *cl,int index){
96      string name,phone,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

```
97      int age;
98
99      cout<<"Name: ";
100     cin>>name;
101     cl[index].setName(name);
102     cout<<"Age: ";
103     cin>>age;
104     cl[index].setAge(age);
105     cout<<"Phone: ";
106     cin>>phone;
107     cl[index].setPhone(phone);
108     cout<<"Medicine Given: ";
109     cin>>med;
110     cl[index].setMedicine(med);
111     cout<<"Patient Successfully Added!\n";
112 }
113
114 //-------------------------------------------
115 void printArray(patient *clinic, int index){
116     cout<<"PRINTING ALL DATA"<<endl<<"-----------------"<<endl;
117     for (int i=0;i<index;i++){
118         cout<<"----- Patient "<<i+1<<" -----"<<endl;
119         cout<<"Name: "<<clinic[i].getName()<<endl;
120         cout<<"Age: "<<clinic[i].getAge()<<endl;
121         cout<<"Phone: "<<clinic[i].getPhone()<<endl;
122         cout<<"Medicine: "<<clinic[i].getMedicine()<<endl;
123     }
124 }
125
126 //-------------------------------------------
127 // returns chosen index or -1
128 int searchByName(patient *clinic, int index){
129     string name;
130     cout<<"Enter Name to search: ";
131     cin>>name;
132
133     int found[100];
134     int cnt=0;
135
136     for(int i=0;i<index;i++){
137         if(clinic[i].getName()==name){
138             cout<<cnt+1<<") Index: "<<i<<" | "<<clinic[i].getName()<<", "<<clinic[i].getAge()<<", "<<clinic[i].getPhone()<<", "<<clinic[i].getMedicine()<<endl;
139             found[cnt]=i;
140             cnt++;
141         }
142     }
143
144     if(cnt==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.

```
145         cout<<"No patient found!\n";
146         return -1;
147     }
148
149     if(cnt==1) return found[0];
150
151     int choice;
152     cout<<"Choose which one (1-"<<cnt<<"): ";
153     cin>>choice;
154     return found[choice-1];
155 }
156 void editPatient(patient *clinic,int pos){
157     int ch;
158     cout<<"What do you want to edit?\n";
159     cout<<"1. Name\n2. Age\n3. Phone\n4. Medicine\n";
160     cin>>ch;
161     if(ch==1){
162         string x; cout<<"New Name: "; cin>>x; clinic[pos].setName(x);
163     }else if(ch==2){
164         int x; cout<<"New Age: "; cin>>x; clinic[pos].setAge(x);
165     }else if(ch==3){
166         string x; cout<<"New Phone: "; cin>>x; clinic[pos].setPhone(x);
167     }else if(ch==4){
168         string x; cout<<"New Medicine: "; cin>>x; clinic[pos].setMedicine(x);
169     }
170     cout<<"Updated!\n";
171 }
172 void deletePatient(patient *clinic,int &index,int pos){
173     for(int i=pos;i<index-1;i++){
174         clinic[i]=clinic[i+1];
175     }
176     index--;
177     cout<<"Patient deleted!\n";
178 }
179 void searchByMedicine(patient *clinic, int index){
180     string med;
181     cout<<"Medicine to search: ";
182     cin>>med;
183     bool f=false;
184     for(int i=0;i<index;i++){
185         if(clinic[i].getMedicine()==med){
186             f=true;
187             cout<<"Patient "<<i+1<<" -> "<<clinic[i].getName()<<", "<<clinic[i].getAge()<<", "<<clinic[i].getPhone()<<endl;
188         }
189     }
190     if(!f) cout<<"No match.\n";
191 }
192
```

Am folosit toate noțiunile noi pentru a crea un program cu un meniu ce adaugă un pacient nou, editeaza un pacient deja introdus, șterge un pacient, caută un pacient după nume/medicamentul dat și afișează toți pacienții, în funcție de numărul introdus.

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

Stagiu practic - Arduino

arduino_first.ino

```
const int trigPin = 4;
const int echoPin = 5;
const int ledPin  = 10;

long duration;
long cm;

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

void loop() {

  // send pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // receive echo
  duration = pulseIn(echoPin, HIGH);

  // convert to cm
  cm = duration / 27 / 2;

  Serial.print("Distance: ");
  Serial.print(cm);
  Serial.println("cm");

