



ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

კომპანიაში დასაქმებული ადამიანების მონაცემთა საჩვენებელი  
აპლიკაციის პროექტირება ობიექტზე ორიენტირებული  
პროგრამირების მიდგომით

საკურო პროექტი საგანში:

ობიექტზე ორიენტირებული დაპროგრამება და აპლიკაციები

კურსი IV, ჯგუფი 108134

სტუდენტის სახელი და გვარი: ალი რაჰიმი

სასწავლო კურსის წამყვანი პროფესორი: ინგა აბულაძე

საქართველოს ტექნიკური უნივერსიტეტი

თბილისი, 0160, საქართველო

2025 წელი

## სარჩევი

1.	შესავალი .....	3
2.	ამოცანის დასმა .....	4
3.	ამოცანის გადაწყვეტის ალგორითმი.....	5
4.	დასმული ამოცანის გადაწყვეტის ალგორითმის შესაბამისი პროგრამული კოდი .....	7
4.1.	მონაცემთა ბაზის სტრუქტურის განსაზღვრა.....	7
4.2.	მონაცემთა ინიციალიზაცია: .....	8
4.3.	მომხმარებლის ინტერფეისის აგება .....	8
4.4.	მონაცემთა ჩატვირთვა და დინამიკურად ჩვენება.....	9
4.5.	გამოსახულებების დამუშავება .....	11
4.6.	ობიექტზე ორიენტირებული პროგრამირების გამოყენება .....	11
5.	პროგრამის რეალიზაციის შედეგი - კოდი.....	12
6.	დასკვნა .....	14
7.	გამოყენებული ლიტერატურა .....	15
8.	დანართი.....	16

## 1. შესავალი

შემუშავებული პროექტი წარმოადგენს .NET Windows Forms აპლიკაციას, რომელიც შექმნილია სხვადასხვა კომპანიაში დასაქმებული თანამშრომლების შესახებ ინფორმაციის მოსახერხებელი და ორგანიზებული ჩვენებისთვის. აპლიკაცია დაპროგრამებულია C# ენაზე, ხოლო მონაცემთა შენახვა ხორციელდება SQLite მონაცემთა ბაზის საშუალებით.

სისტემა მომხმარებლებს სთავაზობს შემდეგ შესაძლებლობებს:

- დაათვალიერონ წინასწარ განსაზღვრული კომპანიების სია, თითოეულთან დაკავშირებული ძირითადი მონაცემებით: დასახელება, აღწერა, ლოგო, ტელეფონის ნომერი და ელ-ფოსტა;
- ნახონ კონკრეტულ კომპანიაში დასაქმებული თანამშრომლების დეტალური ინფორმაცია, მათ შორის: სრული სახელი, ასაკი, პოზიცია, ხელფასი, საკონტაქტო მონაცემები და ფოტო.

აპლიკაციის შესამუშავებლად გამოყენებულია ობიექტზე ორიენტირებული დაპროგრამების პრინციპები, ხოლო ჩაშენებული მონაცემთა ბაზა უზრუნველყოფს ლოკალურ, სწრაფ წვდომას მონაცემებზე სერვერის გარეშე.

პროექტის მიზანია აჩვენოს, თუ როგორ შეიძლება სამუშაო მაგიდის აპლიკაციების გამოყენება სტრუქტურირებული ბიზნეს ინფორმაციის ეფექტური წარმოდგენისთვის. აპლიკაციას შეიძლება მოეძებნოს პრაქტიკული გამოყენება სხვადასხვა ბიზნეს პრობლემის გადაწყვეტაში, მაგალითად:

- მენეჯერებისთვის, რომლებსაც სჭირდებათ თანამშრომელთა მონაცემებზე წვდომა;
- ახალი თანამშრომლებისთვის, რათა მარტივად გაეცნონ კოლეგებს;
- ან ნებისმიერ ორგანიზაციულ სისტემაში, სადაც თანამშრომლების კატალოგის დათვალიერებაა საჭირო.

## 2. ამოცანის დასმა

თანამედროვე ორგანიზაციებსა და კომპანიებში მნიშვნელოვანია თანამშრომელთა შესახებ ინფორმაციის მოწესრიგებული, სწრაფი და მარტივი წვდომა. მიუხედავად იმისა, რომ არსებობს მრავალი კომპლექსური მართვის სისტემა, ხშირად წარმოიშვება საჭიროება მარტივი და მსუბუქი აპლიკაციისა, რომელიც უზრუნველყოფს მხოლოდ ინფორმაციის ვიზუალიზაციას და არა მის მართვას.

