**Reliability Test**

תיעוד פרויקט גמר בהנדסת תוכנה מערכת מבחני אמינות

מגיש: אורי בורג

בית הספר: ישיבת בני עקיבא גבעת שמואל

שם המורה: גמליאל זאבי

תאריך: אפריל 2023

**תוכן עניינים**

[מסמך יזום**......................................................................................................................4**](#מסמךיזום)

**[מבנה הארגון...................................................................................................................4](#מסמךיזום)**

**[מערכת המידע הקיימת......................................................................................................4](#מסמךיזום)**

**[תרשים DFD-0 לתיאור המערכת הקיימת..............................................................................5](#תרשיםDFD0למערכתקיימת)**

**[בעיות במערכת המידע הקיימת](#בעיותבמערכתהקיימת)**[...........................................................................................](#בעיותבמערכתהקיימת)**[6](#בעיותבמערכתהקיימת)**

[מבוא](#מבוא)**[.............................................................................................................................7](#מבוא)**

**[רקע על המערכת..............................................................................................................7](#מבוא)**

**[פרטים טכניים..................................................................................................................7](#מבוא)**

**[הוראות התקנה................................................................................................................7](#מבוא)**

**[הסיבות לבחירת הנושא.....................................................................................................7](#מבוא)**

[תיאור המערכת](#תיאור)**[...............................................................................................................8](#תיאור)**

**[תיאור כללי......................................................................................................................8](#תיאורמערכת)**

**[תרשים DFD-0 לתיאור המערכת החדשה.............................................................................9](#תרשיםDFD0למערכתהחדשה)**

**[תהליכי המערכת............................................................................................................10](#תהליכיהמערכת)**

[**תהליכי ניהול המידע במערכת...........................................................................................10**](#תהליכיהמערכת)

[**תהליכי גישה נוחה למידע במערכת....................................................................................14**](#תהליכיגישהנוחהלמידע)

[**שאר התהליכים..............................................................................................................20**](#שארהתהליכים)

[**בעיות וחסרונות במערכת.................................................................................................21**](#בעיותוחסרונותבמערכת)

[מסד הנתונים**................................................................................................................22**](#מסד)

[**טבלאות המערכת ושדותיהן.............................................................................................22**](#מסד)

[**תרשים ERD-0 של מסד הנתונים ......................................................................................23**](#תרשיםERDשלמסדהנתונים)

[**תרשים קשרי גומלין........................................................................................................24**](#תרשיםקשריגומלין)

[**פירוט הטבלאות ושדותיהן................................................................................................25**](#פירוטטבלאותושדותיהן)

[מדריך למשתמש**...........................................................................................................27**](#מדריך)

[**התחברות למערכת.........................................................................................................27**](#מדריך)

[**התפריט הראשי.............................................................................................................29**](#התפריטהראשי)

[**טבלאות........................................................................................................................31**](#טבלאות)

[**הוספה..........................................................................................................................35**](#הוספה)

[**עדכון............................................................................................................................39**](#עדכון)

[**דוחות...........................................................................................................................42**](#דוחות)

[**תרשימים......................................................................................................................44**](#תרשימים)

[**חיפוש..........................................................................................................................46**](#חיפוש)

[**יצירת מבחן...................................................................................................................50**](#יצירתמבחן)

[**התחלת מבחן................................................................................................................53**](#התחלתמבחן)

[**ביצוע מבחן...................................................................................................................54**](#ביצועמבחן)

[הקוד במערכת**..............................................................................................................56**](#תיאורקוד)

[טפסים דומים**................................................................................................................56**](#תיאורקוד)

[**טפסי הטבלאות..............................................................................................................56**](#תיאורקוד)

[**טפסי הוספה..................................................................................................................57**](#ט_הוספה)

[**טפסי העדכון.................................................................................................................60**](#ט_עדכון)

[**טפסי הדוחות.................................................................................................................63**](#ט_דוחות)

[**טפסי התרשימים...........................................................................................................65**](#ט_תרשימים)

[**טפסי החיפוש................................................................................................................66**](#ט_חיפוש)

[טפסים ייחודים**..............................................................................................................67**](#ט_התחברות)

[**טופס ההתחברות...........................................................................................................67**](#ט_התחברות)

