**עסק ללימוד שחיה**

**שם העסק: SWIM4LIFE**

**מגישה: בר סימון**

**תיכון מקיף יא אשדוד**

**תוכן עניינים:**

**שער..................................................................................................1**

**תוכן עניינים.......................................................................................2**

**נושא ורקע הפרוייקט...........................................................................3**

**תהליכים מרכזיים בתוכנה....................................................................4**

**תרשימים...........................................................................................5**

**מבנה DB...........................................................................................7**

**קישורים...........................................................................................10**

**הסבר על מסכי התוכנה.....................................................................13**

**קוד.................................................................................................18**

**נושא הפרוייקט:**

עסק ללימוד ואימון שחייה.

**רקע על הפרוייקט:**

הפרוייקט הינו בנושא עסק ללימוד שחיה. העסק הוא עסק שאבי מנהל ולכן בחרתי לבנות תוכנה לעסק זה, על מנת לייעל ולשפר את התנהלות העסק. העסק פועל כבר שנתיים בכל רחבי הארץ, ישנם מספר מאמנים המאמנים רבוצות שונות ברחבי הארץ בכל הגילאים. ישנם שיעורים בבריכות במקומות שונים כאשר אלו יכולים להיות שיעורים פרטיים, קבוצות קטנות, שיעורי "אחד על אחד" וגם קבוצות גדולות.

בנוסף התחברתי לנושא התוכנה שאותה בניתי מכיוון שאני מאוד אוהבת את התחום הזה ומאוד מתחברת לתחום השחייה והשמירה על הכושר הגופני.

**מהות התוכנה:**

התכנה נכתבה במטרה לייעל את דרכי ההתנהלות של העסק. עסק זה מתנהלת ברובו ע"י קביעת מפגשים, ארגון קבוצות, מינוי מדריכים ואימונים באופן ידני ביומן. בנוסף לכך, הוצאות העסק השוטפות מתנהלות גם הן באופן ידני המקשה מאוד על יעילות והצלחת העסק, הדבר לא מאפשר מעקב פשוט ונוח אחר היסטוריית האימונים, הסדנאות, ההוצאות של העסק וכן מעקב פשוט ונוח אחרי הפעילות השוטפת. לכן, החלטתי לבנות עבור העסק תוכנה שתועיל לפישוט ונוחות פעולות אלו ותהפוך את ניהול העסק ומעקב אחרי הנעשה בו לפשוטה ומסודרת יותר, משום שכל המידע יהיה מרוכז בבסיס הנתונים ופעולות הגישה אליו הן קצרות ונעשות ידנית ע"י התכנה ומציגות בפשטות את הנתונים למתפעל התכנה ללא צורך בהבנה עמוקה של התוכנה. התוכנה יעילה וידידותית מאוד למשתמש.

**תהליכים מרכזיים בתוכנה:**

**ניהול תלמידים וקבוצות:**

מנהל העסק יכול לדעת כל פרט על קבוצות השחייה השונות ועל המשתתפים בהם ולעקוב אחרי פעילות הקבוצות.

**ניהול עובדים:**

בעזרת תהליך זה מנהל העסק יכול לראות את הפרטים המלאים על כל עובד ולהכיר לעומק את האנשים אותם הוא מעסיק, באופן זה יוכל המנהל לדעת מה טוב בעבור העסק.

**פעילויות העשרה ופרטי בריכות:**

התוכנה מאפשרת מעקב אחרי סדנאות העשרה לעובדים ומעקב אחרי פרטי הבריכות בהן מתאמנות הקבוצות השונות.

**הוצאות:**

בעזרת פירוט ההוצאות השוטף הכולל את תאריך ההוצאה ואת פרטי ההוצאה, יכול מנהל העסק לעקוב באופן יעיל אחרי המאזן הכלכלי בעסק.

**ניהול לוח זמנים שוטף:**

מנהל העסק יכול בקלות וביעילות לעקוב אחרי האימונים השונים שנקבעו ואחרי המשתתפים בהם.

**תרשימי זרימה :**

עמוד ראשי

ניהול לו"ז

ניהול לו"ז

ניהול לו"ז

ניהול לו"ז

ניהול לו"ז

לו"ז

קבוצות

סדנאות לעובדים

הוצאות

תלמידים

עובדים

בריכות

מוצרים

הוספת הוצאות

עדכון הוצאות

חישוב הוצאות לפי כמות מוצר

מעקב אחר הוצאות

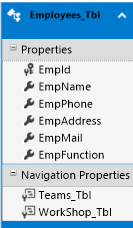
הוצאות

צפייה ברשימת השיעורים

קביעת שיעורים

ביטול שיעורים

עריכת פרטי שיעור

**מבנה הDB:**

שיוך קבוצה ועובד לבריכה

הוספת בריכה ועדכון פרטיה

שיוך עובד לסדנא

הוספת סדנאות חדשות ועדכון פרטיהן

סדנאות העשרה לעובדים

בריכות

שיוך העובד לקבוצה

עדכון פרטי עובד

מחיקת עובד

הוספת עובד

ניהול מערך עובדים

עדכון פרטי תלמיד \ קבוצה

הוספת תלמיד \ קבוצה

מחיקת תלמיד \ קבוצה

מעקב התקדמות

ניהול תלמיד \ קבוצה

**עובדים - Employees:**

**EmpId** – מספר סידורי של עובד (נקבע אוטומטית)

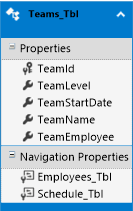
EmpName- שם עובד

EmpPhone- מספר טלפון של עובד

EmpAddress- כתובת מגורי העובד

EmpMail- כתובת מייל של עובד

EmpFunction- תפקיד העובד

**קבוצות – Teams:**

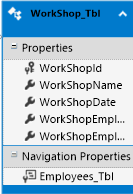
**TeamId**- מספר סידורי של קבוצה (נקבע אוטומטית)

TeamLevel- רמת הקבוצה

TeamStartDate- תאריך ההתחלה של הקבוצה

TeamName- שם הקבוצה

TeamEmployee- מאמן הקבוצה

**סדנאות הדרכה לעובדים – WorkShop:**

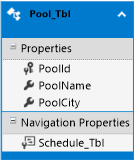
**WorkShopId-** מספר סידורי של הסדנא (נקבע אוטומטית)