  // condition
  if (cm <= 30) {
    digitalWrite(ledPin, HIGH);   // LED on
  } else {
    digitalWrite(ledPin, LOW);    // LED off
  }

  delay(100);
}

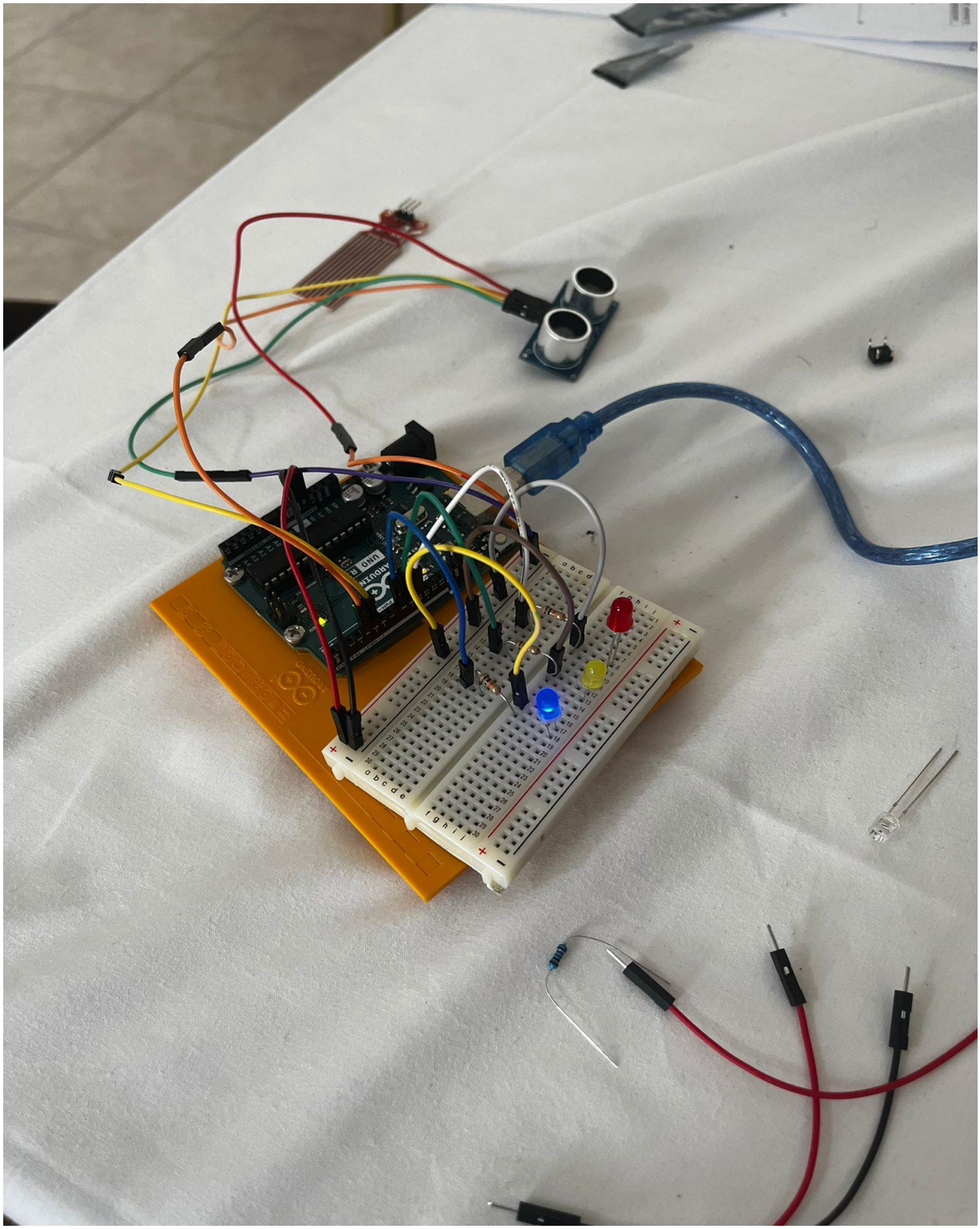
```

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.