[**טופס התפריט הראשי.....................................................................................................68**](#ט_תפריטראשי)

[**טופס יצירת מבחן...........................................................................................................69**](#ט_יצירתמבחן)

[**טופס התחלת מבחן........................................................................................................70**](#ט_התחלתמבחן)

[**טופס ביצוע מבחן...........................................................................................................71**](#ט_ביצועמבחן)

[קוד המערכת................................................................................................................72](#קודהמערכת)

**[קוד טופס הוספת תשובות במבחן..................................................................................................72](#קודהמערכת)**

[**קוד טופס הוספת תכונות..............................................................................................................83**](#א)

[**קוד טופס הוספת תכונות בטופס....................................................................................................86**](#ב)

[**קוד טופס הוספת ערים................................................................................................................91**](#ג)

[**קוד טופס הוספת טפסים..............................................................................................................94**](#ד)

**[קוד טופס הוספת שאלות..............................................................................................................98](#ה)**

[**קוד טופס הוספת שאלות לתכונה.................................................................................................101**](#ו)

[**קוד טופס הוספת שאלות בטופס..................................................................................................109**](#ז)

[**קוד טופס הוספת מבחנים..........................................................................................................117**](#ח)

[**קוד טופס הוספת משתמשים......................................................................................................122**](#ט)

[**קוד טופס תרשים שאלות לתכונות...............................................................................................128**](#אא)

[**קוד טופס תרשים שאלות למבחנים..............................................................................................133**](#בב)

[**קוד טופס תרשים שאלות לטפסי בחינה........................................................................................138**](#גג)

[**קוד טופס התחברות..................................................................................................................143**](#דד)

[**קוד טופס תפריט......................................................................................................................149**](#הה)

[**קוד טופס דוח שאלות לתכונה.....................................................................................................164**](#וו)

[**קוד טופס דוח מבחנים...............................................................................................................169**](#זז)

[**קוד טופס דוח טפסי מבחן..........................................................................................................181**](#חח)

[**קוד טופס חיפוש תשובות במבחן.................................................................................................189**](#טט)

[**קוד טופס חיפוש תכונות.............................................................................................................192**](#יי)

[**קוד טופס חיפוש תכונות בטופס...................................................................................................195**](#ככ)

**[קוד טופס חיפוש ערים...............................................................................................................198](#ממ)**

[**קוד טופס חיפוש טפסים.............................................................................................................201**](#ננ)

**[קוד טופס חיפוש שאלות.............................................................................................................204](#סס)**

**[קוד טופס חיפוש שאלות לתכונה..................................................................................................207](#עע)**

**[קוד טופס חיפוש שאלות בטופס...................................................................................................210](#פפ)**

**[קוד טופס חיפוש מבחנים...........................................................................................................213](#צצ)**

**[קוד טופס חיפוש משתמשים.......................................................................................................216](#קק)**

[**קוד טופס טבלת תשובות במבחן.................................................................................................220**](#אאא)

[**קוד טופס טבלת תכונות.............................................................................................................223**](#בבב)

[**קוד טופס טבלת תכונות בטופס...................................................................................................226**](#גגג)

[**קוד טופס טבלת ערים................................................................................................................229**](#דדד)

[**קוד טופס טבלת טפסים.............................................................................................................232**](#ההה)

[**קוד טופס טבלת שאלות.............................................................................................................235**](#ווו)

[**קוד טופס טבלת שאלות לתכונה.............****.....................................................................................238**](#זזז)

[**קוד טופס טבלת שאלות בטופס...................................................................................................241**](#חחח)

[**קוד טופס טבלת מבחנים............................................................................................................244**](#טטט)

[**קוד טופס טבלת משתמשים........................................................................................................247**](#ייי)

[**קוד טופס יצירת מבחן................................................................................................................250**](#כככ)

[**קוד טופס התחלת מבחן.............................................................................................................262**](#ללל)

[**קוד טופס תהליך ביצוע מבחן......................................................................................................268**](#מממ)

[**קוד טופס עדכון תשובות במבחן..................................................................................................283**](#נננ)

[**קוד טופס עדכון תכונות..............................................................................................................295**](#ססס)

[**קוד טופס עדכון טפסים..............................................................................................................300**](#עעע)