WorkShopName- שם הסדנא

WorkShopDate- תאריך הסדנא

WorkShopEmployeeName- שם העובד המשתתף בסדנא

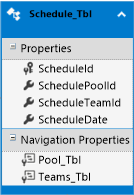
WorkShopEmployeeName- מספר סידורי של העובד המשתתף בסדנא

**בריכות – Pool:**

**PoolId**- מספר סידורי של בריכה (נקבע אוטומטית)

PoolName- שם הבריכה

PoolCity- עיר בה הבריכה נמצאת

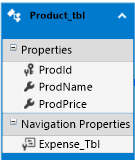
**לוח זמנים – Schedule:**

**ScheduleId**- מספר סידורי של המפגש (נקבע אוטומטית)

SchedulePoolId- מספר סידורי של הבריכה בה מתקיים המפגש

ScheduleTeamId- מספר סידורי של הקבוצה המשתתפת במפגש

ScheduleDate- תאריך המפגש

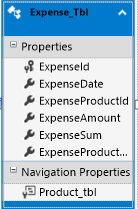


**מוצרים – Product:**

**ProdId**- מספר סידורי של המוצר (נקבע אוטומטית)

ProdName- שם המוצר

ProdPrice- מחיר המוצר

 **הוצאות – Expense:**

**ExpenseId**- מספר סידורי של ההוצאה (נקבע אוטומטית)

ExpenseDate- תאריך הוצאה

ExpenseProductId- המספר הסידורי של המוצר שנקנה בהוצאה זו

ExpenseAmount- כמות היחידות של המוצר

ExpenseSum- סכום ההוצאה הכולל

ExpenseProductName- שם המוצר שנקנה

**חניכים – Students:**

**StudentId**- מספר סידורי של תלמיד (נקבע אוטומטית)

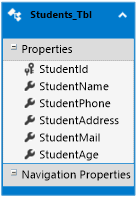
StudentName- שם התלמיד

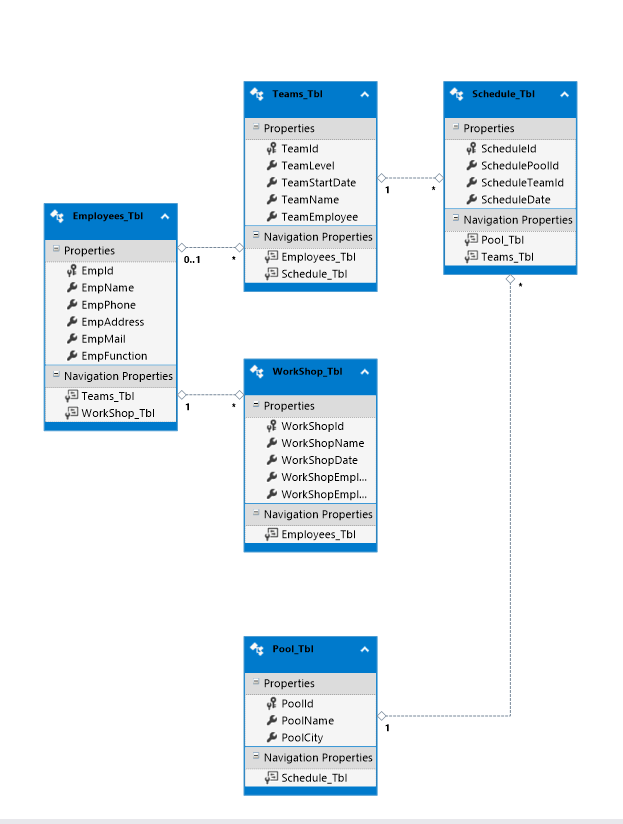
StudentPhone- מספר טלפון של החניך

StudentAddress- כתובת מגורי החניך

StudentMail- כתובת המייל של החניך

StudentAge- גיל החניך

 **הסבר על הקישורים:**



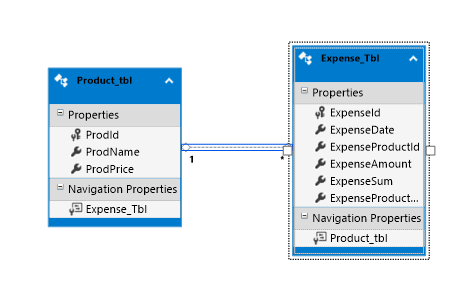
**טבלת העובדים**- מקושרת אל טבלת הקבוצות וטבלת הסדנאות. בטבלת הקבוצות יופיע שדה המאפשר לבחור מי הוא העובד האחראי על קבוצה מסויימת ובטבלת הסדנאות לעובדים יופיע שדה המאפשר לבחור איזה עובד לוקח חלק בסדנא מתוך טבלת העובדים. זהו בעצם הקשר בין הטבלאות.

**טבלת הבריכות**- מקושרת אל טבלת לוח הזמנים, בטבלת לוח הזמנים ניתן לבחור מתוך טבלת הבריכות באיזו בריכה יתקיים מפגש.

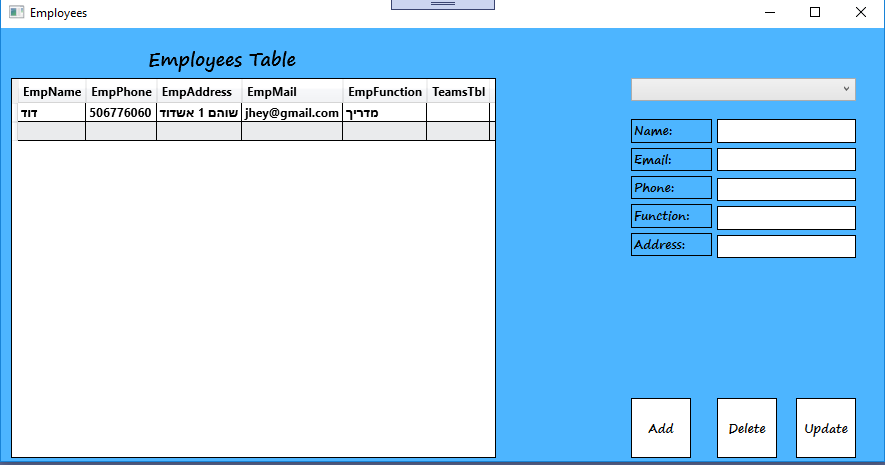
**טבלת הקבוצות**- מקושרת אל טבלת לוח הזמנים, בטבלת לוח הזמנים ניתן לבחור מתוך טבלת הקבוצות איזו קבוצה משתתפת במפגש. טבלה זו בנוסף מתקשרת לטבלת העובדים .

