

BUDGET MANAGEMENT APP

ΜΑΘΗΜΑ : ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΦΟΙΤΗΤΕΣ : ΝΤΕΝΙΣΟΝ ΜΠΑΡΔΗ, ΑΓΓΕΛΟΣ ΠΕΤΡΑΚΙΔΗΣ

Τι αφορά το project και ποιο πρόβλημα λύνει;

Το παρόν project είναι μια full stack εφαρμογή υπολογισμού εξόδων (Budget Management App) η οποία επιτρέπει στους χρήστες να καταγράφουν και να παρακολουθούν τα έσοδα και τα έξοδά τους. Η εφαρμογή υποστηρίζει πολλαπλούς χρήστες, σύνδεση με βάση δεδομένων, και παρέχει μια απλή και φιλική προς τον χρήστη διεπαφή για καλύτερο οικονομικό έλεγχο.

Γιατί επιλέξαμε αυτό το αντικείμενο;

Επιλέξαμε αυτό το αντικείμενο γιατί είναι πρακτικό και αφορά την καθημερινή οικονομική διαχείριση.

Ποιοι είναι οι βασικοί στόχοι και ποιες δυνατότητες περιλαμβάνονται;

Η δημιουργία μιας λειτουργικής εφαρμογής για την καταγραφή και παρακαλούθηση εξόδων και εσόδων στην οποία περιλαμβάνονται δυνατότητες αποθήκευσης στοιχείων ενός χρήστη εξυπηρετώντας έτσι την ιδιωτικότητά του. Επίσης παρουσιάζει το συνολικό υπόλοιπο του χρήστη και δίνει τέλος την δυνατότητα εκκαθάρισης δεδομένων.

Πώς λειτουργεί το σύστημα;

Ο χρήστης κάνει login ή εγγραφή.

Εισάγει budget/expenses.

Τα δεδομένα αποθηκεύονται και υπολογίζεται το υπόλοιπο.

Ποιες είναι οι βασικές του συνιστώσες;

- User Authentication

LoginForm: Σύνδεση χρηστών (SQL queries).

RegisterForm: Εγγραφή νέων χρηστών.

- GUI & Business Logic

MainForm:

TextBoxes (εισαγωγή budget/εξόδων).

Buttons (υποβολή, εκκαθάριση).

Labels (εμφάνιση υπολοίπου).

- Data Management

SQL DATABASE: Αποθήκευση δεδομένων χρηστών(πίνακας users)

- Validation

Έλεγχοι:

TryParse για αριθμούς.

Έλεγχος υπολοίπου πριν την αφαίρεση.

Τεχνολογίες που Χρησιμοποιήθηκαν:

Γλώσσα : C++

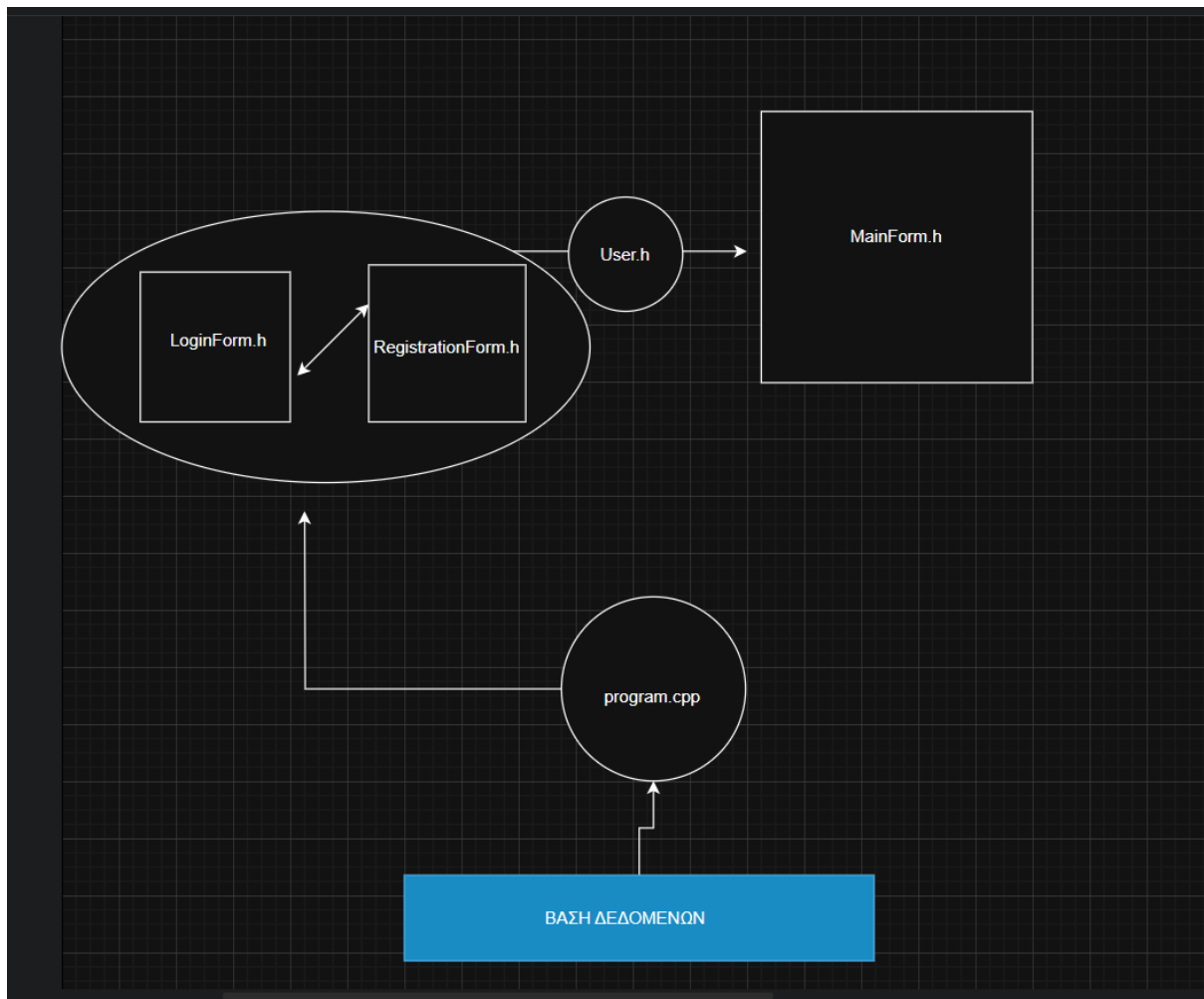
Περιβάλλον : Visual Studio 2022

GUI : Windows Forms

Βάση Δεδομένων : SQL SERVER MANAGEMENT STUDIO

Includes : Pragma Once

Διάγραμμα Ροής



Repository Link: <https://github.cub/trovaotrova/ergakekatos>

Αποτέλεσμα και Demo:

Login

Email

Password

OK

cancel

[register](#)

REGISTER

NAME:

EMAIL:

PHONE:

ADD:

PWD:

REGISTER

cancel

[login](#)

MainForm

Welcome, Jack!

Enter Your Budget:

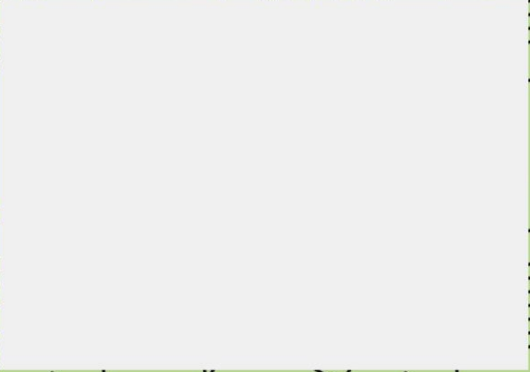
Expense

Expense-Amount

add budget

add expense

CLEAR



MainForm

Welcome, Jack!

Enter Your Budget:

Budget: 900,00 €

Expense

Expense-Amount

add budget

add expense

CLEAR

Your Expenses

laptop: -100,00 €

MainForm

Welcome, Jack!