[**קוד טופס עדכון שאלות..............................................................................................................307**](#פפפ)

[**קוד טופס עדכון שאלות לתכונה...................................................................................................313**](#צצצ)

[**קוד טופס עדכון שאלות בטופס....................................................................................................322**](#קקק)

[**קוד טופס עדכון מבחנים.............................................................................................................230**](#ררר)

[**קוד טופס עדכון משתמשים.........................................................................................................239**](#ששש)

**מסמך יזום**

מבנה הארגון:

התוכנה מיועדת לבעלי חברות עם עובדים, מנהלי גיוס של חברות ואנשים בכללי הרוצים לבדוק לצרכים שלהם התאמה של אנשים לתכונות מסוימיות ובדיקת הדומננטיות של תכונה/תכונות מסוימות אצל אנשים אחרים.

התוכנה מאפשרת למנהלי התוכנה ליצור בקלות מבחני אמינות לפי תכונות הניתנות לבחירתם, ונותנת גם למשתמשים האחרים בתוכנה לבצע את המבחנים שנוצרו ולבדוק את התאמתם לתכונות הנבדקות במבחן בעזרת ציונים ודוחות הנוצרים במערכת הניתנים לצפיה.

בכך המערכת מחולקת ל2 סוגי משתמשים- המשתמשים המנהלים היכולים גם לנהל את המידע במערכת וגם ליצור ולבצע מבחנים ולהוסיף אופציות חדשות כמו תכונות חדשות או שאלות חדשות, ומשתמשים הלא מנהלים אשר לא יכולים לנהל את המידע הנמצא במערכת אך יכולים ליצור מבחנים ולבצע אותם וגם לגשת לדוחות והתרשימים.

מערכת המידע הקיימת:

כיום מערכת המידע הקיימת היא בעזרת אתרי אינטרנט שונים הקיימים באינטרנט, כל אחד נפרד מהשני ונוצר לבד אך הם עובדים בצורה דומה בדרך כלל, מופיעות אמירות כגוון- "אני מרגיש שמח" והמשתמש צריך לדרג עד כמה הוא מסכים עם האמירה הזו, ולפי כך המבחנים מביאים תשובה שהיא או ציון או אמירה מובנית כגוון- "לפי התוצאות הינך אדם מופנם", במערכות הקיימות כיום לא ניתן ליצור מבחנים משלך ובדרך כלל הם מכילים כל אתר מבחן אחד ספציפי מובנה מראש ללא אפשרות לשינוי.

דוגמאות למבחני אישיות הקיימים כיום-

1) אצל צהל מתקיימים כבר הרבה שנים מבחן הנקרא מבחן 300 שהוא מבחן אישיות הבודק באמצעות שאלות מהירות וקצרות ובאמצעות משפטים עליהם אתה נדרש להגיד עד כמה אתה מסכים עם האמירה, ולפי זה צהל מקבל עליך נתונים אישים שיכולים להציג את האישיות שלך והתכונות שלך, המבחן שם כמו שצינתי מיקודם אינו פתוח לציבור הרחב והתוצאות אינם מפורסמות לציבור או למשתמש העושה את המבחנים.

2) באתר iPersonic קיים מבחן אישיות שבוא אתה אומר לאיזה מספר אמירות אתה יותר מתאים ולפי כך לאיזה סוג אדם אתה מתאים יותר- לדוגמא מוחצן או מופנם, ולאחר שעשית כמה כאלו אתה מקבל תוצאה האומרת לך לאיזה סוג אישיות אתה מתאים מתוך כמה סוגים, לדוגמא החושב היוזם או האידיאליסט המאורס ועוד, במערכת זו אומנם התוצאות אכן ניגשות לציבור אך האתר כמו שצינתי אינו מערכת ואין ניתן ליצור באמצעותו מבחנים אישותים נוספים.

3) ישנם עוד הרבה מבחני אישיות כל אחד עם הסגנון שלו המפורסמים בציבור באינטרנט אך בהם התוצאות אינם תמיד אמינות כי איננו יכולים לדעת מי יצר את המבחן והאם הוא אמין ואיך הוא יצר את המבחן הזה והאם בצורה שאכן נותנת תוצאות לפי תשובותיך או אלגוריתם אמין או סתם "מחרטט" לך תשובות שתרצו לשמוע, או סתם באופן אקראי, וגם בהם הציבור אינו יכול ליצור שאלון אמין משלו לשימוש אישי.