**טבלת לוח הזמנים** – מתקשרת לטבלת הבריכות ולטבלת הקבוצות.

**טבלת הסדנאות לעובדים** - מתקשרת לטבלת העובדים.



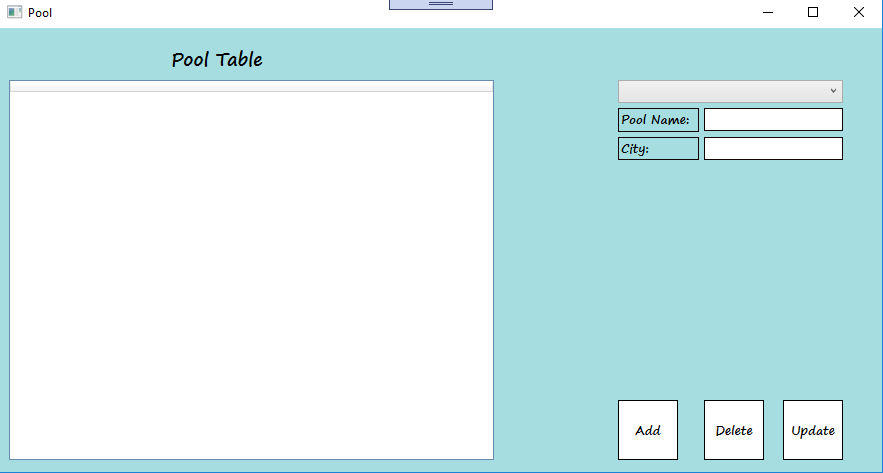
טבלת המוצרים וטבלת ההוצאות מקושרות אחת לשניה, בטבלת ההוצאות סכום כל הוצאה מחושב על ידי מוצר כלשהו מתוך טבלת המוצרים וע"י הכמות בה נקנה המוצר. יש לבחור מתוך טבלת ההוצאות את שם המוצר אותו רוצים לקנות בהוצאה מסויימת וכך יחושב סכום ההוצאה.

**הסבר על מסכי התוכנה השונים:**

\*במסך זה ניתן להוסיף פרטי עובד ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את שם העובד ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".



\*במסך זה ניתן להוסיף פרטי בריכה ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

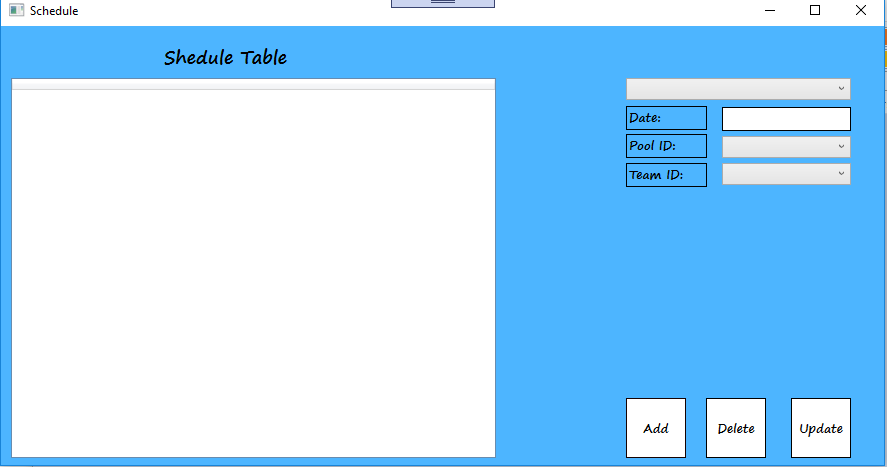
בכדי לעדכן שורה בטבלה יש לבחור בתיבה את שם הבריכה ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".

\*במסך זה ניתן להוסיף פרטי מוצר ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את שם המוצר ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

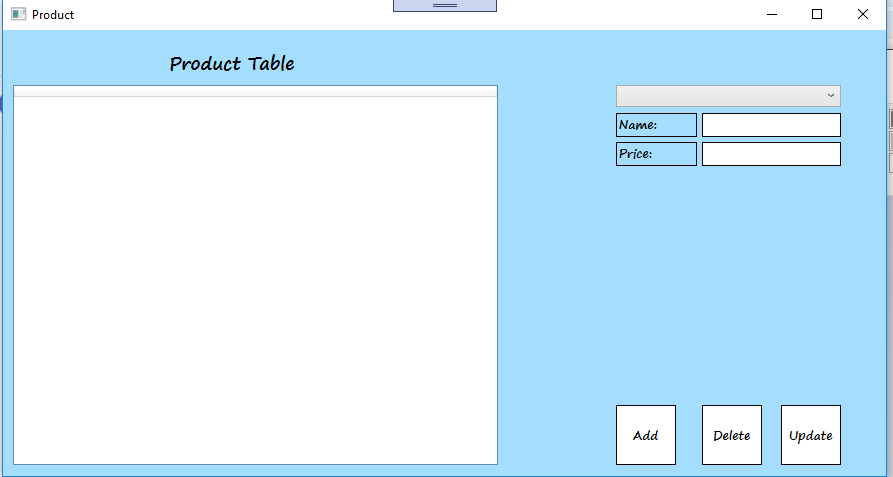
בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".

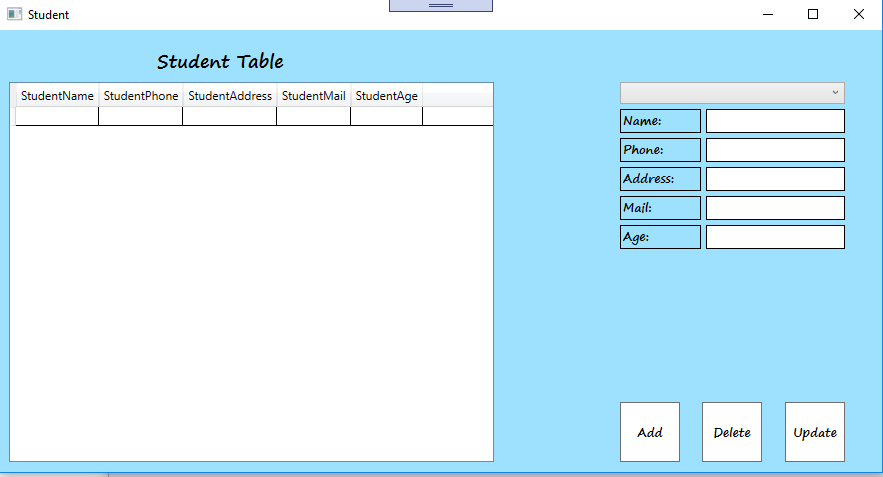


\*במסך זה ניתן להוסיף פרטי לו"ז ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את התאריך המתאים ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".