ამ პროექტის მიზანია ისეთი desktop აპლიკაციის შექმნა, რომელიც საშუალებას აძლევს მომხმარებელს წინასწარ განსაზღვრული კომპანიებისა და მათში დასაქმებული თანამშრომლების შესახებ ინფორმაციის ეფექტურად დათვალიერებას. ამის შესასრულებლად საჭიროა შემდეგი მოთხოვნები:

- სხვადასხვა კომპანიის პროფილების გამოჩენა შესაბამისი ატრიბუტებით (დასახელება, აღწერა, ლოგო, ტელეფონი, ელ-ფოსტა);
- თითოეული კომპანიისთვის თანამშრომლების სიის გამოტანა დეტალური მონაცემებით (სახელი, ასაკი, პოზიცია, ხელფასი, საკონტაქტო ინფორმაცია, ფოტო);
- მარტივი, მომხმარებლისთვის გასაგები ინტერფეისის შექმნა;
- მონაცემთა ლოკალურად შენახვა და წვდომა SQLite ბაზის გამოყენებით;
- აპლიკაციის სტაბილურად მუშაობა .NET Windows Forms ტექნოლოგიაზე დაფუძნებულ გარემოში.



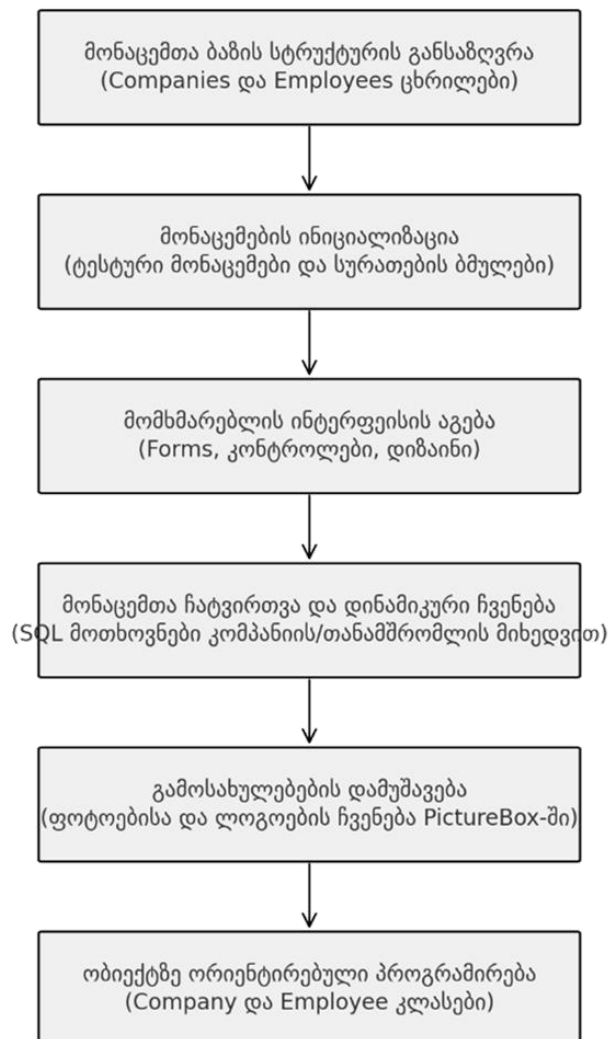
ნახ. 1. თანამშრომელთა ნახაზი (Dall.E 3)

### 3. ამოცანის გადაწყვეტის ალგორითმი

აპლიკაციის ამოცანის შესრულება ეფუძნება სტრუქტურირებულ, თანმიმდევრულ ალგორითმს, რომელიც უზრუნველყოფს მონაცემთა სწრაფად წარდგენასა და მომხმარებლისთვის მარტივი ინტერფეისის შექმნას. ალგორითმი მოიცავს შემდეგ ეტაპებს:

1. მონაცემთა ბაზის სტრუქტურის განსაზღვრა:
  - შეიქმნა SQLite ბაზა ორი ძირითადი ცხრილით: Companies და Employees.
  - თითოეულ კომპანიას აქვს უნიკალური იდენტიფიკატორი და შესაბამისი ატრიბუტები (დასახელება, აღწერა, ტელეფონი, ელ.ფოსტა, ლოგო).
  - Employees ცხრილში ინახება თანამშრომელთა დეტალური ინფორმაცია და ბმული შესაბამის კომპანიაზე უცხო გასაღების (Foreign Key) მეშვეობით.
2. მონაცემთა ინიციალიზაცია:
  - ბაზაში წინასწარ შეყვანილია რამდენიმე კომპანიისა და თანამშრომლის მონაცემი ტესტირების მიზნით.
  - ლოგოები და ფოტოსურათები ინახება ლოკალური ფაილის გზით და იტვირთება აპლიკაციაში შესაბამის ადგილას.
3. მომხმარებლის ინტერფეისის აგება:
  - შექმნილია მთავარი ფორმა, რომელიც შეიცავს კომპანიის სიის ჩვენების საშუალებას (მაგ. ListView ან DataGridView).
  - კომპანიის არჩევისას, ინტერფეისის მეორე ნაწილი აჩვენებს მის შესახებ დეტალურ ინფორმაციას და შესაბამის თანამშრომელთა სიას.
  - თანამშრომლის არჩევისას ჩანს მისი დეტალური პროფილი — სურათითა და ყველა პირადი მონაცემით.
4. მონაცემთა ჩატვირთვა და დინამიკურად ჩვენება:
  - აპლიკაციის გაშვებისას ხდება ბაზასთან კავშირი და კომპანიის სიის ჩატვირთვა.
  - მომხმარებლის მიერ კომპანიის არჩევისას ხდება SQL მოთხოვნით შესაბამისი თანამშრომლების გამოტანა.
  - ინფორმაცია დინამიკურად აისახება ინტერფეისზე, ყოველ ჯერზე კომპანიის ან თანამშრომლის შეცვლისას.
5. გამოსახულებების დამუშავება:
  - ლოგოსა და თანამშრომლის ფოტოს მისამართები ინახება ბაზაში როგორც ტექსტი (ფაილის ლოკაცია).