תרשים DFD-0 לתיאור המערכת הקיימת:

* פרטי משתמשים
* בחינות
* תשובות בבחינה
* ציון הבחינה

משתמש

* ציון הבחינה

מנהל אתר

בעיות במערכת המידע הקיימת:

מערכת המידע הקיימת היום למבחנים מסוג זה של אמינות-אישיות ובחינת תכונות אינה נותנת את המידע של איך היא עובדת לציבור כיום, וכל חברה הרוצה מבחן כזה צריכה ליצור אותו בעצמה אחד משלה, מה שגם יביא תוצאות שונות הגם לא בהכרח אמינות, וגם מערכות אלו אינם נותנת לציבור ליצור מבחנים כאלו משלו, דברים אלו יוצרים מספר בעיות:

1) המבחן הזה אינו דומה בין חברות הרוצות לבדוק את אותו דבר משום המערכות השונות, כך חברה הרוצה לקבל נתונים דומים לפי אותו מבחן אינה יכולה וצריכה ליצור מבחן משלה הנותן תוצאות שונות ובודק דברים שונים.

2) המערכת אינה נגישה לציבור הרחב שאינו יכול ליצור לעצמו כל אחד באופן אישי מערכת משלו, וככה אנשים אישים לא יכולים ליצור מבחנים כאלו ולבדוק בעצמם לצורכם האישי במבחן אישיות אנשים אחרים, וכך גם לעסקים קטנים הרוצים לבדוק גם אנשים לצורך העסקתם.

3) המבחנים כעת כרוב נוצרים באופן מיוחד לצורך ספציפי, כלדוגמא מבחן הנוצר במיוחד לבדיקת מועמדים בצבא (מבחן 300) וככה קשה ליצור מבחר מבחנים כאלו לצרכים שונים, ואין מערכת היוצרת מבחנים כאלו אוטמטי בכמה בחירות קטנות, לפחות לא שגלויה מערכת כזו לציבור.

**מבוא**

רקע על המערכת:

המערכת באה ליצור דרך שבה הציבור הרחב יוכל ליצור מבחני אישיות בצורה מופשטת וקלה בלי צורך לתכנת או לעסוק באלגוריתם שמאחורי התוכנה, וככה מנהלי חברות קטנות או גדולות או אישים אחרים יוכלו ליצור מבחן משלהם, לפי הצורך שלהם, למי שהם צריכים ללא צורך בהשכרת מתכנת ליצור להם מבחן חדש מותאם משלהם. כך המערכת נותנת למנהלים להוסיף שאלות ותכונות חדשות למערכת בקלות, ואז ניתן ליצור בפשטות מבחן אמין שאתה מחליט מה יהיה התכונות שהוא יבדוק, מה יהיה האורך שלו, וכמה זמן התשובה שלו משאלה לשאלה יהיה, ולראות את התוצאות על ידי ציון או על ידי דוחות או תרשימים המפרטים יותר מידע.

פרטים טכניים:

מחשב: DESKTOP-PJ6TLVH

מערכת הפעלה: Windows 10 Home

סביבת פיתוח: Microsoft Office Professional Plus

מסד נתונים: Microsoft Office Access

הוראות התקנה:

יש לחלץ את תיקיית המערכת (Project\_OriBurg) אל הכתובת C:\Projects\_2023,להיכנס לתיקיית Release שבתוך תיקיית המערכת, ולהריץ את הקובץ ReliabilityTest ניתן גם ליצור קיצור דרך של קובץ ריצה זה בכל מקום במחשב ולהריץ את המערכת דרך קיצור הדרך.