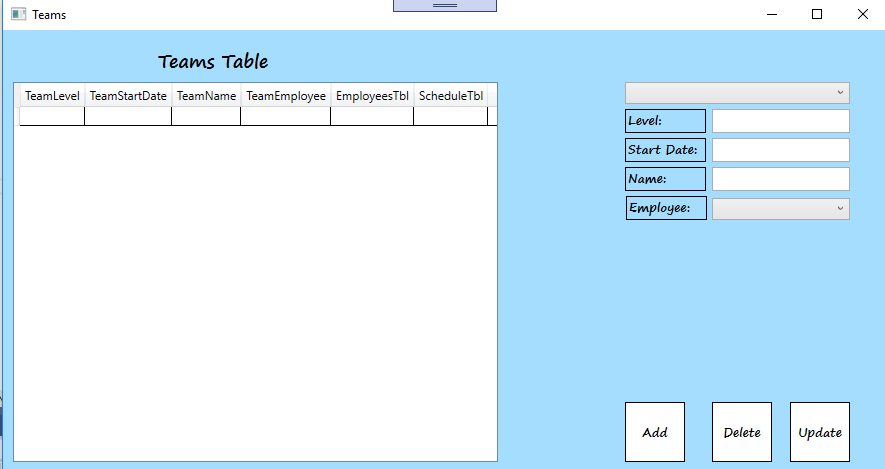




\*במסך זה ניתן להוסיף פרטי תלמיד ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את שם התלמיד ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

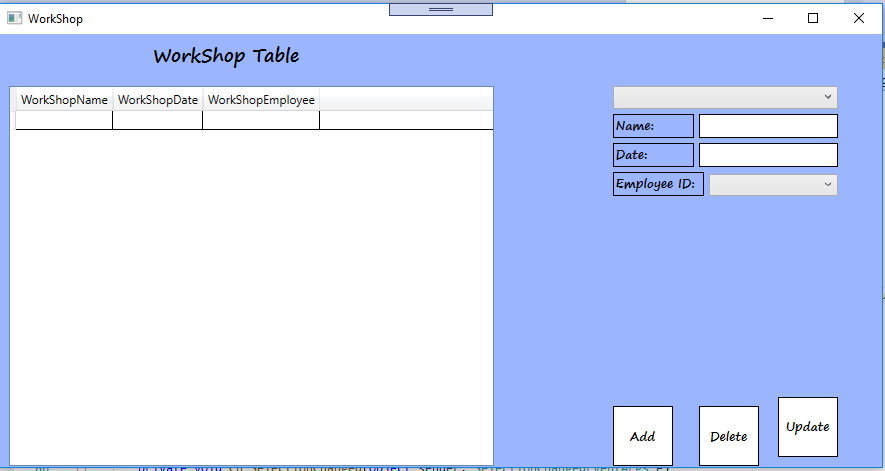
בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".



\*במסך זה ניתן להוסיף פרטי קבוצה ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את שם הקבוצה ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

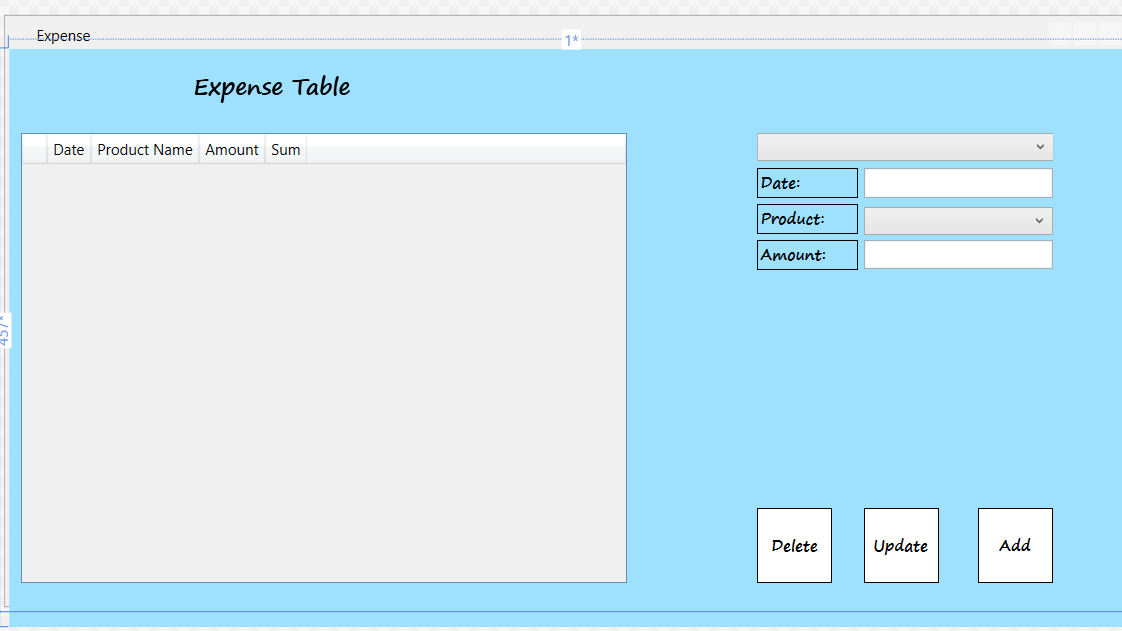
בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".