```
19  void loop() {
20
21    // send pulse
22    digitalWrite(trigPin, LOW);
23    delayMicroseconds(2);
24    digitalWrite(trigPin, HIGH);
25    delayMicroseconds(10);
26    digitalWrite(trigPin, LOW);
27
28    // receive echo
29    duration = pulseIn(echoPin, HIGH);
30
31    // convert to cm
32    cm = duration / 27 / 2;
33
34    Serial.print("Distance: ");
35    Serial.print(cm);
36    Serial.println("cm");
37
38    // condition
39    if (cm <= 50) {
40      digitalWrite(ledPin, HIGH);   // LED on
41    } else {
42      digitalWrite(ledPin, LOW);    // LED off
43    }
44    if (cm > 50 && cm <= 100) {
45      digitalWrite(ledPin1, HIGH);   // LED on
46    } else {
47      digitalWrite(ledPin1, LOW);    // LED off
48    }
49    if (cm > 100) {
50      digitalWrite(ledPin2, HIGH);   // LED on
51    } else {
52      digitalWrite(ledPin2, LOW);    // LED off
53    }
54
55
56    delay(100);
57  }
58
```

Program care aprinde un beculeț specific, în funcție de distanța citită de senzorul ultrasonic HC-SR04.

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.



Circuitul programului anterior

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

```
1
2   #define LED_PIN     2
3   #define POWER_PIN   7
4   #define SIGNAL_PIN  A5
5   #define THRESHOLD   300
6
7   int value = 0;
8
9   void setup() {
10    Serial.begin(9600);
11    pinMode(LED_PIN,   OUTPUT);   // configure D2 pin as an OUTPUT
12    pinMode(POWER_PIN, OUTPUT);   // configure D7 pin as an OUTPUT
13    digitalWrite(POWER_PIN, LOW); // turn the sensor OFF
14    digitalWrite(LED_PIN,   LOW); // turn LED OFF
15  }
16
17  void loop() {
18    digitalWrite(POWER_PIN, HIGH);  // turn the sensor ON
19    delay(10);                      // wait 10 milliseconds
20    value = analogRead(SIGNAL_PIN); // read the analog value from sensor
21    digitalWrite(POWER_PIN, LOW);   // turn the sensor OFF
22
23    if (value > THRESHOLD) {
24      Serial.print("The water is detected");
25      digitalWrite(LED_PIN, HIGH);  // turn LED ON
26    } else {
27      digitalWrite(LED_PIN, LOW);   // turn LED OFF
28    }
29  }
30
```

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.

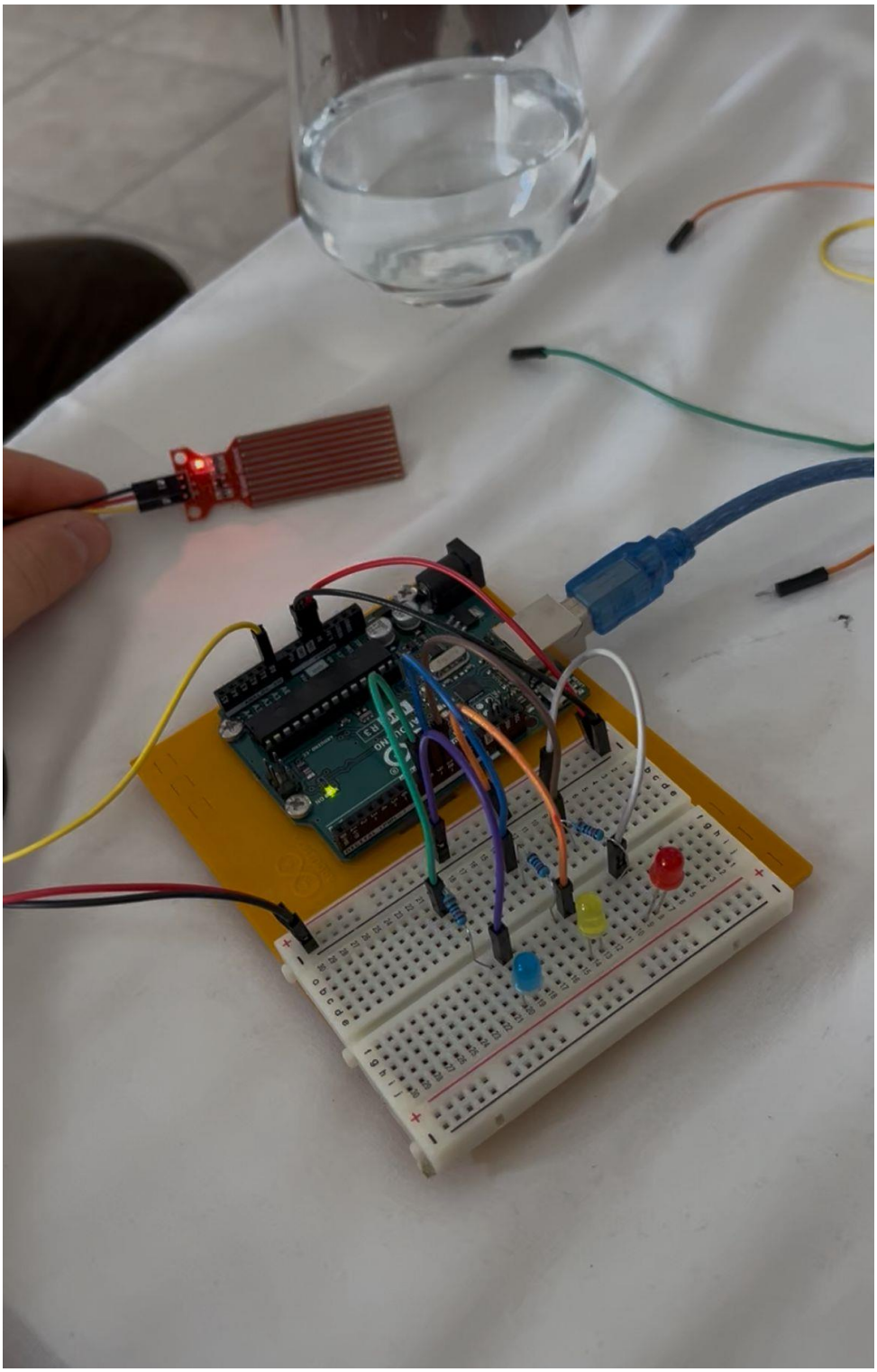
```
int lowerThreshold = 460;
int upperThreshold = 520;
#define sensorPin A0
int val = 0;
int blueLED = 8;
int yellowLED = 9;
int redLED = 10;
void setup() {
  Serial.begin(9600);

