



Telemarketing

מערכת לניהול תרומות

שם העבודה: טלמרקטינג

מגישה: תהילה הויזמן

ת.ז.: *****

המנחה: חני *****

חלופה: מערכות מנהליות

שנה: קיץ תשפ"ב

תוכן עניינים:

מבוא.....	3
תיאור הארגון.....	3
תיאור התוכנה.....	3
סיבות לבחירת הנושא.....	4
תרשים מבנה ארגוני של הארגון.....	5
מהות המשימה.....	5
תיחום המערכת.....	6
גורמים מעורבים.....	6
תיחום ארגוני.....	6
תיחום תהליכי.....	6
יעדים.....	6
אילוצי המערכת.....	7
מבנה וארכיטקטורה.....	7
מדריך למפתח.....	8
שכבת הנתונים.....	8
Dal.cs - המחלקה.....	8
BLL - שכבת העסק.....	10
GeneralDB - המחלקה.....	10
nameTableDB - מחלקות הרבים:.....	11
nameTable-מחלקות היחיד -.....	20
בסיס נתונים.....	36
קשרי גומלין.....	36
טבלאות.....	36
מדריך למשתמש.....	40
מסכי הפתיחה וטפסים.....	40-48
עץ תפריטים.....	49
רפלקציה.....	50

מבוא

תיאור הארגון

חברת "טלמרקטינג" היא חברה שהוקמה על מנת לתת מענה לארגוני הצדקה השונים ולעזור להם בארגון תרומות והתרמות טלפוניות. מטרת התוכנה: ניהול וסדר במבצע ההתרמה אשר מונע בזבוז זמן מיותר ותקלות מביכות. בעקבות ניסיון העבר כאשר ההתרמות נוהלו ע"י דפים ובשלב מתקדם יותר ע"י קבצי אקסל, נוצר מצב אשר תורמים פוטנציאליים טובים פוספסו, ולעומת זאת אנשים שכבר הותרמו לעיתים פנו אליהם שנית. עקב כך נוצר צורך במערכת אשר תנהל את מסע ההתרמה. המערכת נבנתה מתוך ניסיון מעשי ולמידת צרכי הארגונים.

תיאור התוכנה

המערכת נותנת מענה ל-2 סוגי משתמשים עיקריים, מנהל וטלפנית:

- 1. מנהל** - פיקוח ובקרה על התרומות הנכנסות, מעקב אחר עבודת הטלפניות, מערכת שעות העובדים, סכומי הכנסה, אפשרות לעריכת והוספת טלפניות וניהול כללי של פרויקטים.
- 2. טלפנית** - מערכת לשיחות יוצאות, הכנסת תרומות, ומעקב אחר התיק האישי.

התוכנה מאפשרת:

- ניהול תורמים פוטנציאליים, עריכה, הוספה, הצגה וכן אופציה לחיוג.
- ניהול תרומות, אפשרות לעריכה וכן פתיחת תרומה חדשה.
- מערכת לשיחות יוצאות המתנהלת ע"י כך שהמערכת שולפת בכל פעם תורם, והטלפנית מחייגת ומעדכנת סטטוס (לא ענה/ סרב/השהיה/פתיחת תרומה). במקרה של אי מענה המערכת משהה את התורם לחצי שעה. במידה והטלפנית נתבקשה לחזור מאוחר יותר, היא תלחץ על כפתור ההשהיה ותבחר שעת חזרה רצויה, ועד אז יושהה התורם. במקרה שהתורם הפוטנציאלי אינו מעוניין לתרום, הסטטוס שלו יתעדכן כתורם שהותרם ויושהה מהפרויקט הנוכחי.
- במידה והינו רוצה לתרום יפתח טופס לתרומה חדשה.
- נתונים סטטיסטיים שונים כמו סכום תרומה מקסימלי, סכום תרומות בפרויקט עד כה, סכום יעד בפרויקט וכו'.

להרשאת מנהל בלבד:

- ניהול עובדים, עריכה, הוספה וכן ניהול סיסמאות ושמות משתמשים.
- ניהול פרויקטים, עריכה והוספת פרויקטים חדשים, וכן צפיה בנתוני הפרויקטים השונים.
- ניהול מעקב עובדים, מעקב אחר שעות פעילות העובדים, שעת כניסה, יציאה וכו'.
- ניהול מערכת שעות, קביעת משמרות והצגת משמרות העובדים

סיבות לבחירת הנושא

בחרתי בנושא זה מאחר ועבדתי בעבר במס' ארגונים שונים ורובם ככולם התנהלו בצורה פרימיטיבית עד בלתי אפשרית, דבר שהראה לי את הצורך העז בתוכנה מסודרת לניהול, בנוסף אבי במקצועו כמתכנת עוסק לעיתים בתחום זה, בתור אחד שבונה מערכות לניהול מכירות סיניות, מאצ'ינג וכו' יצא לו מס' פעמים לבנות מערכות ניהול לתרומות וארגונים, דבר שסייע לי מאוד בהבנת העניין ובנייתו.

תרשים מבנה ארגוני של הארגון

מהות המשימה:

במערכת ההתרמה הידנית בה נערכו קמפייני ההתרמות בעבר נוצרו בעיות אחדות אשר המערכת הטכנולוגית עונה עליהם ואף נותנת שירותים נוספים:

1. המערכת הטכנולוגית בשונה מהמערכת הידנית שומרת פרטי מידע חשובים הנשמרים בצורה אוטומטית, כמו: פרטי שיחות, מעקב עובדים, ופרטים חשובים אשר במערכת ידנית לא ניתן לשמור.

2. המערכת הידנית עלולה לגרום לחוסר סינכרון, ולמצב בו יחייגו לתורם פעמיים ולעומת זאת לאחר לא יחייגו.

3. המערכת הידנית הקשתה על עבודת הטלפניות אשר היו צריכות לכתוב הכל בדפים ולעיתים קרובות אף נגרמו בעיות בהבנת הכתוב.

4. במערכת הטכנולוגית כאשר תורם מבקש שיחזרו אליו, המערכת מתזכרת בבא העת, בשונה ממערכת ידנית שלעיתים שוכחים לחזור.

5. במערכת הטכנולוגית בשונה מהידנית ניתן לראות היסטורית תרומות ונתונים מפרויקטים קודמים, פרטים שבידנית הולכים לאיבוד.

6. במערכת ידנית קשה לעקוב אחר נתוני סיכום וסטטיסטיקה.

תיחום המערכת:גורמים מעורבים:

גורם	תחום אחריות	רמת מעורבות
מנהל	-מעקב שעות כניסת ויציאת העובדים -ניהול מערכת שעות ומשמרות עובדים -ניהול פרויקטים -ניהול טלפניות -צפיה בנתוני מעקב עובדים	משתמש עיקרי
טלפנית	-ניהול תורמים -ביצוע תרומות -חיוג לתורמים ומערכת שיחות -חזרה לשיחות מושהות -צפיה בנתוני פרויקט	משתמש עיקרי

תיחום ארגוני:

המערכת נועדה לפעילותם של מנהל האירגון והטלפניות (מאותו תפריט (למעט מס' קטגוריות הפתוחות רק למנהל)).

תיחום תהליכי:

1. ניהול תורמים
2. ניהול עובדים
3. ניהול תרומות
4. ניהול שיחות
5. ניהול מעקב עובדים
6. ניהול פרויקטים
7. מערכת שיחות אוטומטית
8. מערכת שעות ומשמרות עובדים
9. מעקב שיחות מושהות
10. נתונים נוספים על הפרויקט
11. המערכת אינה מטפלת בחישוב משכורות עובדים.

יעדים:

שם היעד	תיאור
ניהול מאגרים	ניהול מאגרי המידע הבאים: תורמים, תרומות, שיחות, עובדים, פרויקטים, משמרות עובדים.
טיפול בפרטי תורם	<ul style="list-style-type: none"> קליטת נתונים פורמליים של תורם והוספתם למאגר התורמים. עדכון פרטי תורם לפי דרישת הטלפנית או המנהל. מציאת פרטי תורם בחיפוש חכם או עפ"י קטגוריה וכן אפשרות סינון. הצגת פרטי תורם מסוים.
טיפול בפרטי תרומה	<ul style="list-style-type: none"> קליטת נתונים פורמליים של תרומה והוספתם למאגר התרומות. עדכון פרטי תרומה לפי דרישת הטלפנית או המנהל. מציאת פרטי תרומה בחיפוש חכם או עפ"י קטגוריה וכן אפשרות סינון. הצגת פרטי תרומה מסוימת.
טיפול בפרטי שיחה	<ul style="list-style-type: none"> קליטת נתוני שיחה בצורה אוטומטית ע"י המערכת והוספתם למאגר השיחות. מציאת פרטי שיחה בחיפוש חכם או עפ"י קטגוריה וכן אפשרות סינון. הצגת פרטי שיחה מסוימת.
טיפול בפרטי עובד	<ul style="list-style-type: none"> קליטת נתונים פורמליים של עובד (טלפנית) והוספתם למאגר העובדים. עדכון פרטי עובד לפי דרישת הטלפנית או המנהל. מציאת פרטי עובד בחיפוש חכם או עפ"י קטגוריה וכן אפשרות סינון. הצגת פרטי עובד מסוים.
טיפול בפרטי פרויקט	<ul style="list-style-type: none"> קליטת נתונים פורמליים של פרויקט והוספתם למאגר הפרויקטים. עדכון פרטי פרויקט לפי דרישת המנהל. מציאת פרטי פרויקט בחיפוש חכם או עפ"י קטגוריה וכן אפשרות סינון. הצגת פרטי פרויקט מסוים.