\*במסך זה ניתן להוסיף פרטי סדנא ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את שם הסדנא ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

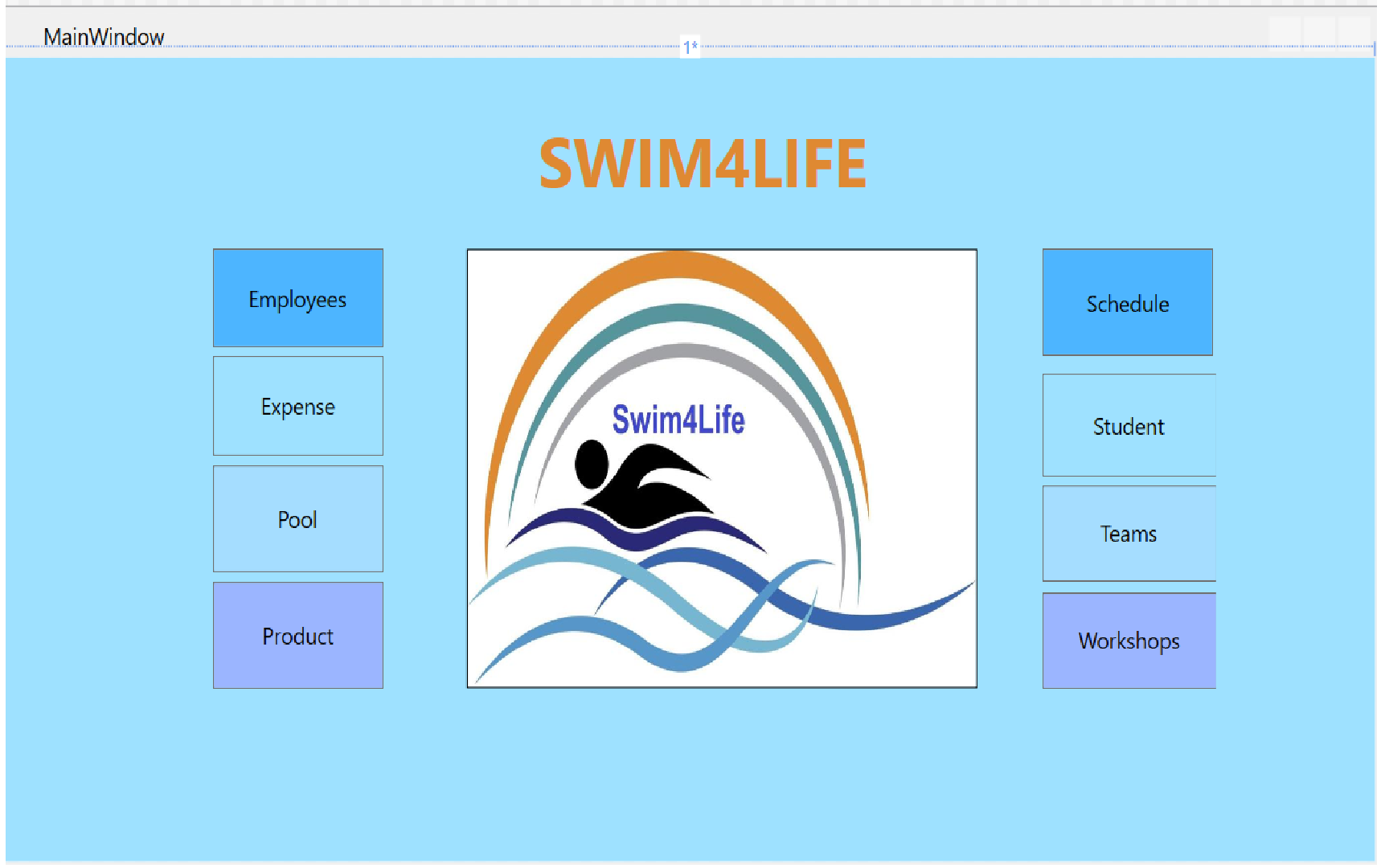
בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".



\*במסך זה ניתן להוסיף פרטי הוצאות ע"י מילוי השדות השונים ולחיצה על הכפתור "Add".

בכדי לעדכן שורה בטבלה יש לבחור בתיבה את תאריך ההוצאה ולמלא את הפרטים שנרצה לעדכן ולאחר מכן ללחוץ על הכפתור "Update".

בכדי למחוק שורה בטבלה יש לסמן אותה באמצעות העכבר וללחוץ על "Delete".



מסך פתיחה

קוד:

**Employee:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Employees.xaml

/// </summary>

public partial class Employees : Window

{

/\*database גישה \*/

dbEntities db = new dbEntities();

public Employees()

{

InitializeComponent();

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Employees\_Tbl> lst = (from s in db.Employees\_Tbl select s).ToList();

emp\_tbl.ItemsSource = lst;

emp\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

/\*מעביר את הנתונים שהוכנסו למשתנה מתאים

EmpN name of textBox\*/

String Name = EmpN.Text;

String Email = EmpE.Text;

String Phone = EmpP.Text;

String Function = EmpF.Text;

String Address = EmpA.Text;

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (EmpN.Text == "" || EmpF.Text == "" || EmpP.Text == "" || EmpA.Text == "")

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

Employees\_Tbl a1 = new Employees\_Tbl();

a1.EmpName = Name;

a1.EmpMail = Email;

a1.EmpPhone =Phone;

a1.EmpFunction = Function;

a1.EmpAddress = Address;

db.Employees\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "Employe Add";

EmpN.Text = "";

EmpE.Text = "";

EmpP.Text = "";

EmpF.Text = "";

EmpA.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

Employees\_Tbl a1 = (Employees\_Tbl)CB.SelectedItem;

Employees\_Tbl a2 = (from s in db.Employees\_Tbl where s.EmpId == a1.EmpId select s).FirstOrDefault();

a2.EmpName = EmpN.Text;

a2.EmpMail = EmpE.Text;

a2.EmpAddress = EmpA.Text;

a2.EmpFunction = EmpF.Text;

a2.EmpPhone = EmpP.Text;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void DataGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void OnLoad(object sender, RoutedEventArgs e)

{

/\*כאשר המסך נפתח\*/

UpdateDGV();

updateCB();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

/\*לחיצה על מחיקה\*/

db.Employees\_Tbl.Remove((Employees\_Tbl)emp\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Employees\_Tbl> lst = (from s in db.Employees\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "EmpName";

CB.SelectedValuePath = "EmpId";

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

/\* combobox change\*/

Employees\_Tbl a1 = (Employees\_Tbl)CB.SelectedItem;

if (a1 == null)

{

EmpN.Text = "";

EmpE.Text = "";

EmpP.Text = "";

EmpF.Text = "";

EmpA.Text = "";

return;

}

EmpN.Text =a1.EmpName;

EmpE.Text =a1.EmpMail;

EmpP.Text =a1.EmpPhone;

EmpF.Text =a1.EmpFunction;

EmpA.Text = a1.EmpAddress;

}

}

}

**Expense:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Expense.xaml

/// </summary>

public partial class Expense : Window

{

dbEntities db = new dbEntities();

public Expense()

{

InitializeComponent();