- აპლიკაცია იტვირთავს ამ სურათებს PictureBox კომპონენტში და აჩვენებს ვიზუალურად.
6. ობიექტზე ორიენტირებული პროგრამირების გამოყენება:
- შექმნილია კლასები Company და Employee, რომლებიც წარმოადგენს მონაცემთა სტრუქტურას.
  - მონაცემების დამუშავება და ჩვენება ხორციელდება ამ კლასებზე დაფუძნებით.



ნახ. 2. ამოცანის გადაწყვეტის ბლოკ-სქემა

## 4. დასმული ამოცანის გადაწყვეტის ალგორითმის შესაბამისი პროგრამული კოდი

### 4.1. მონაცემთა ბაზის სტრუქტურის განსაზღვრა

ნახ. 3-ში წარმოდგენილია კოდი, რომლის მეშვეობითაც ხდება მთავარი პროგრამის დაკავშირება მონაცემთა ბაზასთან, მისი დასახელების ინიციალიზაცია და შექმნა არ არსებობის შემთხვევაში.

```
public static class DatabaseHelper
{
    private static readonly string dbFilePath = "company_db.sqlite";
    private static readonly string connectionString = $"Data Source={dbFilePath};Version=3;";

    2 reference
    public static string ConnectionString => connectionString;

    1 reference
    public static void InitializeDatabase()
    {
        if (!File.Exists(dbFilePath))
        {
            SQLiteConnection.CreateFile(dbFilePath);
        }
    }
}
```

ნახ. 3. მონაცემთა ბაზის შექმნა

Employees და Companies ცხრილების შექმნისთვის დაწერილია ნახ. 4-ში მოცემული კოდი, რომელიც ამოწმებს არსებობს თუ არა ეს ცხრილები და არ არსებობის შემთხვევაში კი შეიქმნება ისინი.

```
using (var connection = new SQLiteConnection(connectionString))
{
    connection.Open();

    string createCompaniesTable = @"
        CREATE TABLE IF NOT EXISTS Companies (
            Id INTEGER PRIMARY KEY AUTOINCREMENT,
            Name TEXT NOT NULL,
            LogoPath TEXT,
            Description TEXT,
            PhoneNumber TEXT,
            Email TEXT
        );";

    string createEmployeesTable = @"
        CREATE TABLE IF NOT EXISTS Employees (
            Id INTEGER PRIMARY KEY AUTOINCREMENT,
            CompanyId INTEGER NOT NULL,
            FirstName TEXT NOT NULL,
            LastName TEXT NOT NULL,
            Age INTEGER,
            Position TEXT,
            Salary REAL,
            Email TEXT,
            PhoneNumber TEXT,
            PhotoPath TEXT,
            FOREIGN KEY (CompanyId) REFERENCES Companies(Id)
        );";

    using (var command = new SQLiteCommand(createCompaniesTable, connection))
    {
        command.ExecuteNonQuery();
    }

    using (var command = new SQLiteCommand(createEmployeesTable, connection))
    {
        command.ExecuteNonQuery();
    }
}
```