טיפול בפרטי משמרות עובדים	•קליטת טלפניות למשמרת ע"י בחירת המנהל בעובדות הרצויות והוספתם למאגר משמרות עובדים (Scedule). •הצגת עובדות המשמרת בריבוע המשמרת המתאים.
---------------------------	--

אילוצי המערכת:

- א. לוו"ז- הגשת הפרויקט עד סוף יוני 2022.
- ב. כ"א- פיתוח ומימוש התבצע ע"י תלמידת תיכון במסגרת פרויקט גמר.
- ג. טכנולוגיה- שימוש במחשב PC.
- ד. סביבת פיתוח- visual studio 10 , C# , מסד נתונים Access 2016 בטכנולוגית Window Forms.

מבנה וארכיטקטורה:

תוכנת 'טלמרקטינג' היא תוכנה לניהול ארגון המתייחסת לתהליך המרכזי:

התרמת תורמים, ניהול משמרות עובדים, וניהול פרויקטים.

התוכנה נכתבה בשפת #C בסביבת Visual Studio 2019.

מחלקות ועצמים, קלט/פלט, ממשקי משתמש ידידותיים, תמיכה בבסיס נתונים, הורשה, טפסים.

התוכנית נבנתה בצורת חשיבה של DNA – מודל שלושת השכבות.

התוכנית חולקה לשלוש שכבות כאשר כל שכבה מטפלת בתחום שונה ומכילה כמה מחלקות ו/או טפסים שונים. בתוך כל מחלקה הוגדרו משתנים ואובייקטים מתאימים המיוחדים למחלקה. כל מחלקה מטפלת בתחום מסוים ומתקשרת לטופס/המחלקה המתאימים לה. צורת כתיבה זו נועדה להקל על המתכנת הן בכתיבת הקוד והן בתחזוקה השוטפת של הקוד. כמובן שבין המחלקות והטפסים שבשכבות השונות קיים קשר, וקימת התייחסות מתוך מחלקה אחת לעצמים המוגדרים במחלקות אחרות אך הכל נעשה תוך שמירה על בטחון והבטחת הנתונים והפונקציות.

פרוט השכבות:

1. DAL Data Access Layer – שכבה המטפלת בקישור התוכנה עם בסיס הנתונים.
2. BLL Business Logic Layer – שכבה המטפלת בכל הפונקציות הקשורות לתוכנה עצמה.
3. GUI Graphic User Interface – שכבה המטפלת בממשק החיצוני של התוכנה.

מדריך למפתח

שכבת הנתונים

שכבה זו מטפלת בקישור התוכנה עם בסיס הנתונים.

ובה מחלקה לתקשורת עם ה-Dal.cs . DataBase

המחלקה Dal.cs

היחידה ש"מדברת" עם ה- DataBase כל שאר המחלקות פונות ל- DataBase דרך מחלקה זו. במחלקה זו יש מספר פונקציות ששולפות, מעדכנות, מוחקות ומוסיפות ל- DataBase.

```

internal class DalA
{
    private static DalA instance;
    private OleDbConnection con;
    private DataSet ds;
    private DalA(string connectionString)
    {
        con = new OleDbConnection(connectionString);
        ds = new DataSet();
    }
    public static DalA GetInstance()
    {
        if (instance == null)
        {
            string path = System.IO.Directory.GetCurrentDirectory();
            int x = path.IndexOf("\\bin");
            path = path.Substring(0, x) + "\\Dal\\TelemarketingDB.accdb";
            instance = new DalA(@"provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + path + ";Persist Security Info=True");
        }
        return instance;
    }
    public void AddTable(string tableName)
    {
        if (!ds.Tables.Contains(tableName))
        {
            OleDbDataAdapter adapter = new OleDbDataAdapter("Select * from " + tableName, con);
            adapter.Fill(ds, tableName);
        }
    }
    public DataTable GetTable(string tableName)
    {
        if (!ds.Tables.Contains(tableName))
        AddTable(tableName);
        return ds.Tables[tableName];
    }
}

```



```

}
public bool RemoveTable(string tableName)
{
    bool succeed = true;

```

```

try
{
    ds.Tables.Remove(tableName);
}
catch
{
    succeed = false;
}
return succeed;
}
public DataTable GetSelectTable(string sql)
{
    DataTable dt = new DataTable();
    OleDbDataAdapter adapter = new OleDbDataAdapter(sql, con);
    adapter.Fill(dt);
    return dt;
}
public void Update(string tableName)
{
    OleDbDataAdapter adapter = new OleDbDataAdapter("select * from " + tableName, con);
    OleDbCommandBuilder builder = new OleDbCommandBuilder(adapter);
    adapter.InsertCommand = builder.GetInsertCommand();
    adapter.UpdateCommand = builder.GetUpdateCommand();
    adapter.DeleteCommand = builder.GetDeleteCommand();
    adapter.Update(ds, tableName);
}
public void Update()
{
    foreach (DataTable table in ds.Tables)
    {
        Update(table.TableName);
    }
}
}

```

שכבת העסק - BLL

שכבה המטפלת בכל הפונקציות והלוגיקה הקשורות לתוכנה עצמה.

המחלקות המוכלות בשכבה:

מחלקת בסיס לטיפול בטבלה אחת מתוך ה DataBase – GeneralDB

מחלקה עבור כל טבלה מה DataBase היורשת ממהמחלקה GeneralDB

מחלקות nameTableRow – תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני.

המחלקה GeneralDB

היא היחידה המתקשרת עם מחלקת Dal. מנהלת קשר עם טבלה בודדת מתוך ה DataBase. לא ניתן ליצור

ממנה עצמים כיוון שהיא כללית ולא ספציפית לטבלה מסוימת ב DataBase. המחלקות היורשות ממנה הם

מתייחסות כל אחת לטבלה אחרת מהמאגר.

תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני.

```
internal class GeneralDB
{
    protected DataTable table;
    public GeneralDB(string tableName)
    {
        Dal.DalA.GetInstance().AddTable(tableName);
        table = Dal.DalA.GetInstance().GetTable(tableName);
    }
    public DataTable GetTable()
    {
        return this.table;
    }
    public virtual void Save()
    {
        Dal.DalA.GetInstance().Update(table.TableName);
        table = Dal.DalA.GetInstance().GetTable(table.TableName);
    }
    public int Size()
    {
        return table.Rows.Count;
    }
    public bool IsEmpty()
    {
        return table.Rows.Count == 0;
    }
    public void Add(DataRow dr)
    {
        table.Rows.Add(dr);
        this.Save();
    }

    public void Update()
    {
        Dal.DalA.GetInstance().Update(table.TableName);
    }
    public void Remove(DataRow dr)
    {

```

```

        table.Rows.Remove(dr);
        this.Save();
    }
}

```

מחלקות הרבים: nameTableDB:

כל אחת מן המחלקות הבאות יורשת מהמחלקה GeneralDB ומנהלת את הטיפול בטבלה בודדת מתוך ה DataBase

המחלקה CitiesDB

```

internal class CitiesDB:GeneralDB
{
    protected List<City> list = new List<City>();
    public CitiesDB() : base("Cities") { }
    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new City(dr));
        }
    }
    public List<City> GetList()
    {
        list.Clear();
        DataTableToList();
        return list;
    }
    public void AddNew(City c)
    {
        c.Dr = table.NewRow();
        c.PutInto();
        this.Add(c.Dr);
    }
    public void RemoveCustomer(City c)
    {
        this.Remove(c.Dr);
    }
    public City Find(int code)
    {
        return this.GetList().Find(x => x.CityID == code);
    }
    public void UpdateRow(City c)
    {
        c.PutInto();
        this.Update();
    }
    public void DeleteRow(int code)
    {
        City City = this.Find(code);
        if (City != null)
        {
            City.Dr.Delete();
            this.Update();
        }
    }
}

```

```

    }
    public int GetNextKey()
    {
        if (this.Size() == 0)
            return 1;
        return this.GetList().Max(x => x.CityID) + 1;
    }
}

```

המחלקה ConversationsDB

```

internal class ConversationsDB:GeneralDB
{
    protected List<Conversation> list = new List<Conversation>();
    public ConversationsDB() : base("Conversations") { }
    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new Conversation(dr));
        }
    }
    public List<Conversation> GetList()
    {
        list.Clear();
        DataTableToList();
        return list;
    }
    public void AddNew(Conversation c)
    {
        c.Dr = table.NewRow();
        c.PutInto();
        this.Add(c.Dr);
    }
    public void RemoveCustomer(Conversation c)
    {
        this.Remove(c.Dr);
    }
    public Conversation Find(int code)
    {
        return this.GetList().Find(x => x.ConversationID == code);
    }
    public void UpdateRow(Conversation c)
    {
        c.PutInto();
        this.Update();
    }
    public void DeleteRow(int code)
    {
        Conversation conversation = this.Find(code);
        if (conversation != null)
        {
            conversation.Dr.Delete();
            this.Update();
        }
    }
}