}

private void OnLoad2(object sender, RoutedEventArgs e)

{

UpdateDGV();

updateCB();

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Expense\_Tbl a1 = (Expense\_Tbl)CB.SelectedItem;

if (a1 == null)

{

ExpD.Text = "";

ExpA.Text = "";

ProdCombo.Text = "";

return;

}

ExpA.Text = a1.ExpenseAmount.ToString();

ExpD.Text = a1.ExpenseDate;

ProdCombo.Text = a1.Product\_tbl.ProdName;

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

int Price = (ProdCombo.SelectedItem as Product\_tbl).ProdPrice;

int Amount = Convert.ToInt32(ExpA.Text);

String Date = ExpD.Text;

String Pname = (ProdCombo.SelectedItem as Product\_tbl).ProdName;

int Id;

try { Id = (ProdCombo.SelectedItem as Product\_tbl).ProdId; }

catch

{

this.Title = "Erorr";

return;

}

int Sum = Price \* Amount;

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (ExpA.Text == "" || ExpD.Text == "")

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

Expense\_Tbl a1 = new Expense\_Tbl();

a1.ExpenseDate = Date;

a1.ExpenseAmount = Amount;

a1.ExpenseProductId = Id;

a1.ExpenseSum = Sum;

a1.ExpenseProductName = Pname;

db.Expense\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "Employee Add";

ExpD.Text = "";

ExpA.Text = "";

ProdCombo.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

Expense\_Tbl a1 = (Expense\_Tbl)CB.SelectedItem;

Expense\_Tbl a2 = (from s in db.Expense\_Tbl where s.ExpenseProductId == a1.ExpenseProductId select s).FirstOrDefault();

a2.ExpenseDate = ExpD.Text;

a2.ExpenseAmount = Convert.ToInt32(ExpA.Text);

a2.ExpenseProductId = (ProdCombo.SelectedItem as Product\_tbl).ProdId;

a2.ExpenseProductName = (ProdCombo.SelectedItem as Product\_tbl).ProdName;

a2.ExpenseSum = Convert.ToInt32(ExpA.Text) \* (ProdCombo.SelectedItem as Product\_tbl).ProdPrice;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

db.Expense\_Tbl.Remove((Expense\_Tbl)exp\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Expense\_Tbl> lst = (from s in db.Expense\_Tbl select s).ToList();

exp\_tbl.ItemsSource = lst;

exp\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Expense\_Tbl> lst = (from s in db.Expense\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "ExpenseDate";

CB.SelectedValuePath = "ExpenseId";

ProdCombo.ItemsSource = GetProduct\_s();

ProdCombo.DisplayMemberPath = "ProdName";

}

public List<Product\_tbl> GetProduct\_s()

{

return db.Product\_tbl.ToList();

}

}

}

**Main Window:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

dbEntities db = new dbEntities();

public MainWindow()

{

InitializeComponent();

}

private void Employee\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Employees employees = new Employees();

employees.ShowDialog();

this.Show();

}

private void Expense\_Click\_1(object sender, RoutedEventArgs e)

{

this.Hide();

Expense expense = new Expense();

expense.ShowDialog();

this.Show();

}

private void Product\_Click\_2(object sender, RoutedEventArgs e)

{

this.Hide();

Product product = new Product();

product.ShowDialog();

this.Show();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

}

private void schedule(object sender, RoutedEventArgs e)

{

this.Hide();

Schedule schedule = new Schedule();

schedule.ShowDialog();

this.Show();

}

private void student\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Student student = new Student();

student.ShowDialog();

this.Show();

}

private void Team\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Teams teams = new Teams();

teams.ShowDialog();

this.Show();

}

private void WorkShop\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

WorkShop work = new WorkShop();

work.ShowDialog();

this.Show();

}

private void pool\_Click\_3(object sender, RoutedEventArgs e)

{

this.Hide();

Pool pool = new Pool();

pool.ShowDialog();

this.Show();

}

}

}

**Pool:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Pool.xaml

/// </summary>

public partial class Pool : Window

{

dbEntities db = new dbEntities();

public Pool()

{

InitializeComponent();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

String Name = PoolN.Text;

String City = PoolC.Text;

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (PoolN.Text == "" || PoolC.Text == "")

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

Pool\_Tbl a1 = new Pool\_Tbl();

a1.PoolCity = City;

a1.PoolName = Name;

db.Pool\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "Employee Add";

PoolN.Text = "";

PoolC.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

db.Pool\_Tbl.Remove((Pool\_Tbl)pool\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

Pool\_Tbl a1 = (Pool\_Tbl)CB.SelectedItem;

Pool\_Tbl a2 = (from s in db.Pool\_Tbl where s.PoolId == a1.PoolId select s).FirstOrDefault();

a2.PoolCity = PoolC.Text;

a2.PoolName = PoolC.Text;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Pool\_Tbl a1 = (Pool\_Tbl)CB.SelectedItem;

if (a1 == null)

{

PoolN.Text = "";

PoolC.Text = "";

return;

}

PoolN.Text = a1.PoolName;

PoolC.Text = a1.PoolCity;

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Pool\_Tbl> lst = (from s in db.Pool\_Tbl select s).ToList();

pool\_tbl.ItemsSource = lst;

pool\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Pool\_Tbl> lst = (from s in db.Pool\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "PoolName";

CB.SelectedValuePath = "PoolId";

}

private void OnLoad(object sender, RoutedEventArgs e)

{

updateCB();

UpdateDGV();

}

}

}

**Product:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Product.xaml

/// </summary>

public partial class Product : Window

{

dbEntities db = new dbEntities();

public Product()

{

InitializeComponent();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

String Name = ProdN.Text;

int Price;

try { Price = Convert.ToInt32(ProdP.Text); }

catch

{

this.Title = "Erorr";

return;

}

if (ProdN.Text == "" || ProdP.Text == "")

{

this.Title = "Erorr";

return;

}

Product\_tbl a1 = new Product\_tbl();

a1.ProdName = Name;

a1.ProdPrice = Price;

db.Product\_tbl.Add(a1);

db.SaveChanges();

this.Title = "Employee Add";

ProdN.Text = "";

ProdP.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

db.Product\_tbl.Remove((Product\_tbl)prod\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

Product\_tbl a1 = (Product\_tbl)CB.SelectedItem;

Product\_tbl a2 = (from s in db.Product\_tbl where s.ProdId == a1.ProdId select s).FirstOrDefault();

a2.ProdName = ProdN.Text;

a2.ProdPrice = Convert.ToInt32(ProdP.Text);