  pinMode(blueLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(redLED, OUTPUT);

  digitalWrite(blueLED, LOW);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, LOW);
}
void loop() {
  int level = readSensor();
  if (level == 0) {
    Serial.println("Water Level: Empty");
    digitalWrite(blueLED, LOW);
    digitalWrite(yellowLED, LOW);
    digitalWrite(redLED, LOW);
  } else if (level <= lowerThreshold) {
    Serial.println("Water Level: Low");
    digitalWrite(blueLED, HIGH);
    digitalWrite(yellowLED, LOW);
    digitalWrite(redLED, LOW);
  } else if (level <= upperThreshold) {
    Serial.println("Water Level: Medium");
    digitalWrite(blueLED, LOW);
    digitalWrite(yellowLED, HIGH);
    digitalWrite(redLED, LOW);
  } else {
    Serial.println("Water Level: High");
    digitalWrite(blueLED, LOW);
    digitalWrite(yellowLED, LOW);
    digitalWrite(redLED, HIGH);
  }
  delay(1000);
}
int readSensor() {
  val = analogRead(sensorPin);
  return val;
}
```

Program ce aprinde un beculeț specific, în funcție de nivelul apei de pe senzorul de apă.

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.



Circuitul programului anterior

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
#define flameSensorPin 2
#define fireLED 8
#define normalLED 9
#define buzzerPin 10

void setup() {
  pinMode(flameSensorPin, INPUT);
  pinMode(fireLED, OUTPUT);
  pinMode(normalLED, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  digitalWrite(fireLED, LOW);
  digitalWrite(normalLED, LOW);

  Serial.begin(9600);
}

void loop() {
  int flameState = digitalRead(flameSensorPin);

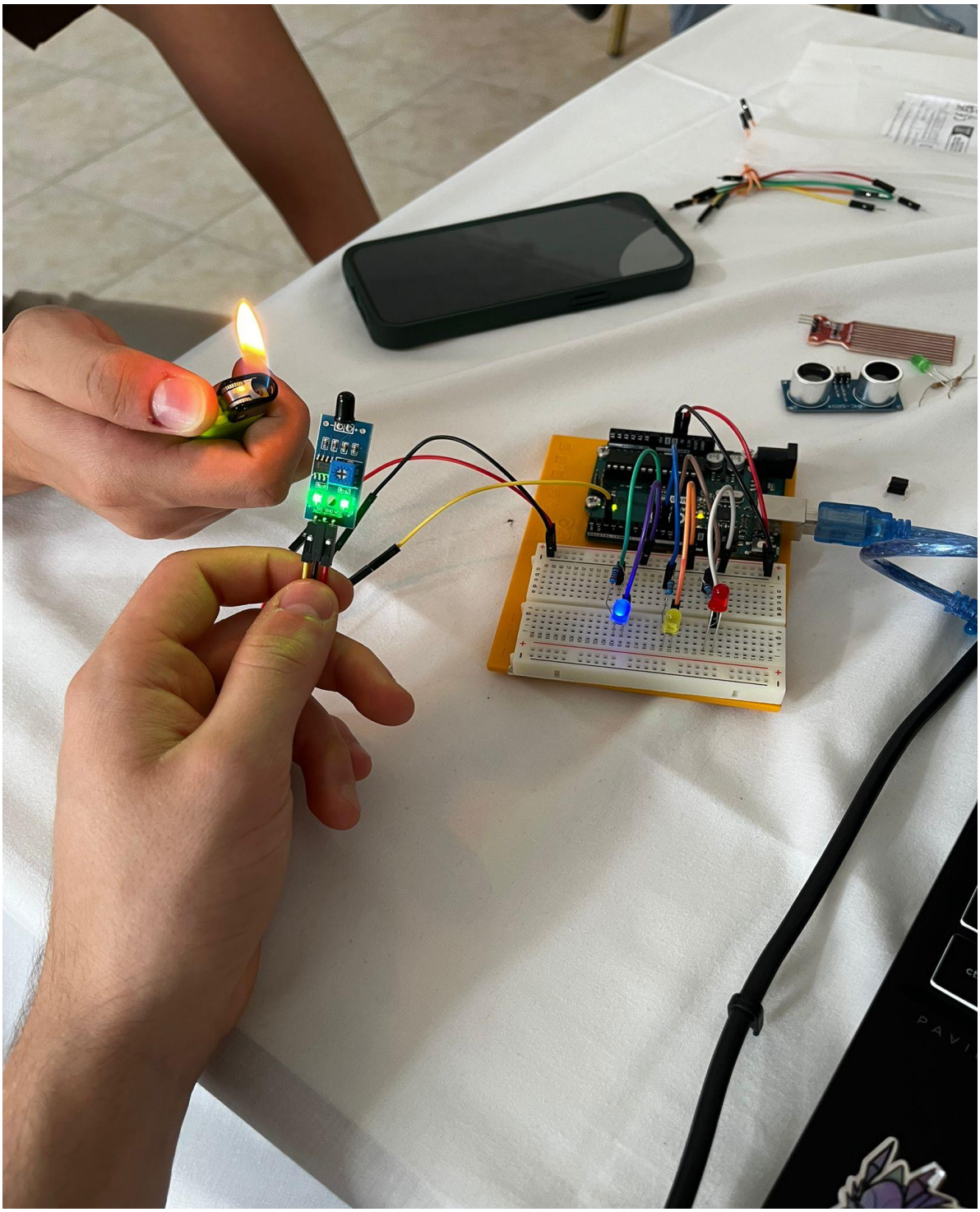