```

```

public int GetNextKey()
{
    if (this.Size() == 0)
        return 1;
    return this.GetList().Max(x => x.ConversationID) + 1;
}
}

```

המחלקה DonationsDB

```

internal class DonationsDB : GeneralDB
{
    protected List<Donation> list = new List<Donation>();
    public DonationsDB() : base("Donations") { }

    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new Donation(dr));
        }
    }
    public List<Donation> GetList()
    {
        list.Clear();
        DataTableToList();
        return list;
    }
    public void AddNew(Donation d)
    {
        d.Dr = table.NewRow();
        d.PutInto();
        this.Add(d.Dr);
    }
    public void RemoveCustomer(Donation d)
    {
        this.Remove(d.Dr);
    }
    public Donation Find(int code)
    {
        return this.GetList().Find(x => x.DonationID == code);
    }
    public void UpdateRow(Donation d)
    {
        d.PutInto();
        this.Update();
    }
    public void DeleteRow(int code)
    {
        Donation donation = this.Find(code);
        if (donation != null)
        {
            donation.Dr.Delete();
            this.Update();
        }
    }
    public int GetNextKey()
    {

```

```

        if (this.Size() == 0)
            return 1;
        return this.GetList().Max(x => x.DonationID) + 1;
    }
}

```

המחלקה DonorsDB

```

internal class DonorsDB : GeneralDB
{
    protected List<Donor> list = new List<Donor>();

    public DonorsDB() : base("Donors") { }

    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new Donor(dr));
        }
    }

    public List<Donor> GetList()
    {
        list.Clear();

        DataTableToList();

        return list;
    }

    public void AddNew(Donor d)
    {
        d.Dr = table.NewRow();

        d.PutInto();

        this.Add(d.Dr);
    }

    public void RemoveDonor(Donor d)
    {
        this.Remove(d.Dr);
    }
}

```

```

public Donor Find(int code)
{
    return this.GetList().Find(x => x.DonorID == code);
}

public void UpdateRow(Donor d)
{
    d.PutInto();
    this.Update();
}

public void DeleteRow(int code)
{
    Donor donor = this.Find(code);
    if (donor != null)
    {
        donor.Dr.Delete();
        this.Update();
    }
}

public int GetNextKey()
{
    if (this.Size() == 0)
        return 1;

    return this.GetList().Max(x => x.DonorID) + 1;
}
}

```

המחלקה EmployeeMonitoringDB

```

internal class EmployeeMonitoringDB : GeneralDB
{
    protected List<EmployeeMonitoring> list = new List<EmployeeMonitoring>();
    public EmployeeMonitoringDB() : base("Employee_Monitoring") { }
    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new EmployeeMonitoring(dr));
        }
    }
}

```

```

    }
}
public List<EmployeeMonitoring> GetList()
{
    list.Clear();
    DataTableToList();
    return list;
}
public void AddNew(EmployeeMonitoring em)
{
    em.Dr = table.NewRow();
    em.PutInto();
    this.Add(em.Dr);
}
public void RemoveCustomer(EmployeeMonitoring em)
{
    this.Remove(em.Dr);
}
public EmployeeMonitoring Find(int code)
{
    return this.GetList().Find(x => x.EmployeeMonitoringID == code);
}
public void UpdateRow(EmployeeMonitoring em)
{
    em.PutInto();
    this.Update();
}
public void DeleteRow(int code)
{
    EmployeeMonitoring employeeMonitoring = this.Find(code);
    if (employeeMonitoring != null)
    {
        employeeMonitoring.Dr.Delete();
        this.Update();
    }
}
public int GetNextKey()
{
    if (this.Size() == 0)
        return 1;
    return this.GetList().Max(x => x.EmployeeMonitoringID) + 1;
}
}

```

המחלקה EmployeesDB

```

internal class EmployeesDB : GeneralDB
{
    protected List<Employee> list = new List<Employee>();
    public EmployeesDB() : base("Employees") { }

    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new Employee(dr));
        }
    }
}

```



```

    }

    public List<Employee> GetList()
    {
        list.Clear();
        DataTableToList();
        return list;
    }
    public void AddNew(Employee em)
    {
        em.Dr = table.NewRow();
        em.PutInto();
        this.Add(em.Dr);
    }
    public void RemoveCustomer(Employee em)
    {
        this.Remove(em.Dr);
    }
    public Employee Find(int code)
    {
        return this.GetList().Find(x => x.EmployeeID == code);
    }

    public void UpdateRow(Employee em)
    {
        em.PutInto();
        this.Update();
    }
    public void DeleteRow(int code)
    {
        Employee employees = this.Find(code);
        if (employees != null)
        {
            employees.Dr.Delete();
            this.Update();
        }
    }

    public int GetNextKey()
    {
        if (this.Size() == 0)
            return 1;
        return this.GetList().Max(x => x.EmployeeID) + 1;
    }
}

```

המחלקה ProjectsDB

```

internal class ProjectsDB : GeneralDB
{
    protected List<Project> list = new List<Project>();
    public ProjectsDB() : base("Projects") { }
    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new Project(dr));
        }
    }
}

```

```

    }
    public List<Project> GetList()
    {
        list.Clear();
        DataTableToList();
        return list;
    }
    public void AddNew(Project p)
    {
        p.Dr = table.NewRow();
        p.PutInto();
        this.Add(p.Dr);
    }
    public void RemoveCustomer(Project p)
    {
        this.Remove(p.Dr);
    }
    public Project Find(int code)
    {
        return this.GetList().Find(x => x.ProjectID == code);
    }

    public void UpdateRow(Project p)
    {
        p.PutInto();
        this.Update();
    }
    public void DeleteRow(int code)
    {
        Project project = this.Find(code);
        if (project != null)
        {
            project.Dr.Delete();
            this.Update();
        }
    }
    public int GetNextKey()
    {
        if (this.Size() == 0)
            return 1;
        return this.GetList().Max(x => x.ProjectID) + 1;
    }
}

```

המחלקה SceduleDB

```

internal class SceduleDB:GeneralDB
{
    protected List<Scedule> list = new List<Scedule>();
    public SceduleDB() : base("Scedule") { }
    public void DataTableToList()
    {
        foreach (DataRow dr in table.Rows)
        {
            list.Add(new Scedule(dr));
        }
    }
    public List<Scedule> GetList()
    {

```

```

        list.Clear();
        DataTableToList();
        return list;
    }
    public void AddNew(Schedule s)
    {
        s.Dr = table.NewRow();
        s.PutInto();
        this.Add(s.Dr);
    }
    public void RemoveCustomer(City c)
    {
        this.Remove(c.Dr);
    }
    public Schedule Find(int code)
    {
        return this.GetList().Find(x => x.ID1 == code);
    }
    public void UpdateRow(Schedule s)
    {
        s.PutInto();
        this.Update();
    }
    public void DeleteRow(int code)
    {
        Schedule schedule = this.Find(code);
        if (schedule != null)
        {
            schedule.Dr.Delete();
            this.Update();
        }
    }
    public int GetNextKey()
    {
        if (this.Size() == 0)
            return 1;
        return this.GetList().Max(x => x.ID1) + 1;
    }
}

```

מחלקות היחיד - nameTable

כל מחלקה מהן, מטפלת ברשומה אחת מתוך טבלה מסוימת.

תכונותיה: משתנים המקבילים לשדות של אותה טבלה (גם בסוג הנתונים), אובייקט מסוג מחלקת DB השייכת לטבלה זו ואובייקט מסוג DataRow.

המחלקה City

```

public class City
{
    DataRow dr;

    private int cityID;

    private string cityName;

    public DataRow Dr { get => dr; set => dr = value; }

    public int CityID { get => cityID; set { cityID = value; } }

    public string CityName { get => cityName; set { if (!Validation.IsHebrew(value)) throw new
Exception("הכנס עיר בעברית בלבד"); cityName = value; } }

    public City() { }

    public City(DataRow dr)
    {
        this.dr = dr;

        this.cityID = Convert.ToInt32(dr["City_ID"]);

        this.cityName = dr["City_Name"].ToString();
    }

    public void PutInto()
    {
        dr["City_ID"] = this.cityID;

        dr["City_Name"] = this.cityName;
    }

    public override string ToString()
    {
        return cityName;
    }
}

```

המחלקה Conversation

```

internal class Conversation
{
    DataRow dr;

    private int conversationID;

    private int donorID;

    private int employeeID;

    private string status;

    private DateTime timeToBack;

    private string note;

    private string state;

    private DateTime date;

    public DataRow Dr { get => dr; set => dr = value; }

    public int ConversationID { get => conversationID; set { conversationID = value; } }

    public int DonorID { get => donorID; set { donorID = value; } }

    public int EmployeeID { get => employeeID; set { employeeID = value; } }

    public string Status { get => status; set => status = value; }

    public DateTime TimeToBack { get => timeToBack; set => timeToBack = value; }

    public string Note { get => note; set => note = value; }

    public DateTime Date { get => date; set => date = value; }

    public string State { get => state; set => state = value; }

    public Conversation() { }

    public Conversation(DataRow dr)
    {
        this.dr = dr;

        this.conversationID = Convert.ToInt32(dr["Conversation_ID"]);

        this.donorID = Convert.ToInt32(dr["Donor_ID"]);
    }