ნახ. 4. მონაცემთა ბაზაში ცხრილების შექმნისთვის შესაბამისი კოდი

## 4.2. მონაცემთა ინიციალიზაცია:

ნახ. 5-ში შეგვიძლია ვნახოთ ბაზაში სატესტო მონაცემთა დამატებისთვის დაწერილ პროგრამულ კოდს, რომლის მეშვეობითაც აპლიკაცია ჯერ ამოწმებს არსებობს თუ არა ბაზაში რაიმე მონაცემი და არ არსებობის შემთხვევაში დაემატება სატესტო მონაცემები.

```
public static void SeedTestData()
{
    using (var connection = new SqlConnection(ConnectionString))
    {
        connection.Open();

        // Check if Companies table is already filled
        string checkQuery = "SELECT COUNT(*) FROM Companies";
        using (var checkCmd = new SqlCommand(checkQuery, connection))
        {
            long count = (long)checkCmd.ExecuteScalar();
            if (count > 0) return; // Already seeded
        }

        // Insert sample companies
        string[] insertCompanies = new string[]
        {
            @"INSERT INTO Companies (Name, LogoPath, Description, PhoneNumber, Email)
            VALUES ('Tech Solutions Ltd.', 'TechSolutionsLtd.png', 'IT and software services provider.', '123-456-7890', 'info@techsolutions.com');",
            @"INSERT INTO Companies (Name, LogoPath, Description, PhoneNumber, Email)
            VALUES ('Green Earth Corp', 'GreenEarthCorp.png', 'Environmental engineering company.', '987-654-3210', 'contact@greenearth.com');",
            @"INSERT INTO Companies (Name, LogoPath, Description, PhoneNumber, Email)
            VALUES ('HealthPlus Inc.', 'HealthPlusInc.png', 'Healthcare and medical services.', '456-789-0123', 'support@healthplus.com');",
            @"INSERT INTO Companies (Name, LogoPath, Description, PhoneNumber, Email)
            VALUES ('FinSecure LLC', 'FinSecureLLC.png', 'Financial consulting and auditing.', '321-654-0987', 'info@finsecure.com');",
        };

        foreach (var insertCompany in insertCompanies)
        {
            using (var cmd = new SqlCommand(insertCompany, connection))
            {
                cmd.ExecuteNonQuery();
            }
        }

        // Insert employees
        string insertEmployees = @"
INSERT INTO Employees (CompanyId, FirstName, LastName, Age, Position, Salary, Email, PhoneNumber, PhotoPath)
VALUES
-- Tech Solutions Ltd (1)
(1, 'Ali', 'Rahimi', 22, 'AI/ML Engineer', 280000, 'ali.rahimi@techsolutions.com', '579-2421', 'AliRahimi.png'),
(1, 'John', 'Doe', 28, 'Software Engineer', 68000, 'john.doe@techsolutions.com', '555-1111', 'JohnDoe.png'),
(1, 'Sarah', 'Mills', 32, 'Project Manager', 72000, 'sarah.mills@techsolutions.com', '555-1112', 'SarahMills.png'),
(1, 'David', 'Clark', 30, 'QA Analyst', 58000, 'david.clark@techsolutions.com', '555-1113', 'DavidClark.png'),
(1, 'Emily', 'Stone', 26, 'UI/UX Designer', 52000, 'emily.stone@techsolutions.com', '555-1114', 'EmilyStone.png'),
(1, 'Ryan', 'Baker', 34, 'DevOps Engineer', 65000, 'ryan.baker@techsolutions.com', '555-1115', 'RyanBaker.png'),
(1, 'Grace', 'Lee', 29, 'Frontend Developer', 58000, 'grace.lee@techsolutions.com', '555-1116', 'GraceLee.png'),
(1, 'Mark', 'Davis', 38, 'CTO', 90000, 'mark.davis@techsolutions.com', '555-1117', 'MarkDavis.png'),
(1, 'Lia', 'Watson', 35, 'Research Analyst', 58000, 'lia.watson@techsolutions.com', '555-4444', 'LiaWatson.png');";

        using (var cmd = new SqlCommand(insertEmployees, connection))
        {
            cmd.ExecuteNonQuery();
        }
    }
}
```