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Product\_tbl a1 = (Product\_tbl)CB.SelectedItem;

if (a1 == null)

{

ProdN.Text = "";

ProdP.Text = "";

return;

}

ProdN.Text = a1.ProdName;

ProdP.Text = a1.ProdPrice.ToString();

}

private void OnLoad(object sender, RoutedEventArgs e)

{

updateCB();

UpdateDGV();

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Product\_tbl> lst = (from s in db.Product\_tbl select s).ToList();

prod\_tbl.ItemsSource = lst;

prod\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Product\_tbl> lst = (from s in db.Product\_tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "ProdName";

CB.SelectedValuePath = "ProdId";

}

}

}

**Schedule:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Schedule.xaml

/// </summary>

public partial class Schedule : Window

{

dbEntities db = new dbEntities();

public Schedule()

{

InitializeComponent();

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Schedule\_Tbl a1 = (Schedule\_Tbl)CB.SelectedItem;

if (a1 == null)

{

Pcombo.Text = "";

SchD.Text = "";

Tcombo.Text = "";

return;

}

Pcombo.Text = a1.Pool;

SchD.Text = a1.ScheduleDate;

Tcombo.Text = a1.Team;

}

private void ADD(object sender, RoutedEventArgs e)

{

int Pid,Tid;

String Pname = (Pcombo.SelectedItem as Pool\_Tbl).PoolName;

String Date = SchD.Text;

String Tname = (Tcombo.SelectedItem as Teams\_Tbl).TeamName;

try { Pid = (Pcombo.SelectedItem as Pool\_Tbl).PoolId; Tid = (Tcombo.SelectedItem as Teams\_Tbl).TeamId; }

catch

{

this.Title = "Erorr";

return;

}

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (SchD.Text == "" )

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

Schedule\_Tbl a1 = new Schedule\_Tbl();

a1.ScheduleDate = Date;

a1.Team = Tname;

a1.ScheduleTeamId = Tid;

a1.Pool = Pname;

a1.SchedulePoolId = Pid;

db.Schedule\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "schedule Add";

SchD.Text = "";

Pcombo.Text = "";

Tcombo.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Delete(object sender, RoutedEventArgs e)

{

db.Schedule\_Tbl.Remove((Schedule\_Tbl)sch\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void Update(object sender, RoutedEventArgs e)

{

Schedule\_Tbl a1 = (Schedule\_Tbl)CB.SelectedItem;

Schedule\_Tbl a2 = (from s in db.Schedule\_Tbl where s.ScheduleId == a1.ScheduleId select s).FirstOrDefault();

a2.ScheduleDate = SchD.Text;

a2.Team = (Tcombo.SelectedItem as Teams\_Tbl).TeamName;

a2.ScheduleTeamId = (Tcombo.SelectedItem as Teams\_Tbl).TeamId;

a2.Pool = (Pcombo.SelectedItem as Pool\_Tbl).PoolName;

a2.SchedulePoolId = (Pcombo.SelectedItem as Pool\_Tbl).PoolId;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Schedule\_Tbl> lst = (from s in db.Schedule\_Tbl select s).ToList();

sch\_tbl.ItemsSource = lst;

sch\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void OnLoad(object sender, RoutedEventArgs e)

{

UpdateDGV();

updateCB();

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Schedule\_Tbl> lst = (from s in db.Schedule\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "ScheduleDate";

CB.SelectedValuePath = "ScheduleId";

Tcombo.ItemsSource = null;

Tcombo.Text = "";

Tcombo.ItemsSource = GetTeams\_s();

Tcombo.DisplayMemberPath = "TeamName";

Pcombo.ItemsSource = GetPool();

Pcombo.DisplayMemberPath = "PoolName";

}

public List<Pool\_Tbl> GetPool()

{

return db.Pool\_Tbl.ToList();

}

public List<Teams\_Tbl> GetTeams\_s()

{

return db.Teams\_Tbl.ToList();

}

}

}

**Student:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Student.xaml

/// </summary>

public partial class Student : Window

{

dbEntities db = new dbEntities();

public Student()

{

InitializeComponent();

}

private void ComboBox\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Students\_Tbl a1 = (Students\_Tbl)CB.SelectedItem;

if (a1 == null)

{

StuN.Text = "";

StuM.Text = "";

StuP.Text = "";

StuAdd.Text = "";

StuAge.Text = "";

return;

}

StuN.Text = a1.StudentName;

StuM.Text = a1.StudentMail;

StuP.Text = a1.StudentPhone;

StuAdd.Text = a1.StudentAddress;

StuAge.Text = a1.StudentAge;

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

String Name = StuN.Text;

String Email = StuM.Text;

String Phone = StuP.Text;

String Age = StuAge.Text;

String Address = StuAdd.Text;

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (StuN.Text == "" || StuAge.Text == "" || StuP.Text == "" || StuAdd.Text == "")

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

Students\_Tbl a1 = new Students\_Tbl();

a1.StudentName = Name;

a1.StudentMail = Email;

a1.StudentPhone = Phone;

a1.StudentAge = Age;

a1.StudentAddress = Address;

db.Students\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "Employe Add";

StuN.Text = "";

StuM.Text = "";

StuP.Text = "";

StuAdd.Text = "";

StuAge.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

db.Students\_Tbl.Remove((Students\_Tbl)stu\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

Students\_Tbl a1 = (Students\_Tbl)CB.SelectedItem;

Students\_Tbl a2 = (from s in db.Students\_Tbl where s.StudentId == a1.StudentId select s).FirstOrDefault();

a2.StudentName = StuN.Text;

a2.StudentMail = StuM.Text;

a2.StudentAddress = StuAdd.Text;

a2.StudentAge = StuAge.Text;

a2.StudentPhone = StuP.Text;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Students\_Tbl> lst = (from s in db.Students\_Tbl select s).ToList();

stu\_tbl.ItemsSource = lst;

stu\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Students\_Tbl> lst = (from s in db.Students\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "StudentName";

CB.SelectedValuePath = "StudentId";

}

private void OnLoad(object sender, RoutedEventArgs e)

{

UpdateDGV();

updateCB();

}

}

}

**Teams:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for Teams.xaml

/// </summary>

public partial class Teams : Window

{

dbEntities db = new dbEntities();

public Teams()

{

InitializeComponent();