```

```

this.employeeID = Convert.ToInt32(dr["Employee_ID"]);

this.status = dr["Status1"].ToString();

this.timeToBack = Convert.ToDateTime(dr["Time_To_Back"]);

this.note = dr["Note1"].ToString();

this.date = Convert.ToDateTime(dr["Date1"]);

this.state = dr["State1"].ToString();
}

public void PutInto()
{
    dr["Conversation_ID"] = this.conversationID;

    dr["Donor_ID"] = this.donorID;

    dr["Employee_ID"] = this.employeeID;

    dr["Status1"] = this.status;

    dr["Time_To_Back"] = this.timeToBack;

    dr["Note1"] = this.note;

    dr["Date1"] = this.date;

    dr["State1"] = this.state;
}

EmployeesDB tblEmployees = new EmployeesDB();

DonorsDB tblDonors = new DonorsDB();

public Employee ThisEmployee()
{
    return tblEmployees.GetList().Find(x => x.EmployeeID == this.EmployeeID);
}

public Donor ThisDonor()
{
    return tblDonors.GetList().Find(x => x.DonorID == this.DonorID);
}

```

```

public override string ToString()
{
    return conversationID.ToString();
}
}

```

המחלקה Donation

```

public class Donation
{
    DataRow dr;

    private int donationID;

    private int donorID;

    private int employeeID;

    private int projectID;

    private int sum;

    private string paymentOption;

    private string creditNumber;

    private DateTime validity;

    private string CVV;

    private int numberPayments;

    private string paymentStatus;

    private DateTime donationDate;

    private string cityName;

    public DataRow Dr { get => dr; set => dr = value; }

    public int DonationID { get => donationID; set => donationID = value; }

    public int DonorID { get => donorID; set => donorID = value; }

    public int EmployeeID { get => employeeID; set => employeeID = value; }

    public int ProjectID { get => projectID; set => projectID = value; }

    public int Sum { get => sum; set => sum = value; }

    public string PaymentOption { get => paymentOption; set => paymentOption = value; }

    public string CreditNumber { get => creditNumber; set => creditNumber = value; }
}

```

```

public DateTime Validity { get => validity; set => validity = value; }

public string CVV1 { get => CVV; set => CVV = value; }

public int NumberPayments { get => numberPayments; set => numberPayments = value; }

public string PaymentStatus { get => paymentStatus; set => paymentStatus = value; }

public DateTime DonationDate { get => donationDate; set => donationDate = value; }

public Donation() { }

public Donation(DataRow dr)

{
    this.dr = dr;

    this.DonationID = Convert.ToInt32(dr["Donation_ID"]);

    this.DonorID = Convert.ToInt32(dr["Donor_ID"]);

    this.EmployeeID = Convert.ToInt32(dr["Employee_ID"]);

    this.ProjectID = Convert.ToInt32(dr["Project_ID"]);

    this.Sum = Convert.ToInt32(dr["Sum1"]);

    this.PaymentOption = dr["Payment_Option"].ToString();

    this.CreditNumber = dr["Credit_Number"].ToString();

    this.Validity = Convert.ToDateTime(dr["Validity"]);

    this.CVV1 = dr["CVV"].ToString();

    this.NumberPayments = Convert.ToInt32(dr["Number_Payments"]);

    this.PaymentStatus = dr["Payment_Status"].ToString();

    this.DonationDate = Convert.ToDateTime(dr["Donation_Date"]);

}

public void PutInto()

{
    dr["Donation_ID"] = this.DonationID;

    dr["Donor_ID"] = this.DonorID;

    dr["Employee_ID"] = this.EmployeeID;

    dr["Project_ID"] = this.ProjectID;

    dr["Sum1"] = this.Sum;

    dr["Payment_Option"] = this.PaymentOption;

```



```

dr["Credit_Number"] = this.CreditNumber;

dr["Validity"] = this.Validity;

dr["CVV"] = this.CVV1;

dr["Number_Payments"] = this.NumberPayments;

dr["Payment_Status"] = this.PaymentStatus;

dr["Donation_Date"] = this.DonationDate;
}

DonorsDB tblDonors = new DonorsDB();

EmployeesDB tblEmployees = new EmployeesDB();

ProjectsDB tblProjects = new ProjectsDB();

public Donor ThisDonor()
{
    return tblDonors.GetList().Find(x => x.DonorID == this.DonorID);
}

public Employee ThisEmployee()
{
    return tblEmployees.GetList().Find(x => x.EmployeeID == this.EmployeeID);
}

public Project ThisProject()
{
    return tblProjects.GetList().Find(x => x.ProjectID == this.ProjectID);
}

public override string ToString()
{
    return DonorID+" "+Sum;
}

public void Confirmed()
{
    this.paymentStatus = "אושר";
}

```

```

    }

    public void Denied()

    {

        this.paymentStatus = "נדחה";

    }

}

```

המחלקה Donor

```

public class Donor

{

    DataRow dr;

    private int donorID;

    private string firstName;

    private string lastName;

    private string phone;

    private int cityID;

    private string street;

    private int houseNumber;

    private string email;

    private string connection;

    private string note;

    private string status;

    private string state;

    public DataRow Dr { get => dr; set => dr = value; }

    public int DonorID { get => donorID; set => donorID = value; }

    public string FirstName { get => firstName; set { if (!Validation.IsHebrew(value)) throw new Exception("הכנס שם בעברית"); firstName = value; } }

    public string LastName { get => lastName; set { if (!Validation.IsHebrew(value)) throw new Exception("הכנס שם בעברית"); lastName = value; } }

    public string Phone { get => phone; set { if (!Validation.IsPelepon(value)&&!Validation.IsTel(value)) throw new Exception("מספר טלפון לא תקין"); phone = value; } }

    public int CityID { get => cityID; set { cityID = value; } }
}

```

```

public string Street { get => street; set { if (!Validation.IsHebrew(value)) throw new Exception("הכנס רחוב בעברית"); street = value; } }

public int HouseNumber { get => houseNumber; set { if (!Validation.IsNum(value.ToString())) throw new Exception("הכנס מספר"); houseNumber = value; } }

public string Email { get => email; set { if (!Validation.IsMail(value)) throw new Exception("אימייל לא תקין"); email = value; } }

public string Connection { get => connection; set => connection = value; }

public string Note { get => note; set => note = value; }

public string Status { get => status; set => status = value; }

public string State { get => state; set => state = value; }

public Donor() { }

public Donor(DataRow dr)
{
    this.dr = dr;

    this.donorID = Convert.ToInt32(dr["Donor_ID"]);

    this.firstName = dr["First_Name"].ToString();

    this.lastName = dr["Last_Name"].ToString();

    this.phone = dr["Phone1"].ToString();

    this.cityID = Convert.ToInt32(dr["City_ID"]);

    this.street = dr["Street1"].ToString();

    this.houseNumber = Convert.ToInt32(dr["House_Number"]);

    this.email = dr["Email1"].ToString();

    this.connection = dr["Connection1"].ToString();

    this.note = dr["Note1"].ToString();

    this.status = dr["Status1"].ToString();

    this.state = dr["State1"].ToString();
}

public void PutInto()
{
    dr["Donor_ID"] = this.donorID;

    dr["First_Name"] = this.firstName;

    dr["Phone1"] = this.phone;

    dr["City_ID"] = this.cityID;
}

```

```
dr["Last_Name"] = this.lastName;

dr["Street1"] = this.street;

dr["House_Number"] = this.houseNumber;

dr["Email1"] = this.email;

dr["Connection1"] = this.connection;

dr["Note1"] = this.note;

dr["Status1"] = this.status;

dr["State1"] = this.state;
}

CitiesDB tblCities = new CitiesDB();

public City ThisCity()
{
    return tblCities.GetList().Find(x => x.CityID == this.CityID);
}

public void Active()
{
    this.status = "פעיל";
}

public void InActive()
{
    this.status = "לא פעיל";
}

public void Completed()
{
    this.state = "טופל";
}

public void NotComplete()
{
    this.state = "לא טופל";
}
```

```

public override string ToString()
{
    return FirstName + " " + LastName;
}
}

```

המחלקה Employee

```

public class Employee
{
    DataRow dr;

    private int employeeID;

    private string firstName;

    private string lastName;

    private string phone;

    private string status;

    private string state;

    private string level;

    private string userName;

    private string password;

    public DataRow Dr { get => dr; set => dr = value; }

    public int EmployeeID { get => employeeID; set => employeeID = value; }

    public string FirstName { get => firstName; set => firstName = value; }

    public string LastName { get => lastName; set => lastName = value; }

    public string Phone { get => phone; set => phone = value; }

    public string Status { get => status; set => status = value; }

    public string State { get => state; set => state = value; }

    public string Level { get => level; set => level = value; }

    public string UserName { get => userName; set => userName = value; }

    public string Password { get => password; set => password = value; }

    public Employee() { }

    public Employee(DataRow dr)