  if (flameState == LOW) {   // FIRE detected
    digitalWrite(fireLED, HIGH);
    digitalWrite(normalLED, LOW);
    tone(buzzerPin, 250);   // start buzzer
    Serial.println("** FIRE DETECTED **");
  }
  else {                    // no fire
    digitalWrite(fireLED, LOW);
    digitalWrite(normalLED, HIGH);
    noTone(buzzerPin);       // stop buzzer
    Serial.println("No fire");
  }

  delay(100);
}

```

Program care aprinde beculețul fireLED (cel albastru) cand detecteaza flacără/foc și revine la normal (se aprinde normalLED, cel galben) cand nu detectează nimic.

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.



Circuitul programului anterior

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

```ino
#define flameSensorPin 2
#define fireLED 8
#define normalLED 9
#define buzzerPin 10

void setup() {
  pinMode(flameSensorPin, INPUT);
  pinMode(fireLED, OUTPUT);
  pinMode(normalLED, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  digitalWrite(fireLED, LOW);
  digitalWrite(normalLED, LOW);

  Serial.begin(9600);
}

void loop() {
  int flameState = digitalRead(flameSensorPin);

  if (flameState == LOW) {   // FIRE detected
    digitalWrite(fireLED, HIGH);
    digitalWrite(normalLED, LOW);
    tone(buzzerPin, 250);   // start buzzer
    Serial.println("** FIRE DETECTED **");
  }
  else {                    // no fire
    digitalWrite(fireLED, LOW);
    digitalWrite(normalLED, HIGH);
    noTone(buzzerPin);      // stop buzzer
    Serial.println("No fire");
  }

  delay(100);
}

```

Program care pornește un buzzer când detectează foc.

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.