הסיבות לבחירת הנושא:

בחרתי לעסוק בנושא זה של שאלון אמינות כי רציתי לעשות פרויקט שיעניין אותי ונתקלתי בכמה מקומות במבחן אמינות-אישיות וראיתי שלא ניתן בקלות ליצור אחד כזה אז חשבתי שיהיה מעניין ליצור מערכת כזו שתיתן לכך מענה על ידי כך שיהיה ניתן בקלות ליצור שאלון כזה ולנגיש אותו לאחרים, וחשבתי שיהיה מעניין ליצור לכך אלגוריתם לפי המחשבה שלי ליצירתו באופן אמין בכך שיצור מבחן באופן אקראי, שבכך יוסיף לאמינות שלו, וגם יהיה בו אלמנטים היוסיפו לאמינותו כמו זמן תשובה מהיר וחופשיותו של מנהל המערכת להוסיף שאלות משל עצמו ותכונות משל עצמו לפי בחירתו, רציתי לממש את הרעיון ולכן רציתי ליצור את המערכת הזו העונה על הצורך הציבורי הלא קיים וגם כפרויקט שמעניין אותי.

**תיאור המערכת**

תיאור כללי:

תיאור המערכת: המערכת מאפשרת ליצור מבחני אמינות בקלות, ושומרת את המבחנים שנוצרו לשימוש חוזר על ידי משתמשים הבאים, וביצוע מבחנים אלו בצורה המביאה אמינות לתוצאות המבחן, בנוסף בתוכנה בתור מנהל אפשר בקלות להגיע לכל המידע הנאגר בתוכנה ולנהל אותו בעזרת נגישות אל מאגר המידע בצורה ידנית בעזרת הטבלאות, או בעזרת עדכון או הוספת מידע בצורה מונגשת ופשוטה יותר, וגם ניתן לראות את תוצאות המבחנים או מידע על טפסי המבחן הקיימים בקלות בעזרת דוחות, תרשימים וחיפוש בטבלאות הנמצאים גם הם במערכת, ובאמצעותם ניתן בקלות להבין תוצאות מבחנים, ואת טפסי המבחן שנוצרו במערכת.

מטרת הפרויקט: מטרת המערכת היא ליצור מערכת למבחני האמינות-אישיות הלא קיימת כיום שככה ניתן יהיה ליצור מבחנים בקלות וללא צורך בשום תכנות או ידע בתכנות, המערכת נוצרה כמאוד מונגשת וקלה להבנה ושימוש ובכך יכולה להיות משומשת לציבור הרחב הלא קיבל גישה לכך עד היום, המערכת גם מכילה אלגוריתם ליצירת המבחנים לבד ללא צורך עשייה מיוצר המבחן מלבד הנחיותיו למה יכיל המבחן, אם זה אילו תכונות הוא יבדוק, מה יהיה אורך המבחן, או כמה זמן יהיה לעושה המבחן לענות, ככה ניתן יהיה לחסוך הרבה זמן מיצירת מבחן כזה באופן ידני וחשיבה על כל שאלה מחדש בכל מבחן שייווצר, אלא ככה השאלות כבר ימצאו במאגר שהם נוספו אליו פעם אחת בלבד, ואז יוכלו לשמש לעשרות מבחנים שיוצרו בעתיד. המערכת גם מנגישה מידע רחב מלבד ציון או עובר-לא עובר בלבד, ככה שמנהל התוכנה או גם האיש העושה את המבחן, יכולים לראות את התוצאות של המבחן, להבין במה המועמד או המשתמש יותר חזק ופחות חזק, והאם הוא מתאים לתכונה הרצויה לחברה, או התכונה שהמשתמש עצמו רצה לבדוק על עצמו.

אוכלוסיית היעד: אוכלוסיית היעד למערכת זו היא כמה סוגי אנשים- בעלי חברות ואנשי הגיוס שלהם הרוצים ליצור שאלון אמינות בשביל לראות ביותר קלות ואמינות מי המועמדים שלהם שהם מקבלים, הם יכולים להסתמך רק על מבחן זה או להשתמש בו כעזר כחלק משלבי קבלה וראיון אישי, ולראות האם הנתונים המתקבלים משלבי הקבלה האחרים מתאימים גם לתוצאות מהשאלון האמינות, עוד אוכלוסייה היא הציבור הרחב הלא צריך ניסיון מיוחד בשביל תפעול המערכת, ויכול ליצור מבחן לכל מטרה שיצטרך, אם זה עסקים קטנים, מורי בית ספר הרוצים להכיר יותר את תלמידיהם, או סתם עשיית מבחן אישי לכיף להכרה עצמית יותר טובה.