```

```
{  
  
    this.dr = dr;  
  
    this.employeeID = Convert.ToInt32(dr["Employee_ID"]);  
  
    this.firstName = dr["First_Name"].ToString();  
  
    this.lastName = dr["Last_Name"].ToString();  
  
    this.phone = dr["Phone1"].ToString();  
  
    this.status = dr["Status1"].ToString();  
  
    this.state = dr["State1"].ToString();  
  
    this.level = dr["Level1"].ToString();  
  
    this.userName = dr["User_Name"].ToString();  
  
    this.password = dr["Password1"].ToString();  
  
}  
  
public void PutInto()  
  
{  
  
    dr["Employee_ID"] = this.employeeID;  
  
    dr["First_Name"] = this.firstName;  
  
    dr["Last_Name"] = this.lastName;  
  
    dr["Phone1"] = this.phone;  
  
    dr["Status1"] = this.status;  
  
    dr["State1"] = this.state;  
  
    dr["Level1"] = this.level;  
  
    dr["User_Name"] = this.userName;  
  
    dr["Password1"] = this.password;  
  
}  
  
public void Active()  
  
{  
  
    this.status = "פעיל";  
  
}  
  
public void InActive()  
  
{
```

```

        this.status = "לא פעיל";
    }

    public void Connected()
    {
        this.state = "מחובר";
    }

    public void NotConnected()
    {
        this.state = "לא מחובר";
    }

    public override string ToString()
    {
        return firstName+" "+lastName;
    }
}

```

המחלקה EmployeeMonitoring

```

internal class EmployeeMonitoring
{
    DataRow dr;

    private int employeeMonitoringID;

    private DateTime entranceTime;

    private DateTime leavingTime;

    private int employeeID;

    private DateTime totalTime;

    private int totalHours;

    private int totalMinutes;

    public DataRow Dr { get => dr; set => dr = value; }

    public int EmployeeMonitoringID { get => employeeMonitoringID; set {employeeMonitoringID = value; } }

    public DateTime EntranceTime { get => entranceTime; set => entranceTime = value; }
}

```

```

public DateTime LeavingTime { get => leavingTime; set => leavingTime = value; }

public int EmployeeID { get => employeeID; set { employeeID = value; }}

public int TotalHours { get => totalHours; set => totalHours = value; }

public int TotalMinutes { get => totalMinutes; set => totalMinutes = value; }

public EmployeeMonitoring() { }

public EmployeeMonitoring(DataRow dr)
{
    this.dr = dr;

    this.employeeMonitoringID = Convert.ToInt32(dr["Employee_Monitoring_ID"]);

    this.entranceTime = Convert.ToDateTime(dr["Entrance_Time"]);

    this.leavingTime = Convert.ToDateTime(dr["Leaving_Time"]);

    this.employeeID = Convert.ToInt32(dr["Employee_ID"]);

    this.totalHours = Convert.ToInt32(dr["Total_Hours"]);
}

public void PutInto()
{
    dr["Employee_Monitoring_ID"] = this.employeeMonitoringID;

    dr["Entrance_Time"] = this.entranceTime;

    dr["Leaving_Time"] = this.leavingTime;

    dr["Employee_ID"] = this.employeeID;

    dr["Total_Hours"] = this.TotalHours;
}

EmployeesDB tblEmployees = new EmployeesDB();

public Employee ThisEmployee()
{
    return tblEmployees.GetList().Find(x => x.EmployeeID == this.EmployeeID);
}

public override string ToString()
{

```



```

        return employeeMonitoringID.ToString();
    }
}

```

המחלקה Project

```

public class Project
{
    DataRow dr;

    private int projectID;

    private string projectName;

    private double target;

    private string status;

    ProjectsDB tblProjects = new ProjectsDB();

    public DataRow Dr { get => dr; set => dr = value; }

    public int ProjectID { get => projectID; set => projectID = value; }

    public string ProjectName { get => projectName; set => projectName = value; }

    public double Target { get => target; set => target = value; }

    public string Status { get => status; set => status = value; }

    public Project() { }

    public Project(DataRow dr)
    {
        this.dr = dr;

        this.projectID = Convert.ToInt32(dr["Project_ID"]);

        this.projectName = dr["Project_Name"].ToString();

        this.target = Convert.ToInt32(dr["Target"]);

        this.status = dr["Status1"].ToString();
    }

    public void PutInto()
    {
        dr["Project_ID"] = this.projectID;

        dr["Project_Name"] = this.projectName;
    }
}

```

```

dr["Target"] = this.target;

dr["Status1"] = this.status;

}

public void Active()

{

    foreach (Project item in tblProjects.GetList())

    {

        item.Status = "לא פעיל";

    }

    this.status = "פעיל";

}

public override string ToString()

{

    return projectName;

}

}

```

המחלקה Scedule

```

public class Scedule

{

    DataRow dr;

    private int ID;

    private string day;

    private string shift;

    private string status;

    private int employeeID;

    EmployeesDB tblEmployees = new EmployeesDB();

    public DataRow Dr { get => dr; set => dr = value; }

    public int ID1 { get => ID; set => ID = value; }

    public string Day { get => day; set => day = value; }

    public string Shift { get => shift; set => shift = value; }

```

```
public int EmployeeID { get => employeeID; set => employeeID = value; }
```

```
public string Status { get => status; set => status = value; }
```

```
public Scedule() { }
```

```
public Scedule(DataRow dr)
```

```
{
```

```
    this.dr = dr;
```

```
    this.ID = Convert.ToInt32(dr["ID"]);
```

```
    this.day = dr["Day1"].ToString();
```

```
    this.shift = dr["Shift1"].ToString();
```

```
    this.Status = dr["Status1"].ToString();
```

```
    this.employeeID = Convert.ToInt32(dr["EmployeeID"]);
```

```
}
```

```
public void PutInto()
```

```
{
```

```
    dr["ID"] = this.ID;
```

```
    dr["Day1"] = this.day;
```

```
    dr["Shift1"] = this.shift;
```

```
    dr["Status1"] = this.status;
```

```
    dr["EmployeeID"] = this.employeeID;
```

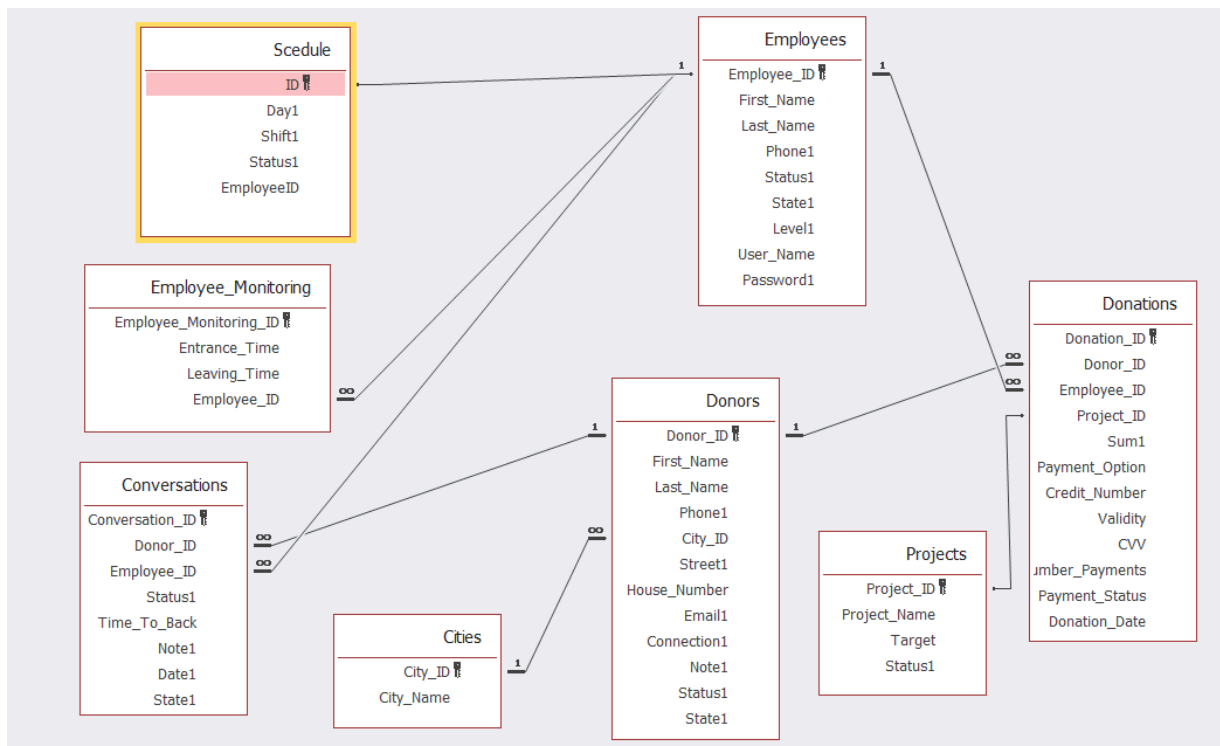
```
}
```

```
public override string ToString()
```

```
{
```

```
    return day+" "+shift+" "+tblEmployees.Find(employeeID).ToString();
```

```
}
```