Enter Your Budget: 1000

Budget: 855,00 €

Expense

Expense-Amount

add budget

add expense

CLEAR

Your Expenses

laptop: -100,00 €

telephone: -45,00 €

Κώδικας και Υλοποίηση

Το πρόγραμμα απαρτίζεται από αρχεία .cpp και .h, η σύνδεση με τη βάση υλοποιήθηκε εύκολα από το εξέχον γνώρισμα που παρέχει το περιβάλλον visual studio 2022, server explorer το οποίο συνδέεται αναζητώντας το server name ενώ η διαμόρφωση της βάσης (δημιουργία πινάκων) γίνεται από το εργαλείο Microsoft SQL Server Management Studio στο οποίο προσθέσαμε database name (db.myapp) πίνακα Users με Columns id name email phone password.

Το αρχείο ξεκινάει ανοίγοντας το LoginForm και πατώντας το button ok για να προχωρήσεις στο MainForm, στην λειτουργία του κουμπιού επιτυγχάνεται μια επικοινωνία με τη βάση.....

```

}
private: System::Void expensebtn_Click(System::Object^ sender, System::EventArgs^ e) {
    String^ expenseName = tbexpense->Text;
    String^ amountText = tbAmount->Text;

    double expenseAmount;
    if (String::IsNullOrEmpty(expenseName)) {
        MessageBox::Show("Please enter a valid expense name and amount.");
        return;
    }

    if (expenseAmount > currentBalance) {
        MessageBox::Show("Not enough balance.");
        return;
    }
    if (Double::TryParse(amountText, expenseAmount)) {
        currentBalance -= expenseAmount;

        expenseList += expenseName + ": -" + expenseAmount.ToString("F2") + " €\n";
        EXPENSES->Text = "Your Expenses\n\n" + expenseList;
        balance->Text = "Budget: " + currentBalance.ToString("F2") + " €";
    }

    tbexpense->Clear();
    tbAmount->Clear();
}

```

Μέσω διαχείρισης εξαιρέσεων try catch ελέγχεται αν τα στοιχεία που καταχώρησε ο χρήστης αντιστοιχούν σε κάποιο αποθηκευμένο προϋπάρχων χρήστη στη βάση και αν ναι τότε προχωράει στο MainForm.


```

private: System::Void btnOk_Click(System::Object^ sender, System::EventArgs^ e) {
    String^ email = this->tbEmail->Text;
    String^ password = this->tbPassword->Text;

    if (email->Length == 0 || password->Length == 0) {
        MessageBox::Show("please enter email and password", "", MessageBoxButtons::OK);
        return;
    }
    try {
        String^ connString = "Data Source=localhost\\MSSQLSERVER01;Initial Catalog=myapp;Integrated Security=True";
        SqlConnection sqlConn(connString);
        sqlConn.Open();
        String^ sqlQuery = "SELECT * FROM users WHERE email=@email AND password=@pwd";
        SqlCommand command(sqlQuery, % sqlConn);
        command.Parameters->AddWithValue("@email", email);
        command.Parameters->AddWithValue("@pwd", password);

        SqlDataReader^ reader = command.ExecuteReader();
        if (reader->Read()) {
            user = gcnew User;
            user->id = reader->GetInt32(0);
            user->name = reader->GetString(1);
            user->email = reader->GetString(2);
            user->phone = reader->GetString(3);
            user->address = reader->GetString(4);
            user->password = reader->GetString(5);
            this->Close();
        }
        else {
            MessageBox::Show("Email or Password is incorrect", "Email or Password Error", MessageBoxButtons::OK);
        }
    }
    catch (Exception^ ex) {
        MessageBox::Show("Failed to connect to database:\n" + ex->Message, "Database connection error", MessageBoxButtons::OK);
    }
}

public: bool switchToRegister = false;
private: System::Void Register_LinkClicked(System::Object^ sender, System::Windows::Forms::LinkLabelLinkClickedEventArgs^ e) {
    this->switchToRegister = true;
    this->Close();
};
};
}

```

Έπειτα ο χρήστης έχει το δικαίωμα να δώσει τιμές και είδη εξόδων. Αυτές οι πληροφορίες αποθηκεύονται. Για παράδειγμα με το πάτημα του κουμπιού `expensebtn` στο `string expenselist` που είναι property της κλάσης `MainForm`, προσθέτεται το καταχωρημένο `value`, εφόσον το ποσό πληρεί τις απαραίτητες αριθμητικές προϋποθέσεις.