תרשים DFD-0 לתיאור המערכת החדשה:

* משתמשים
* ערים
* שאלות
* תכונות
* שאלות לתכונה
* טפסים
* תכונות בטופס
* שאלות בטופס
* מבחנים
* תשובות במבחן
* משתמשים
* שאלות
* תכונות
* שאלות לתכונה
* טפסים
* שאלות בטופס
* מבחנים
* תשובות במבחן

משתמש

* שאלות
* שאלות לתכונה
* משתמשים
* מבחנים
* תכונות בטופס
* שאלות בטופס
* תשובות במבחן
* טפסים
* תכונות
* שאלות לתכונה
* מבחנים
* שאלות לטופס
* תכונות
* שאלות לתכונה
* טפסים
* שאלות
* משתמשים
* מבחנים
* תכונות בטופס
* שאלות בטופס
* תשובות במבחן

תהליכי המערכת:

במערכת ישנם כמה תהליכים שבזכותם ניתן לנהל בקלות את המערכת, וליצור ולבצע מבחנים, ישנם כמה קבוצות של תהליכים במערכת: תהליכי ניהול המידע במערכת, ותהליכי גישה נוחה למידע במערכת ועוד כמה נוספים, אלו הסברים על תהליכי המערכת לפי הקבוצות שתיארתי כעת.

**תהליכי ניהול המידע במערכת:**

תהליכי ניהול המידע במערכת הם תהליכים שמטרתם היא להיות מונגשים למשתמשים עם גישת מנהל שבעזרתם משתמשים אלו יכולים לגשת בקלות ולהוסיף או לעדכן בפשטות נתונים במערכת כמו- הוספת משתמשים, הוספת טפסי בחינה, עדכון מידע על משתמש מערכת קיים, עדכון שאלות בטופס בחינה ועוד.

לכל הטבלאות ישנם תהליכי הוספה אך לא לכולם תהליכי עדכון לפי שישנם טבלאות המורכבים משדות ייחודיים בלבד, אותם לא ניתן לעדכן.

**תהליכי ניהול המידע במערכת מוצגים בתרשימים הבאים:**

תרשים לתהליכי העדכון וההוספה של שחקנים

טבלת ערים

טבלת משתמשים

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליך ההוספה של ערים

טבלת ערים

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של שאלות

טבלת שאלות

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של תכונות

טבלת תכונות

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של שאלות לתכונה

טבלת שאלות לתכונה

טבלת תכונות

טבלת שאלות

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של טפסים

טבלת משתמשים

טבלת טפסים

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליך ההוספה של תכונות בטופס

טבלת תכונות

טבלת טפסים

טבלת תכונות בטופס

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של שאלות בטופס

טבלת שאלות בטופס

טבלת טפסים

טבלת שאלות

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של מבחנים

טבלת מבחנים

טבלת משתמשים

טבלת טפסים

משתמש עם גישת מנהל

משתמש עם גישת מנהל

תרשים לתהליכי העדכון וההוספה של תשובות במבחן

טבלת תשובות במבחן

טבלת מבחנים

טבלת שאלות

משתמש עם גישת מנהל

משתמש עם גישת מנהל

**תהליכי גישה נוחה למידע במערכת:**

תהליכי גישה נוחה למידע במערכת הם תהליכים שמטרתם היא להיות מונגשים לכל המשתמשים במערכת ולהציג מידע שנמצא במערכת בצורה נוחה ומובנת, אם זה כולל תהליכי דוחות המראים בסיכום דברים כמו טפסי מבחן או מבחנים, תהליכי חיפוש הנותנים למשתמש את האופציה לחפש מידע בטבלה בצורה פשוטה, ותהליכי תרשימים המראים באופן ויזואלי מידע על דברים כמו מספר השאלות לטפסים או מספר השאלות לתכונות.

לכל הטבלאות ישנם תהליכי חיפוש ותהליכי דוחות ותרשימים ישנם 3 לכל אחד.