בסיס נתוניםקשרי גומליןטבלאותטבלת תורמים

שם הטבלה: Donors, תורמים				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	Donor_ID	קוד תורם	מספר	
	First_Name	שם פרטי	טקסט קצר	שם בעברית
	Last_Name	שם משפחה	טקסט קצר	שם בעברית
	Phone1	טלפון	טקסט קצר	מס' פלאפון תקין
	City_ID	קוד עיר	מספר	
	Street1	רחוב	טקסט קצר	שם בעברית
	House_Number	מס' בית	מספר	מספר
	Email1	מייל	טקסט קצר	כתובת מייל תקינה
	Connection1	קשר למוסד	טקסט קצר	

	טקסט ארוך	הערה	Note1	
	טקסט קצר	סטטוס התורם בהתרמה הנוכחית (הותרם/לא הותרם)	Status1	
	טקסט קצר	האם התורם עוד רלוונטי (פעיל/ לא פעיל)	State1	

טבלת עובדים

שם הטבלה: Employees, עובדים				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	Employee_ID	קוד עובד	מספר	
	First_Name	שם פרטי	טקסט קצר	
	Last_Name	שם משפחה	טקסט קצר	
	Phone1	טלפון	טקסט קצר	
	Status1	האם העובד עוד רלוונטי (פעיל/ לא פעיל)	מספר	
	State1	האם כרגע עובדת (מחוברת-לא מחוברת)	טקסט קצר	
	Level1	תפקיד, דרגה	מספר	
	User_Name	שם משתמש	טקסט קצר	
	Password1	סיסמה	טקסט קצר	

טבלת תרומות

שם הטבלה: Donations, תרומות				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	Donation_ID	קוד תרומה	מספר	
	Donor_ID	קוד תורם	מספר	
	Employee_ID	קוד עובד	מספר	
	Project_ID	קוד פרויקט	מספר	
	Sum1	סכום תרומה	מספר	
	Payment_Option	אמצעי תשלום	טקסט קצר	
	Credit_Number	מס' כרטיס אשראי	טקסט קצר	

	תאריך/שעה	תוקף	Validity	
	טקסט קצר	3 ספרות בגב הכרטיס	CVV	
	מספר	מס' תשלומים	Number_Payments	
	טקסט קצר	סטטוס עיסקה	Payment_Status	
	תאריך/שעה	תאריך תרומה	Donation_Date	

בס"ד

טבלת שיחות

שם הטבלה: Conversations, שיחות				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	Conversation_ID	קוד תרומה	מספר	
	Donor_ID	קוד תורם	מספר	
	Employee_ID	קוד עובד	מספר	
	Status1	סטטוס שיחה (לא ענה, לא מעונין, השהיה, תרם)	טקסט קצר	
	Time_To_Back	שעת חזרה (במקרה של השהיה)	תאריך/שעה	
	Note1	הערה	טקסט קצר	
	Date1	תאריך	תאריך/שעה	
	State1	פעיל/לא פעיל	טקסט קצר	

טבלת ערים

שם הטבלה: Cities, ערים				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	City_ID	קוד עיר	מספר	
	City_Name	שם עיר	טקסט קצר	שם בעברית

טבלת פרויקטים

שם הטבלה: Projects, פרויקטים

	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	Project_ID	קוד פרויקט	מספר	
	Project_Name	שם פרויקט	טקסט קצר	
	Target	סכום יעד	מספר	
	Status1	סטטוס פרויקט (פעיל/לא פעיל)	טקסט קצר	

טבלת מעקב עובדים

שם הטבלה: Employee_Monitoring, מעקב עובדים				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	Employee_Monitoring_ID	קוד מעקב עובד	מספר	
	Entrance_Time	זמן כניסה	תאריך/שעה	
	Leaving_Time	זמן יציאה	תאריך/שעה	
	Employee_ID	קוד עובד	מספר	
	Status1	סטטוס מעקב	טקסט קצר	

טבלת לוח זמנים

שם הטבלה: Scedule, לוח זמנים				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח ראשי	ID	קוד לוח	מספר	
	Day1	יום	טקסט קצר	
	Shift1	משמרת	טקסט קצר	
	Status1	סטטוס	טקסט קצר	
	Employee_ID	קוד עובד	מספר	

מדריך למשתמש

ברוך הבא לתוכנת 'טלמרקטינג'.

תוכנה זו באה לעזור לארגונים בניהול פרויקטים של התרמות.

בכל שלב המנהל יכול לעקוב אחר תהליכי ההתרמות, החל מרשימות התורמים, תרומות ועד ניהול העובדים ושעות עבודתם. הכניסה לתוכנה נתונה רק למשתמשים רשומים - ההנהלה והטלפניות ועפ"י רמת המשתמש מתאימה לו את הקטגוריות בהן מורשה לצפות. כדי להשתמש נכון בתוכנה עליך להכיר את הממשק הגרפי ולקרוא את ההוראות הבאות. מאחלים לך בהצלחה ובהנאה!

מסך הכניסה

זהו מסך הכניסה לתוכנה, יש להזדהות באמצעות שם משתמש וסיסמה וללחוץ על "כניסה".

מסך הפתיחה

מכאן ניתן לגשת לכל הטפסים העיקריים בפרויקט: תורמים, תרומות, קבלת שיחה, נתונים נוספים, שיחות, פרויקטים, עובדים, מערכת שעות.



טופס תורמים

- בטבלה מוצגים כל התורמים הנמצאים במאגר התורמים.
- חיפוש: ניתן לכתב ב"חפש תורם" כל פרט הקשור לתורם (כמו שם פרטי, שם משפחה, וכדו') והנתונים יוצגו עפ"י הפרטים שנכתבו.
- מיון: ניתן לבחור ב"מיון לפי" -עפ"י איזה קטגוריה התורמים ימוינו.
- סינון: ניתן לבחור ב"סנן לפי" -איזה תורמים יופיעו בטבלה עפ"י הקטריונים שבחרת.
- הוספת תורם: ניתן להוסיף תורם כך: הוספת תורם -> מילוי פרטי תורם -> לחיצה על שמור.
- עריכת תורם: ניתן לערוך את פרטי התורם כך: בחירת התורם הרצוי בטבלה -> לחיצה על עריכת תורם -> שינוי הפרטים הרצויים -> לחיצה על שמור.
- הצגת תורם: ניתן להציג את פרטי התורם כך:בחירת התורם הרצוי בטבלה -> לחיצה על הצגת תורם.
- חיוג לתורם: ניתן לחייג לתורם כך: בחירת התורם הרצוי בטבלה -> לחיצה על חייג לתורם -> לאחר השיחה עם התורם עדכן סטטוס שיחה עפ"י לחיצה על המקשים הרצויים (ניתן להוסיף הערה בשדה המתאים. (במידה והשיחה מושהית למועד מאוחר יותר: לחיצה על השהיה -> בחירת זמן חזרה -> לחיצה על אישור)

telemarketing חכמה לניהול תרומות

תהילה הויזמן מנהל

בית

תורמים

תרומות

קבלת שיחה

נתונים נוספים

שיחות

פרויקטים

עובדים

מערכת שעות

סנן לפי: מיון לפי:

קוד	שם פרטי	שם משפחה	עיר	רחוב	מספר בית	טלפון
1	חיים	מנדלוביץ	בת ים	רשי	2	76205
2	משה	כהן	חדרה	הנרקיסים	3	89563
3	לאה	לוי	גדרה	יהודה הנשיא	12	25632
4	מיכל	שמש	עפולה	בן זכאי	2	26523
5	נועה	בוטרמן	לוד	הרימון	30	89652
6	יוסף	זכריה	אשדוד	אבטליון	23	25632
7	שפרה	קרביץ	אלעד	רשי	3	62132
8	רבקה	פפא	זכרון יעקב	הלל	12	85621
9	מרדכי	מרקוס	גדרה	הזית	5	86523
11	רחלי	ברסליוער	אלעד	רשי	28	86921
12	אילה	סילבר	אלעד	הגליל	3	76204
13	רבקי	ליבוניץ	בת ים	הכלנית	23	66325
14	מאיר	לוי	בני ברק	אבטליון	15	24632
15	יעל	דזיזון	בית שמש	רשי	10	82563
16	משה	בנדיקט	בני ברק	הרקפת	5	82563

הוספת תורם

עריכת תורם

הצגת תורם

חייג לתורם

8 שיחות ממתינות לך

טופס תרומות

- בטבלה מוצגות כל התרומות הנמצאות במאגר התרומות.
- חיפוש: ניתן לכתב ב"חפש תרומה" כל פרט הקשור לתרומה (כמו תורם, סכום תרומה, וכדו') והנתונים יוצגו עפ"י הפרטים שנכתבו.
- מיון: ניתן לבחור ב"מיון לפי" -עפ"י איזה קטגוריה התרומות ימוינו.
- סינון: ניתן לבחור ב"סנן לפי" -איזה תרומות יופיעו בטבלה עפ"י הקריטריונים שבחרת.
- הוספת תרומה: ניתן להוסיף תרומה כך: הוספת תרומה -> מילוי פרטי תרומה -> לחיצה על שמור.
- עריכת תרומה: ניתן לערוך את פרטי התרומה כך: בחירת התרומה הרצויה בטבלה -> לחיצה על עריכת תרומה -> שינוי הפרטים הרצויים -> לחיצה על שמור.
- הצגת תרומה: ניתן להציג את פרטי התרומה כך:בחירת התרומה הרצויה בטבלה -> לחיצה על הצגת תרומה.