ნახ. 5. ბაზაში სატესტო მონაცემების დამატება

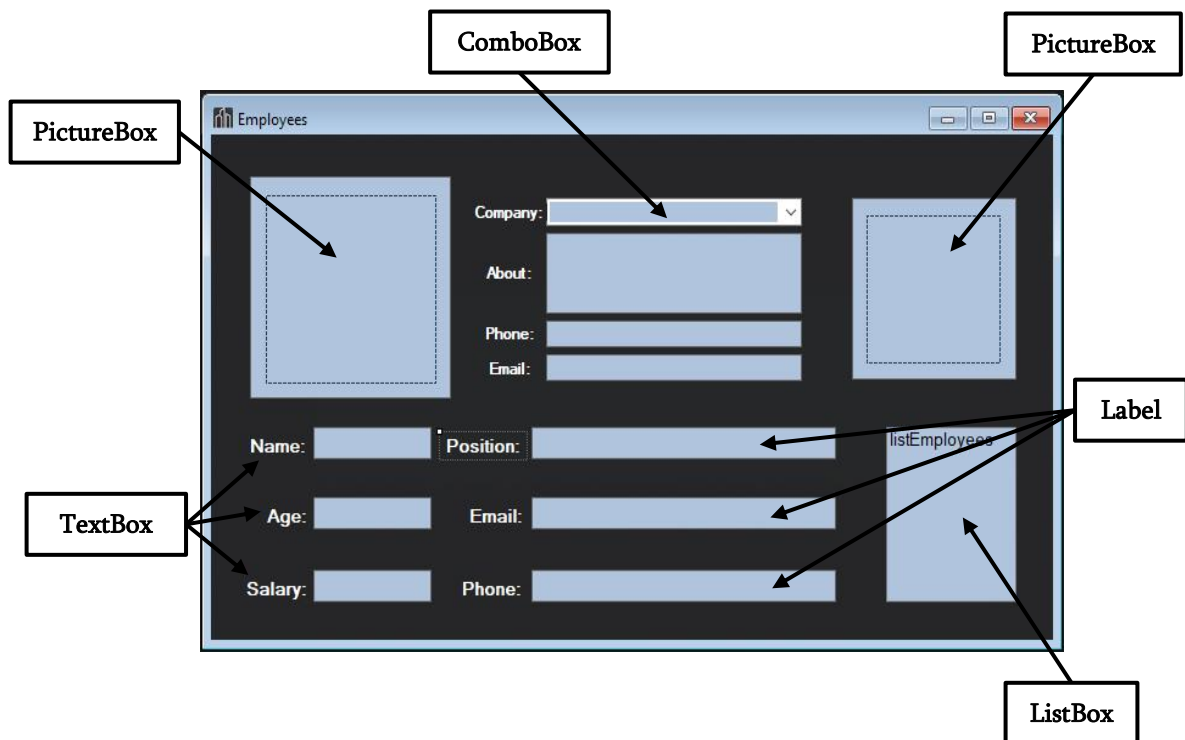
## 4.3. მომხმარებლის ინტერფეისის აგება

სამომხმარებლო ინტერფეისის ასაწყობად გამოვიყენე Form Designer, რომლის საშუალებითაც დავამატე ვიზუალური კომპონენტები მონაცემთა საჩვენებლად. მონაცემებისთვის შესაბამისი კომპონენტები წარმოდგენილია შემდეგ ჩამონათვალში:

1. TextBox: ბაზიდან მიღებული ძირითადი ტექსტური ინფორმაციის საჩვენებლად
2. Label: ტექსტური მონაცემებისთვის დასახელების ველების შესაქმნელად
3. ComboBox: კომპანიების ასარჩევად
4. ListBox: დასაქმებულების სიის შესაქმნელად



5. PictureBox: კომპანიების ლოგოებისა და დასაქმებულების სურათების საჩვენებლად



ნახ. 6. სამომხმარებლო ინტერფეისი აწყობა

#### 4.4. მონაცემთა ჩატვირთვა და დინამიკურად ჩვენება

კომპანიებისა და დასაქმებულების ცხრილების წაკითხვა ხდება DatabaseHelper სტატიკურ კლასში. ჯერ ხდება კომპანიათა სიისა და მონაცემთა ჩატვირთვა, შემდეგ კი - დასაქმებულების ცხრილებისა, რომლებიც დაკავშირებულია თითო ჩატვირთული კომპანიის ID გასაღებთან (იხ. ნახ. 9). როგორც არის წარმოდგენილი ნახ. 7, 8-ში, Form1.cs ფაილში ხდება ამ ინფორმაციების მიწოდება თითო კლასის ობიექტის შესაქმნელად.

```
companies = DatabaseHelper.GetAllCompanies();  
comboCompanies.DataSource = companies;
```

ნახ. 7. კომპანიების ობიექტების შექმნა

```
// Load employees  
var employees = DatabaseHelper.GetEmployeesByCompanyId(selectedCompany.Id);  
listEmployees.DataSource = employees;
```

ნახ. 8. დასაქმებულების ობიექტების შექმნა

```

public static List<Company> GetAllCompanies()
{
    List<Company> list = new List<Company>();
    using (var connection = new SQLiteConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT * FROM Companies";
        using (var command = new SQLiteCommand(query, connection))
        using (var reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                list.Add(new Company
                {
                    Id = Convert.ToInt32(reader["Id"]),
                    Name = reader["Name"].ToString(),
                    LogoPath = reader["LogoPath"].ToString(),
                    Description = reader["Description"].ToString(),
                    PhoneNumber = reader["PhoneNumber"].ToString(),
                    Email = reader["Email"].ToString()
                });
            }
        }
    }
    return list;
}

1 reference
public static List<Employee> GetEmployeesByCompanyId(int companyId)
{
    List<Employee> list = new List<Employee>();

    using (var connection = new SQLiteConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT * FROM Employees WHERE CompanyId = @CompanyId";

        using (var command = new SQLiteCommand(query, connection))
        {
            command.Parameters.AddWithValue("@CompanyId", companyId);

            using (var reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    list.Add(new Employee
                    {
                        Id = Convert.ToInt32(reader["Id"]),
                        CompanyId = Convert.ToInt32(reader["CompanyId"]),
                        FirstName = reader["FirstName"].ToString(),
                        LastName = reader["LastName"].ToString(),
                        Age = Convert.ToInt32(reader["Age"]),
                        Position = reader["Position"].ToString(),
                        Salary = Convert.ToDouble(reader["Salary"]),
                        Email = reader["Email"].ToString(),
                        PhoneNumber = reader["PhoneNumber"].ToString(),
                        PhotoPath = reader["PhotoPath"].ToString()
                    });
                }
            }
        }
    }
}