**תהליכי גישה נוחה למידע במערכת מוצגים בתרשימים הבאים:**

תרשים לתהליך החיפוש של משתמשים

טבלת משתמשים

משתמש

משתמש

תרשים לתהליך החיפוש של שאלות

טבלת שאלות

משתמש

משתמש

תרשים לתהליך החיפוש של תכונות

טבלת תכונות

משתמש

משתמש

תרשים לתהליך החיפוש של טפסים

טבלת טפסים

משתמש

משתמש

תרשים לתהליך החיפוש של תשובות במבחן

טבלת תשובות במבחן

משתמש

משתמש

תרשים לתהליך החיפוש של תכונות בטופס

טבלת תכונות בטופס

משתמש

משתמש

תרשים לתהליך החיפוש של ערים

טבלת ערים

משתמש

משתמש

תרשים לתהליך החיפוש של שאלות לתכונה

טבלת שאלות לתכונה

משתמש

משתמש

תרשים לתהליך החיפוש של שאלות לתכונה

טבלת שאלות בטופס

משתמש

משתמש

תרשים לתהליך החיפוש של מבחנים

טבלת מבחנים

משתמש

משתמש

תרשים לדוח שאלות לתכונה

טבלת שאלות

טבלת שאלות לתכונה

משתמש

משתמש

תרשים לדוח מבחנים

טבלת שאלות

טבלת מבחנים

טבלת תכונות בטופס

טבלת שאלות בטופס

טבלת משתמשים

משתמש

משתמש

טבלת שאלות לתכונה

טבלת תשובות במבחן

תרשים לדוח טפסי מבחן

טבלת שאלות

טבלת תכונות לטופס

טבלת שאלות לתכונה

משתמש

משתמש

טבלת שאלות בטופס

טבלת טפסים

תרשים הדיאגרמה שאלות לתכונה

טבלת תכונות

טבלת שאלות לתכונה

משתמש

משתמש

תרשים הדיאגרמה שאלות למבחנים

טבלת שאלות לטופס

טבלת מבחנים

משתמש

משתמש

תרשים הדיאגרמה שאלות לטופסי בחינה

טבלת שאלות לטופס

טבלת מבחנים

משתמש

משתמש

**שאר התהליכים:**

בנוסף לתהליכי גישה נוחה למידע במערכת ותהליכי ניהול המידע במערכת, ישנם עוד מספר תהליכים נוספים המפעילים את המערכת, שהם:

1. **תהליך התחברות:** תהליך הכניסה למערכת על ידי התחברות משתמש עם תעודת זהות וסיסמה שלו, במידה ותעודת הזהות והסיסמה קיימים במערכת המשתמש יתחבר.
2. **תפריט ראשי:** לאחר התחברות למערכת מתחברים אל התפריט הראשי מימנו אפשר להגיע אל כל תהליכי המערכת האחרים, באמצעות תהליכים הנמצאים בו הפותחים וסוגרים את שאר התהליכים.
3. **תהליך יצירת מבחן:** תהליך בתפריט הבחנים שבעזרתו ניתן ליצור טופס מבחן חדש, לפי קריטריונים לבחירה שהם: תכונות שיבדקו וכמות שאלות לכל תכונה, והתהליך יצור לך טופס מבחן חדש עם שאלות רנדומליות בסדר רנדומלי בהתאם.
4. **תהליך התחלת מבחן:** תהליך בתפריט הבחנים ששם אתה בוחר משתמש ובוחר את טופס המבחן שאותו תרצה לבצע ואת מהירות הבחינה הרצויה, ואז יכול להתחיל מבחן ויוצר לך מבחן חדש במערכת ותגיע אל תהליך ביצוע המבחן.
5. **תהליך ביצוע המבחן:** תהליך שמגיעים אליו מתהליך התחלת המבחן ששם אתה מבצע את המבחן שיצרת בהתחלת המבחן לפי הטופס שבחרת, מוצג בו פרטי המשתמש הנבחר ובוא תצטרך לענות לפי המהירות הנבחרת על השאלות בטופס המבחן בעזרת סימוני המספרים לפי שנגמר הזמן, בסוף המבחן תקבל את תוצאת המבחן שהגעת אליה ופרטי המבחן עם התשובות שביצעת ישמרו במערכת.