telemarketing חכנה לניהול תרומות

תהילה הויזמן מנהל

תרומות

סנן לפי: מיון לפי:

קוד	תורם	טלפנית	פרויקט	סכום	אמצעי_תש...	תאריך
1	רבקה פפא	תהילה הויזמן	מגבית פורים	400	מזומן	03/2022
3	מאיר לוי	תהילה הויזמן	מגבית פורים	500	אשראי	03/2022
4	חיים מנדלוביץ	תהילה הויזמן	מגבית פורים	100	צ'ק	03/2022
5	לאה לוי	מיכל פרקש	מגבית פורים	1500	צ'ק	03/2022
6	נועה בוטרמן	תהילה הויזמן	מגבית פורים	500	מזומן	03/2022
7	יוסף זכריה	גילי מלצר	מגבית פורים	200	מזומן	03/2022
8	יוסף זכריה	ציפי בן דניאל	מגבית פורים	200	צ'ק	03/2022
9	מרדכי מרקוס	תהילה הויזמן	מגבית פורים	20	מזומן	03/2022
10	מרים פכטהל...	תהילה הויזמן	מגבית פורים	100	מזומן	03/2022
11	רחלי ברסלוי...	תהילה הויזמן	מגבית פורים	400	מזומן	03/2022
12	אילה סילבר	תהילה הויזמן	מגבית פורים	1000	מזומן	03/2022
13	לאה לוי	תהילה הויזמן	מגבית פורים	100	צ'ק	05/2022

חפש תרומה...

תרומה חדשה

עריכת תרומה

הצגת תרומה

8 שיחות ממתינות לך

טופס קבלת שיחה

- טופס זה שולף תורם בצורה אוטומטית ומציג את פרטיו.
- לאחר השיחה עם התורם יש לבחור בסטטוס המתאים:
- -במידה ולא ענה לחץ על "לא ענה".
- -במידה וסירב לחץ על "סירב".
- -במידה ורוצה לבצע תרומה: לחץ על פתיחת תרומה -> מלא פרטים -> לחץ שמור.
- -במידה וביקש לחזור שנית מאוחר יותר: לחץ השהיה -> בחר שעת חזרה -> לחץ אישור.

בכל מצב ניתן להכניס הערה בשדה המתאים

The screenshot displays the 'חיג לתורם' (Call to Donor) screen. The interface includes a sidebar with navigation options: בית (Home), תורמים (Donors), תרומות (Donations), קבלת שיחה (Call Reception), נתונים נוספים (Additional Information), שיחות (Calls), פרויקטים (Projects), עובדים (Employees), and מערכת שעות (Time System). The main content area shows a call log entry for a donor named 'יוסף זכריה' (Yosef Zacharia) with phone number '039025632'. The status is 'פעיל' (Active). Below the call log, there are four buttons: 'פתיחת תרומה' (Open Donation), 'סירב' (Refused), 'לא ענה' (No Answer), and 'השהיה' (Hold). A text field labeled 'הוספת הערה' (Add Note) is also present. At the bottom right, there is a status bar indicating '8 שיחות ממתינות לך' (8 calls waiting for you).

- בתחתית העמוד מופיע מספר השיחות הממתינות שכבר הגיע הזמן לחזור אליהם, בלחיצה על החץ הסגול ליד תיפתח טבלת השיחות הממתינות

טופס נתונים נוספים

טופס זה מציג נתונים כללים בנוגע לפרויקט הגיוס הנוכחי וכן פרטים אישיים הספציפיים למשתמש זה.

טופס שיחות

- בטבלה מוצגות כל השיחות הנמצאות במאגר השיחות.
- מיון: ניתן לבחור ב"מיון לפי" - עפ"י איזה קטגוריה השיחות ימוינו.
- סינון: ניתן לבחור ב"סנן לפי" - איזה שיחות יופיעו בטבלה עפ"י הקטריונים שבחרת.

The dashboard shows a list of calls with the following columns: קוד, תורם, טלפנית, סטטוס, זמן_חזרה, הערה, תאריך, מצב. The table is filtered by 'מיון לפי:' and 'סנן לפי:'.

קוד	תורם	טלפנית	סטטוס	זמן_חזרה	הערה	תאריך	מצב
5	רבקה פפא	תהילה הויזמן	סירב	30/12/1899		29/04/2022	לא פעיל
9	מרדכי מרקוס	תהילה הויזמן	סירב	30/12/1899		03/05/2022	לא פעיל
24	אילה סילבר	תהילה הויזמן	סירב	30/12/1899		09/05/2022	לא פעיל
44	לאה לוי	תהילה הויזמן	תרם	30/12/1899		10/05/2022	לא פעיל
47	יוסף זכריה	תהילה הויזמן	תרם	30/12/1899		24/05/2022	לא פעיל
48	יוסף זכריה	תהילה הויזמן	תרם	30/12/1899		24/05/2022	לא פעיל
2	מרדכי מרקוס	תהילה הויזמן	השחיה	27/04/2022		28/04/2022	לא פעיל
1	רבקה פפא	תהילה הויזמן	לא ענה	28/04/2022		28/04/2022	לא פעיל
3	רבקה פפא	תהילה הויזמן	לא ענה	28/04/2022		28/04/2022	לא פעיל
4	רבקה פפא	תהילה הויזמן	לא ענה	28/04/2022		28/04/2022	לא פעיל
8	אילה סילבר	תהילה הויזמן	השחיה	30/04/2022		30/04/2022	לא פעיל
6	מרדכי מרקוס	תהילה הויזמן	לא ענה	01/05/2022		30/04/2022	לא פעיל
7	רחלי ברסלוי	תהילה הויזמן	לא ענה	01/05/2022		30/04/2022	לא פעיל
12	רבקי ליבוביץ	תהילה הויזמן	השחיה	03/05/2022		03/05/2022	לא פעיל

טופס פרויקטים

- בטבלה מוצגים כל הפרויקטים הנמצאים במאגר הפרויקטים.
- חיפוש: ניתן לכתב ב"חפש פרויקט" כל פרט הקשור לפרויקט (כמו שם פרויקט, סכום יעד, וכדו') והנתונים יוצגו עפ"י הפרטים שנכתבו.
- הוספת פרויקט: ניתן להוסיף פרויקט כך: הוספת פרויקט -> מילוי פרטי פרויקט -> לחיצה על שמור.
- עריכת פרויקט: ניתן לערוך את פרטי הפרויקט כך: בחירת הפרויקט הרצוי בטבלה -> לחיצה על עריכת פרויקט -> שינוי הפרטים הרצויים -> לחיצה על שמור.

The screenshot displays the Telemarketing system interface. On the left is a sidebar with navigation icons for: בית (Home), תורמים (Donors), תרומות (Donations), קבלת שיחה (Accept Call), נתונים נוספים (More Info), שיחות (Calls), פרויקטים (Projects), עובדים (Employees), and מערכת שעות (Time System). The main area is titled 'פרויקטים' (Projects) and contains a table with the following data:

קוד פרויקט	שם פרויקט	סכום יעד	סטטוס
1	מגבית פורים	100000	פעיל
2	מגבית פסח ...	50000	לא פעיל
3	מגבית שבוע...	10000	לא פעיל
4	מגבית ימים ...	80000	לא פעיל

On the right side of the table, there are two buttons: 'הוספת פרויקט' (Add Project) and 'עריכת פרויקט' (Edit Project). At the bottom left of the main area is a 'סגור' (Close) button. The top right corner shows the user's name 'תהילה הויזמן' and a profile icon. The bottom right corner indicates '7 שיחות ממתינות לך' (7 calls waiting for you).

טופס עובדים

- בטבלה מוצגים כל העובדים הנמצאים במאגר העובדים.
- חיפוש: ניתן לכתוב ב"חפש עובד" כל פרט הקשור לעובד (כמו שם פרטי, טלפון, וכדו') והנתונים יוצגו עפ"י הפרטים שנכתבו.
- הוספת עובד: ניתן להוסיף עובד כך: הוספת עובד -> מילוי פרטי עובד -> לחיצה על שמור.
- עריכת עובד: ניתן לערוך את פרטי העובד כך: בחירת העובד הרצוי בטבלה -> לחיצה על עריכת עובד -> שינוי הפרטים הרצויים -> לחיצה על שמור.
- הצגת עובד: ניתן להציג את פרטי העובד כך: בחירת העובד הרצוי בטבלה -> לחיצה על הצגת עובד.
- מיון: ניתן לבחור ב"מיון לפי" - עפ"י איזה קטגוריה העובדים ימוינו.
- סינון: ניתן לבחור ב"סנן לפי" - איזה עובדים יופיעו בטבלה עפ"י הקטריריונים שבחרת.

telemarketing
תהילה הויזמן
מנהל

בית

תורמים

תרומות

קבלת שיחה

נתונים נוספים

שיחות

פרויקטים

עובדים

מערכת שעות

סנן לפי:

מיון לפי:

עובדים

קוד	שם פרטי	שם משפחה	טלפון	שם משתמש	סיסמה	דרגה
1	תהילה	הויזמן	0556790601	tehila0601	Aa12345678	מנהל
2	ציפי	בן דניאל	0533265235	tzipi	5235	טלפנית
3	גילי	מלצר	0554652365	gili	2365	טלפנית
4	מיכל	פרקש	0526565321	michal	5321	טלפנית
7	מלי	לוי	0554547145	mali	4589	טלפנית
9	אורית	פוריים	0542368798	orit	2478	טלפנית
10	ברכי	אפשטיין	0526896532	brachi6532	4502	טלפנית
11	לאה	מנדלבוים	0556325896	lea5896	8588	טלפנית
12	יעל	מיצמר	0548478565	yael	6656	טלפנית
13	שירי	כהן	0542314125	shiri	3778	טלפנית
14	מיטל	ישראלי	0523657898	meital	3698	טלפנית
15	דבורה	שטרן	0556768847	dvorit8847	8847	מנהל

חוספת עובד

עריכת עובד

הצגת עובד

7 שיחות מחתימות לך

טופס מערכת שעות

- טופס זה נועד לנהל את משמרות העובדים
- כל משבצת מסמנת משמרת
- על מנת להוסיף עובדים למשמרת : לחץ על ריבוע המשמרת הרצויה -> תועבר לטופס לבחירת עובדים-> בחר בטלפנית הרצויה והשתמש בחיצים להעבירה למשמרת->בסיום לחץ "שמור".
- לריקון הלוח: לחץ על "רוקן לוח"

telemarketing חכנה לניהול תרומות

תהילה הויזמן מנהל

רוקן לוח

	ראשון	שני	שלישי	רביעי	חמישי
משמרת בוקר	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו
משמרת צהריים	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו
משמרת ערב	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	בחר עובדים למשמרת זו	מיטל ישראלי	בחר עובדים למשמרת זו

7 שיחות ממתינות לך

בית תורמים תרומות קבלת שיחה נתונים נוספים שיחות פרויקטים עובדים מערכת שעות

טופס מעקב עובדים:

- בטבלה מוצגים מעקב פעילות העובדים.
- חיפוש: ניתן לכתב ב"חיפוש" כל פרט הקשור למעקב פעילות העובד (כמו שם עובד, תאריך, וכדו') והנתונים יוצגו עפ"י הפרטים שנכתבו.
- מיון: ניתן לבחור ב"מיון לפי" - עפ"י איזה קטגוריה פרטי הפעילות ימוינו.

telemarketing
הכנה לניהול תרומות

תהילה הויזמן
מנהל

בית

תורמים

תרומות

קבלת שיחה

נתונים נוספים

שיחות

פרויקטים

עובדים

מערכת שעות

מעקב עובדים

מיון לפי:

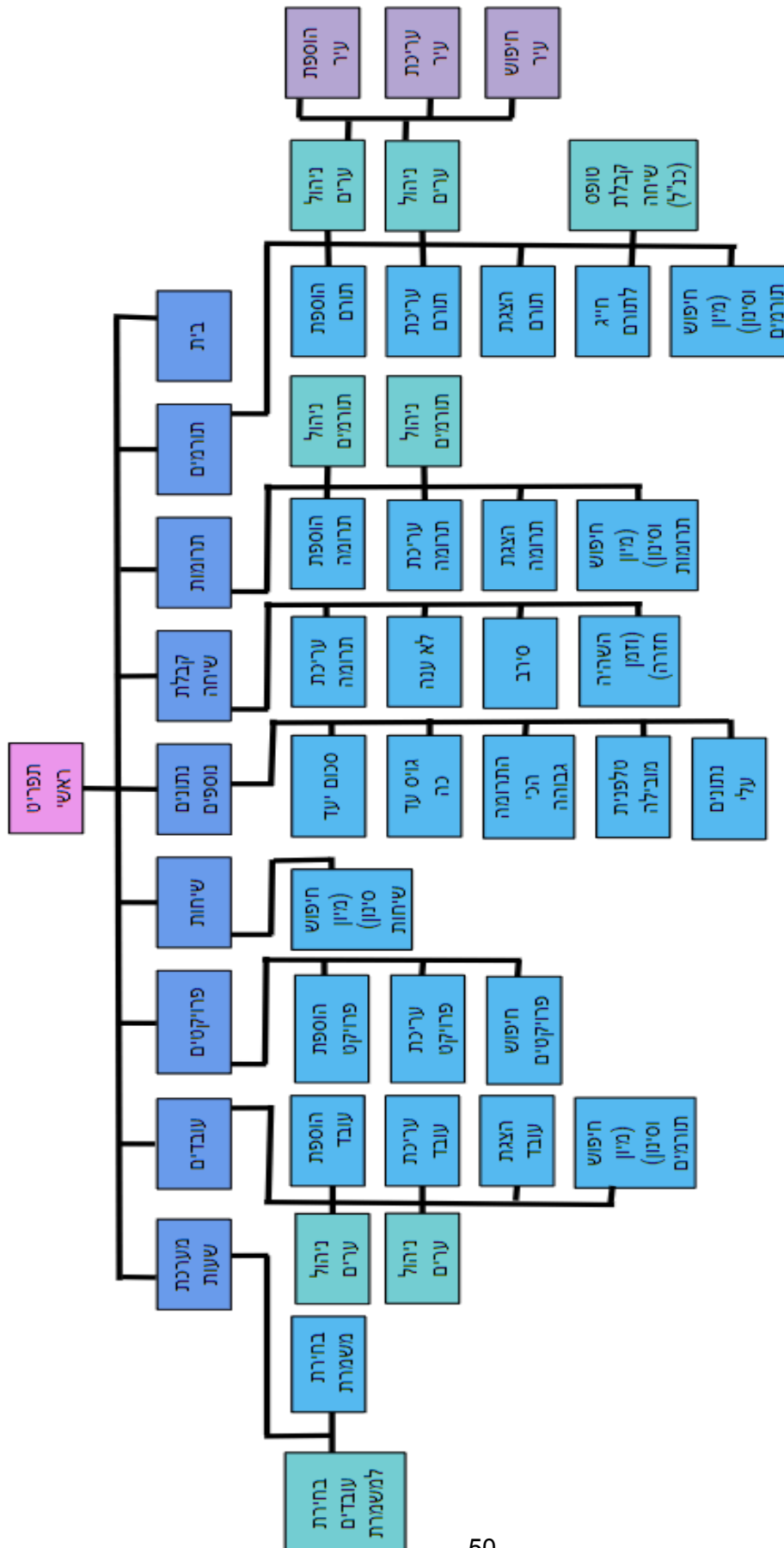
מעקב עובדים

חיפוש...

קוד	שם עובד	שעת כניסה	שעת יציאה	תאריך
1	תהילה הויזמן	14:33:56	00:00:00	09/05/2022
2	תהילה הויזמן	19:46:32	00:00:00	09/05/2022
3	תהילה הויזמן	19:51:37	00:00:00	09/05/2022
4	תהילה הויזמן	20:14:01	00:00:00	09/05/2022
5	תהילה הויזמן	20:18:24	00:00:00	09/05/2022
6	תהילה הויזמן	20:23:43	00:00:00	09/05/2022
7	תהילה הויזמן	20:35:30	00:00:00	09/05/2022
8	תהילה הויזמן	20:38:19	00:00:00	09/05/2022
9	תהילה הויזמן	20:39:36	00:00:00	09/05/2022
10	תהילה הויזמן	20:42:07	00:00:00	09/05/2022
11	תהילה הויזמן	20:46:24	00:00:00	09/05/2022
12	תהילה הויזמן	20:49:45	00:00:00	09/05/2022
13	תהילה הויזמן	20:51:16	00:00:00	09/05/2022
14	תהילה הויזמן	20:59:55	00:00:00	09/05/2022

2 שיחות ממתנות לך

עץ תפריטים



רפלקציה

העבודה על הפרויקט הייתה מאד מאתגרת בשבילי, נהניתי לפתור באגים זה נתן סיפוק להצליח ולהתקדם בסייעתא דשמיא היא גם נתנה לי כלים איך לפטור בעיות בצורה נכונה ומהירה בלי להיבהל וכן לעבוד עבודה עצמית מול המחשב בסבלנות וברציפות וכלים אלו בעז"ה יעזרו לי בהמשך כי בשונה מהלמודים הייתה כאן עבודה עצמית רבה שזה התרגול הכי טוב לדעתי.

קושי שהיה לי זה העבודה העצמית, היה לי חסר עבודת הצוות כל אחת יש לה נושא אחר וקשה מאד להיכנס לפרויקט אחר ולהבין את הפרטים שלו, ואף למורה קשה יותר לאחוז ראש בפרויקטים השונים.

אם עכשיו הייתי מתחילה את הפרויקט הייתי מקפידה יותר על הסדר, כמו כתיבת הערות מפורטות, שמירה מסודרת של הפרויקט (בפרויקט זה שמרתי גירסאות שונות של הפרויקט וזה הקשה וגרם לבילבול רב במציאת הגרסה העדכנית).

מסקנותיי והמלצותיי לאלה שבאות אחרי הם להקפיד במהלך כל הפרויקט על סדר ואירגון, לכתוב מפת עבודה מסודרת, מטלות שנותרו לסיום הפרויקט ושמירה מסודרת של הקבצים.

אני מודה מאד לכל העוסקים במלאכה תודה רבה ובהצלחה לכולם!