}

private void Add\_click(object sender, RoutedEventArgs e)

{

String Name = TeamN.Text;

String Date = TeamS.Text;

String Level = TeamL.Text;

String Ename = (Ecombo.SelectedItem as Employees\_Tbl).EmpName;

int Emp = (Ecombo.SelectedItem as Employees\_Tbl).EmpId;

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (TeamL.Text == "" || TeamN.Text == ""||Ecombo.Text=="")

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

Teams\_Tbl a1 = new Teams\_Tbl();

a1.TeamStartDate = Date;

a1.TeamName = Name;

a1.TeamEmployee =Emp ;

a1.TeamLevel = Level;

a1.TeamEmployeeName = Ename;

db.Teams\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "Employee Add";

TeamS.Text = "";

TeamN.Text = "";

Ecombo.Text = "";

TeamL.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

db.Teams\_Tbl.Remove((Teams\_Tbl)team\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

Teams\_Tbl a1 = (Teams\_Tbl)CB.SelectedItem;

Teams\_Tbl a2 = (from s in db.Teams\_Tbl where s.TeamId == a1.TeamId select s).FirstOrDefault();

a2.TeamStartDate = TeamS.Text;

a2.TeamName = TeamN.Text;

a2.TeamLevel = TeamL.Text;

a2.TeamEmployee = (Ecombo.SelectedItem as Employees\_Tbl).EmpId;

a2.TeamEmployeeName = (Ecombo.SelectedItem as Employees\_Tbl).EmpName;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Teams\_Tbl a1 = (Teams\_Tbl)CB.SelectedItem;

if (a1 == null)

{

TeamS.Text = "";

TeamN.Text = "";

Ecombo.Text = "";

TeamL.Text = "";

return;

}

TeamN.Text = a1.TeamName;

TeamS.Text = a1.TeamStartDate;

if(a1.TeamEmployee==null)

{

Ecombo.Text = "";

}

else

{

Ecombo.Text = a1.Employees\_Tbl.EmpName;

}

TeamL.Text = a1.TeamLevel;

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<Teams\_Tbl> lst = (from s in db.Teams\_Tbl select s).ToList();

team\_tbl.ItemsSource = lst;

team\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<Teams\_Tbl> lst = (from s in db.Teams\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "TeamName";

CB.SelectedValuePath = "TeamId";

Ecombo.ItemsSource = GetEmployees();

Ecombo.DisplayMemberPath = "EmpName";

}

public List<Employees\_Tbl> GetEmployees()

{

return db.Employees\_Tbl.ToList();

}

private void OnLoad(object sender, RoutedEventArgs e)

{

UpdateDGV();

updateCB();

}

}

}

**WorkShop:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace Swim4Life

{

/// <summary>

/// Interaction logic for WorkShop.xaml

/// </summary>

public partial class WorkShop : Window

{

dbEntities db = new dbEntities();

public WorkShop()

{

InitializeComponent();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

int Emp;

String Name = WorkN.Text;

String Date = WorkD.Text;

String EmpName= (Ecombo.SelectedItem as Employees\_Tbl).EmpName;

try { Emp = (Ecombo.SelectedItem as Employees\_Tbl).EmpId; }

catch

{

this.Title = "Erorr";

return;

}

/\*בודק שהוכנסו כל הנתונים שהם חובה

EmpN name of textBox\*/

if (WorkN.Text == "" || WorkD.Text == "" )

{

this.Title = "Erorr";

return;

}

/\* שומר את הנתונים שהוקלדו והוכנסו אל המשתנים המתאימים בטבלה

EmpName השם של העמודה בטבלה\*/

WorkShop\_Tbl a1 = new WorkShop\_Tbl();

a1.WorkShopDate = Date;

a1.WorkShopName = Name;

a1.WorkShopEmployee = Emp;

a1.WorkShopEmployeeName = EmpName;

db.WorkShop\_Tbl.Add(a1);

db.SaveChanges();

this.Title = "Employee Add";

WorkD.Text = "";

WorkN.Text = "";

Ecombo.Text = "";

this.Title = "sucsses";

UpdateDGV();

updateCB();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

db.WorkShop\_Tbl.Remove((WorkShop\_Tbl)work\_tbl.SelectedItem);

db.SaveChanges();

UpdateDGV();

updateCB();

}

private void Update\_Click(object sender, RoutedEventArgs e)

{

WorkShop\_Tbl a1 = (WorkShop\_Tbl)CB.SelectedItem;

WorkShop\_Tbl a2 = (from s in db.WorkShop\_Tbl where s.WorkShopId == a1.WorkShopId select s).FirstOrDefault();

a2.WorkShopDate = WorkD.Text;

a2.WorkShopName = WorkN.Text;

a2.WorkShopEmployee = (Ecombo.SelectedItem as Employees\_Tbl).EmpId;

a2.WorkShopEmployeeName = (Ecombo.SelectedItem as Employees\_Tbl).EmpName;

db.SaveChanges();

updateCB();

UpdateDGV();

}

private void CB\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

WorkShop\_Tbl a1 = (WorkShop\_Tbl)CB.SelectedItem;

if (a1 == null)

{

WorkN.Text = "";

WorkD.Text = "";

Ecombo.Text = "";

return;

}

WorkN.Text = a1.WorkShopName;

WorkD.Text = a1.WorkShopDate;

Ecombo.Text = a1.Employees\_Tbl.EmpName;

}

private void OnLoad(object sender, RoutedEventArgs e)

{

UpdateDGV();

updateCB();

}

private void UpdateDGV()

{

dbEntities db1 = new dbEntities();

db = db1;

List<WorkShop\_Tbl> lst = (from s in db.WorkShop\_Tbl select s).ToList();

work\_tbl.ItemsSource = lst;

work\_tbl.Columns[0].Visibility = Visibility.Hidden;

/\* פעולה לעדכון הטבלה\*/

}

private void updateCB()

{

/\*עדכון קומבובוקס\*/

CB.ItemsSource = null;

CB.Text = "";

List<WorkShop\_Tbl> lst = (from s in db.WorkShop\_Tbl select s).ToList();

CB.ItemsSource = lst;

CB.DisplayMemberPath = "WorkShopName";

CB.SelectedValuePath = "WorkShopId";

Ecombo.ItemsSource = GetEmployees();

Ecombo.DisplayMemberPath = "EmpName";

}

public List<Employees\_Tbl> GetEmployees()

{

return db.Employees\_Tbl.ToList();

}

}

}