```

ნახ. 9. ბაზიდან ინფორმაციის წაკითხვა და ჩატვირთვა

#### 4.5. გამოსახულებების დამუშავება

გამოსახულებების დამუშავება გულისხმობს ლოგოებისა და სურათების წაკითხვას შესაბამისი მისამართებიდან, რაც ხდება System.Drawing ბიბლიოთეკის image მეთოდის გამოყენებით (იხ. ნახ. 10).

```
string fullPath = Path.Combine(Application.StartupPath, "images", selectedCompany.LogoPath);  
if (File.Exists(fullPath))  
{  
    pictureLogo.Image = Image.FromFile(fullPath);  
}  
else  
{  
    pictureLogo.Image = null; // or a placeholder  
}
```

ნახ. 10. ფოტოების წაკითხვა მითითებული მისამართის მიხედვით

#### 4.6. ობიექტზე ორიენტირებული პროგრამირების გამოყენება

კომპანიებისა და დასაქმებულების სიები შევქმენი ობიექტზე ორიენტირებული დაპროგრამების მიდგომით, რაც იმას ნიშნავს, რომ თითოეულის გავწერე კლასი შესაბამისი პარამეტრებით, რომელიც ემთხვევა ბაზაში შექმნილ ველებს (იხ. ნახ. 11).

```
namespace Empolyees.Models  
{  
    5 references  
    public class Employee  
    {  
        1 reference  
        public int Id { get; set; }  
        1 reference  
        public int CompanyId { get; set; }  
        2 references  
        public string FirstName { get; set; }  
        2 references  
        public string LastName { get; set; }  
        2 references  
        public int Age { get; set; }  
        2 references  
        public string Position { get; set; }  
        2 references  
        public double Salary { get; set; }  
        2 references  
        public string Email { get; set; }  
        2 references  
        public string PhoneNumber { get; set; }  
        2 references  
        public string PhotoPath { get; set; }  
  
        1 reference  
        public string FullName => $"{FirstName} {LastName}";  
    }  
}  
  
namespace Empolyees.Models  
{  
    6 references  
    public class Company  
    {  
        2 references  
        public int Id { get; set; }  
        1 reference  
        public string Name { get; set; }  
        2 references  
        public string LogoPath { get; set; }  
        2 references  
        public string Description { get; set; }  
        2 references  
        public string PhoneNumber { get; set; }  
        2 references  
        public string Email { get; set; }  
    }  
}
```

ნახ. 11. მარცხნივ - დასაქმებულების კლასი, მარჯვნივ - კომპანიების კლასი

## 5. პროგრამის რეალიზაციის შედეგი - კოდი

მონაცემთა ბაზიდან ინფორმაციების ამოღების შემდეგ გადაეცემა სამომხმარებლო ინტერფეისის შესაბამის კომპონენტს (იხ. ნახ. 12) და მომხმარებელს აძლევს შესაძლებლობას დაინახოს სასურველი ინფორმაცია. გამშვები კოდი კი დაწერილია Program.cs ფაილში, რომლის კოდიც მოცემულია ნახ. 13-ში.

```
private void comboCompanies_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboCompanies.SelectedItem is Company selectedCompany)
    {
        // Show company info
        txtDescription.Text = selectedCompany.Description;
        txtPhone.Text = selectedCompany.PhoneNumber;
        txtEmail.Text = selectedCompany.Email;

        string fullPath = Path.Combine(Application.StartupPath, "images", selectedCompany.LogoPath);

        if (File.Exists(fullPath))
        {
            pictureLogo.Image = Image.FromFile(fullPath);
        }
        else
        {
            pictureLogo.Image = null; // or a placeholder
        }

        // Load employees
        var employees = DatabaseHelper.GetEmployeesByCompanyId(selectedCompany.Id);
        listEmployees.DataSource = employees;
        listEmployees.DisplayMember = "FullName";
    }
}

1 reference
private void listEmployees_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listEmployees.SelectedItem is Employee emp)
    {
        txtName.Text = emp.FullName;
        txtAge.Text = emp.Age.ToString();
        txtPosition.Text = emp.Position;
        txtSalary.Text = emp.Salary.ToString("F2");
        txtEmpEmail.Text = emp.Email;
        txtEmpPhone.Text = emp.PhoneNumber;

        string fullPath = Path.Combine(Application.StartupPath, "images", emp.PhotoPath);

        if (File.Exists(fullPath))
        {
            picturePhoto.Image = Image.FromFile(fullPath);
        }
        else
        {
            picturePhoto.Image = null; // or a placeholder
        }
    }
}
```

ნახ. 12. ინტერფეისისთვის ინფორმაციის გადაცემის კოდი

```

using System;
using System.Windows.Forms;
using YourProjectNamespace;

namespace Empolyees
{
    0 references
    internal static class Program
    {
        [STAThread]
        0 references
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            DatabaseHelper.InitializeDatabase();
            DatabaseHelper.SeedTestData();
            Application.Run(new Form1());
        }
    }
}

```

ნახ. 13. აპლიკაციის გაშვების კოდი

აპლიკაციის მარტივად გაშვებისთვის შექმნილია .exe ფორმატის ფაილი, რომლის ჩართვისას ნახ. 14-ის შესაბამისად ეკრანზე გამოჩნდება სამომხმარებლო ინტერფეისი, სადაც მომხმარებელს შეუძლია დაინახოს ბაზაში შენახული კომპანიებისა და მათი თანამშრომლების შესახებ არსებული ინფორმაციები, როგორიცაა სახელი, ასაკი, ხელფასი, საკონტაქტო ინფორმაცია, პოზიცია და ა.შ.