Σύγκριση κώδικα με AI Generated Code

```
cpp                                                                    Αντιγραφή  Επιβεργασία

private: System::Void expensebtn_Click(System::Object^ sender, System::EventArgs^ e) {
    String^ expenseName = tbexpense->Text->Trim();
    String^ amountText = tbAmount->Text->Trim();

    double expenseAmount;

    // Έλεγχος αν το όνομα εξόδου ή το ποσό είναι κενά
    if (String::IsNullOrEmpty(expenseName) || String::IsNullOrEmpty(amountText)) {
        MessageBox::Show("Please enter both a valid expense name and amount.");
        return;
    }

    // Έλεγχος αν το ποσό είναι έγκυρος αριθμός
    if (!Double::TryParse(amountText, expenseAmount)) {
        MessageBox::Show("Please enter a valid number for the amount.");
        return;
    }

    // Έλεγχος αν υπάρχει αρκετό υπόλοιπο
    if (expenseAmount > currentBalance) {
        MessageBox::Show("Not enough balance to add this expense.");
        return;
    }

    // Αφαίρεση του ποσού και ενημέρωση της λίστας εξόδων και του υπολοίπου
    currentBalance -= expenseAmount;
    expenseList += expenseName + ": -" + expenseAmount.ToString("F2") + " €\n";
    EXPENSES->Text = "Your Expenses\n\n" + expenseList;
    balance->Text = "Budget: " + currentBalance.ToString("F2") + " €";

    // Καθαρισμός πεδίων
    tbexpense->Clear();
    tbAmount->Clear();
}
```

```

}
private: System::Void expensebtn_Click(System::Object^ sender, System::EventArgs^ e) {
    String^ expenseName = tbexpense->Text;
    String^ amountText = tbAmount->Text;

    double expenseAmount;
    if (String::IsNullOrEmpty(expenseName)) {
        MessageBox::Show("Please enter a valid expense name and amount.");
        return;
    }

    if (expenseAmount > currentBalance) {
        MessageBox::Show("Not enough balance.");
        return;
    }
    if (Double::TryParse(amountText, expenseAmount)) {
        currentBalance -= expenseAmount;

        expenseList += expenseName + ": -" + expenseAmount.ToString("F2") + " €\n";
        EXPENSES->Text = "Your Expenses\n\n" + expenseList;
        balance->Text = "Budget: " + currentBalance.ToString("F2") + " €";
    }

    tbexpense->Clear();
    tbAmount->Clear();
}

```

Οι αλλαγές που παρατηρούνται στο κώδικα τεχνητής νοημοσύνης είναι εμφανείς. Υπάρχει η συνάρτηση Trim() που αφαιρεί κενά στην αρχή και στο τέλος των πεδίων. Πιο ξεκάθαρα μηνύματα σφάλματος. Καλύτερη σειρά ελέγχων. Έλεγχος ποσού πριν από τη χρήση του.

Conclusions & Lessons Learned

Από αυτό το project εξοικειωθήκαμε με τη γλώσσα C++ , με το περιβάλλον ανάπτυξης κώδικα Visual Studio όπως και με το εργαλείο Microsoft SQL Server.

Θα μπορούσε να βελτιωθεί η δομή του κώδικα και η διεπαφή της εφαρμογής.

Μελλοντικοί προέκταση του Project πιθανόν να είναι η μόνιμη αποθήκευση των εισόδων των χρηστών ακόμα και μετά τον τερματισμό του

προγράμματος καθώς και η αυτόματη αποθήκευση της ημερομηνίας και ώρας που γίνονται οι καταχωρήσεις εξόδων.