The screenshot shows a window titled "Employees" with a dark background. On the left is a profile picture of a man. To the right of the picture, there are several input fields for company information: "Company:" (Tech Solutions Ltd.), "About:" (IT and software services provider.), "Phone:" (123-456-7890), and "Email:" (info@techsolutions.com). On the far right is a logo for "Tech Solutions LTD". Below these fields, there are more input fields for employee information: "Name:" (Ali Rahimi), "Position:" (AI/ML Engineer), "Age:" (22), "Email:" (ali.rahimi@techsolutions.com), "Salary:" (200000.00), and "Phone:" (579-2431). On the right side of the employee information section, there is a dropdown menu showing a list of names: Ali Rahimi, David Clark, Emily Stone, Grace Lee, John Doe, Lia Watson, Mark Davis, and Ryan Baker. The "Ali Rahimi" option is currently selected.

ნახ. 14. Tech Solutions Ltd კომპანიისა და მისი ერთ-ერთი დასაქმებულის ინფორმაციები

## 6. დასკვნა

ამ პროექტის ფარგლებში წარმატებით შემუშავდა .NET Windows Forms აპლიკაცია, რომელიც უზრუნველყოფს სხვადასხვა კომპანიის თანამშრომლების შესახებ ინფორმაციის კომპაქტურად და ინტუიციურად ჩვენებას. პროექტმა აჩვენა, თუ როგორ შეიძლება დესკტოპ აპლიკაციის საშუალებით ობიექტურად სტრუქტურირებული მონაცემების მართვა და წარმოდგენა მომხმარებლისთვის მარტივი და გასაგები ფორმით.

აპლიკაცია იყენებს C# ენას და SQLite მონაცემთა ბაზას, რაც უზრუნველყოფს მონაცემებთან სწრაფ და ეფექტურ წვდომას, გარეშე დამატებითი სერვერის საჭიროებისა. ინტერფეისი შექმნილია ისე, რომ მომხმარებელმა შეძლოს კომპანიებისა და თანამშრომლების დეტალური ინფორმაციის დათვალიერება კომფორტულად.

პროექტის ფარგლებში მიღწეულმა შედეგებმა აჩვენა:

- ობიექტზე ორიენტირებული მიდგომის ეფექტიანობა მსგავსი აპლიკაციების განვითარებაში;
- მონაცემთა ინიციალიზაციის, შენახვისა და ვიზუალიზაციის ინტეგრირებული მუშაობის მნიშვნელობა;
- ის, თუ როგორ შეიძლება ტექნოლოგიური გადაწყვეტა გახდეს ორგანიზაციული პროცესების გამარტივების ხელსაწყო.

მომავალში აპლიკაცია შეიძლება გაუმჯობესდეს დამატებითი ფუნქციონალებით, როგორიცაა მონაცემების რედაქტირება, ძიების მექანიზმი, ან ფილტრაცია. ეს პროექტი წარმოადგენს კარგ მაგალითს საწყისი დონის მონაცემებზე დაფუძნებული ბიზნეს აპლიკაციის შექმნისა, რომელსაც შეუძლია ეფექტურად მოემსახუროს რეალურ საჭიროებებს.


## 7. გამოყენებული ლიტერატურა

1. სურგულაძე გ., პეტრიაშვილი ლ. ვიზუალური დაპროგრამება C# ენის ბაზაზე ინფორმაციული სისტემებისათვის (Ms Visual Studio.NET 2019 პლატფორმაზე). თბილისი: სტუ-ს „IT კონსალტინგის ცენტრი“. 2019.
2. სამხარაძე რ., გაჩეჩილაძე ლ. პროგრამირება Windows-თვის (C# ენის ბაზაზე). თბილისი: სტუ-ს "IT-კონსალტინგის სამეცნირო ცენტრი". 2019.

## 8. დანართები

დანართი 1: კომპანიებისა და დასაქმებულების რამდენიმე მაგალითი

Employees




Company: **HealthPlus Inc.**

About: Healthcare and medical services.

Phone: 456-789-0123

Email: support@healthplus.com



Name: **Daniel White**

Age: **45**

Salary: **110000.00**

Position: **Chief Medical Officer**

Email: **daniel.white@healthplus.com**

Phone: **555-3336**

Daniel White

Ella Wilson

Lily Young


Lucas Taylor

Mason Davis

Mia Anderson

Sophia Brown

Employees




Company: **FinSecure LLC**

About: Financial consulting and auditing.

Phone: 321-654-0987

Email: info@finsecure.com



Name: **Henry Wood**

Age: **36**

Salary: **90000.00**

Position: **Investment Advisor**

Email: **henry.wood@finsecure.com**

Phone: **555-4447**

Amelia Bennett

Benjamin Hughes

Charlotte Green


Henry Wood

Isabella Scott

James King

Michael Adams

Employees




Company: **Green Earth Corp**

About: Environmental engineering company.

Phone: 987-654-3210

Email: contact@greenearth.com



Name: **Ethan Clark**

Age: **40**

Salary: **71000.00**

Position: **Environmental Scientist**

Email: **ethan.clark@greenearth.com**

Phone: **555-2224**

Ava Lee

Chloe Wang

Emily Nguyen

Ethan Clark


Liam Johnson

Noah Moore

Olivia Martin



## დანართი 2: SQLite მონაცემთა ბაზის ვიზუალიზაცია


**SQLite Viewer**  
view sqlite file online

Drop file here to load content or click on this box to open file dialog.

Companies (4 rows) Export

SELECT \* FROM 'Companies' LIMIT 0,30
Execute

ID	Name	LogoPath	Description	PhoneNumber	Email
1	Tech Solutions Ltd.	TechSolutionsLtd.png	IT and software services provider.	123-456-7890	info@techsolutions.com
2	Green Earth Corp	GreenEarthCorp.png	Environmental engineering company.	987-654-3210	contact@greeneearth.com
3	HealthPlus Inc.	HealthPlusInc.png	Healthcare and medical services.	456-789-0123	support@healthplus.com
4	FinSecure LLC	FinSecureLLC.png	Financial consulting and auditing.	321-654-0987	info@finsecure.com

Employees (30 rows) Export

SELECT \* FROM 'Employees' LIMIT 0,20
Execute

ID	CompanyId	FirstName	LastName	Age	Position	Salary	Email	PhoneNumber	PhotoPath
1	1	Ali	Rahimi	22	AI/ML Engineer	200000	ali.rahimi@techsolutions.com	579-2431	AliRahimi.png
2	1	John	Doe	28	Software Engineer	60000	john.doe@techsolutions.com	555-1111	JohnDoe.png
3	1	Sarah	Mills	32	Project Manager	72000	sarah.mills@techsolutions.com	555-1112	SarahMills.png
4	1	David	Clark	30	QA Analyst	50000	david.clark@techsolutions.com	555-1113	DavidClark.png
5	1	Emily	Stone	26	UI/UX Designer	52000	emily.stone@techsolutions.com	555-1114	EmilyStone.png
6	1	Ryan	Baker	34	DevOps Engineer	65000	ryan.baker@techsolutions.com	555-1115	RyanBaker.png
7	1	Grace	Lee	29	Frontend Developer	58000	grace.lee@techsolutions.com	555-1116	GraceLee.png
8	1	Mark	Davis	38	CTO	90000	mark.davis@techsolutions.com	555-1117	MarkDavis.png
9	1	Lia	Watson	35	Research Analyst	58000	lia.watson@techsolutions.com	555-4444	LiaWatson.png
10	2	Emily	Nguyen	30	Environmental Engineer	60000	emily.nguyen@greeneearth.com	555-2221	EmilyNguyen.png
11	2	Liam	Johnson	35	Research Analyst	62000	liam.johnson@greeneearth.com	555-2222	LiamJohnson.png
12	2	Olivia	Martin	28	Field Technician	47000	olivia.martin@greeneearth.com	555-2223	OliviaMartin.png
13	2	Ethan	Clark	40	Environmental Scientist	71000	ethan.clark@greeneearth.com	555-2224	EthanClark.png
14	2	Ava	Lee	27	GIS Specialist	53000	ava.lee@greeneearth.com	555-2225	AvaLee.png
15	2	Noah	Moore	31	Sustainability Consultant	64000	noah.moore@greeneearth.com	555-2226	NoahMoore.png
16	2	Chloe	Wang	33	Operations Manager	67000	chloe.wang@greeneearth.com	555-2227	ChloeWang.png
17	3	Sophia	Brown	36	Registered Nurse	58000	sophia.brown@healthplus.com	555-3331	SophiaBrown.png
18	3	Mason	Davis	42	General Practitioner	95000	mason.davis@healthplus.com	555-3332	MasonDavis.png
19	3	Ella	Wilson	29	Lab Technician	52000	ella.wilson@healthplus.com	555-3333	EllaWilson.png
20	3	Lucas	Taylor	31	Medical Assistant	48000	lucas.taylor@healthplus.com	555-3334	LucasTaylor.png

დანართი 3: დასაქმებულთა სატესტო ფოტოებისთვის გამოყენებული Dall-E 3 მოდელის მიერ დაგენერირებული ილუსტრაცია