בעיות וחסרונות במערכת:

למערכת ישנם גם מספר חסרונות, והם:

1. המערכת אינה חסינה לגמרי מנתונים שגוים, יתכן שמשתמש במערכת עם גישה של מנהל בעזרת שינוי מידע שגוי או לא תקין בטבלאות ממסד הנתונים, ובכך יגרום למידע שגוי היכול לגרום לבעיות במערכת או לגרום לה לא לעבוד תקין.
2. יתכן גם הוספת מידע שאינו תקין למערכת שהמערכת אינה בודקת את אמינותו, אם זה בטבלאות או בתהליכי הוספה ועדכון, לדוגמא לעומת תעודת זהות שהמערכת אכן בודקת את אמינותה, איימל של משתמש אינו נבדק על ידי המערכת ומאומת, ולכן יתכן הכנסת איימל לא שלך או מפוברק הלא קיים.
3. במערכת משתמש אומנם יכול בעזרתה ליצור בקלות מבחן חדש לפי התכונות שירצה לבדוק, אורך המבחן, וזמן התשובה הרצוי שלו, אך הוא אינו יכול להחליט באופן מלא על כל דבר במבחן שתיווצר, שלו יש דברים בסיסים הלא ניתנים לשינוי כמו זמן תשובה אחר שהוא לא מהאופציות המובנות או לדוגמא נוספת אורך מבחן יותר גדול מהניתן במערכת כרגע או קטן מאוד הלא ניתן גם במערכת.
4. במערכת ביצוע מבחן חייב להיות באופן ספציפי המובנה במערכת, זהו חיסרון למי שירצה מבחן המובנה בצורה אחרת, לדוגמא עם הפסקות בין כמות שאלות לכמות שאלות מסוימת או לדוגמא נוספת שהתשובה של האם אתה עברת את השאלה תוצג בכל שאלה ושאלה ולא רק אחרי סיום המבחן יהיה ניתן לראות ציון או יותר פרטים בדוחות.

**מסד הנתונים**

הטבלאות ושדותיהן:

מסד הנתונים מכיל 10 טבלאות ואלו הן:

1. **טבלת משתמשים:** טבלה השומרת את פרטי המשתמשים במערכת, שדותיה הם: תעודת זהות, שם פרטי, שם משפחה, תאריך לידה, כתובת, עיר, טלפון, מייל, סיסמה, מנהל (כן או לא) ותמונה.
2. **טבלת ערים:** טבלה המכילה את כל הערים הנמצאים במערכת וניתנים לבחירה למשתמשים, לה רק שדה אחד והוא שם העיר.
3. **טבלת שאלות:** טבלה המכילה את כל השאלות במערכת הניתנות לשימוש במבחנים, לה 2 שדות והם מספר השאלה וטקסט השאלה.
4. **טבלת תכונות:** טבלה המכילה את כל התכונות במערכת הניתנות לבדיקה במבחנים, לה 2 שדות והם שם התכונה ותיאור התכונה.
5. **טבלת שאלות לתכונות:** טבלה המכילה את כל השאלות במערכת ואומרת איזו תכונה כל שאלה בודקת וכמה צריך להיות בניקוד בשביל להתאים לתכונה בתשובה לשאלה, שדותיה הם: שם התכונה, מספר שאלה, ערך התחלה וערך סיום.
6. **טבלת טפסים:** טבלה המכילה את כל טפסי המבחן הקיימים במערכת, שדותיה הם: מספר הטופס, שם המבחן, תאריך יצירה ות"ז יוצר.
7. **טבלת תכונות בטפסים:** טבלה המכילה את כל הטפסים ואומרת איזו תכונות כל טופס בודק, לה 2 שדות והם מספר הטופס ושם התכונה.
8. **טבלת שאלות בטופס:** טבלה המכילה את כל הטפסים והשאלות שבהם ואומרת מה הסדר הצגה שלהם בזמן מבחן, שדותיה הם: מספר הטופס, מספר סידורי של שאלה ומספר השאלה.
9. **טבלת מבחנים:** טבלה המכילה את כל המבחנים שבוצעו במערכת, שדותיה הם: מספר המבחן, מספר הטופס, ת"ז נבחן ותאריך הבחינה.
10. **טבלת תשובות במבחנים:** טבלה המכילה את כל המבחנים שבוצו את התשובות שהנבחן עשה לכל שאלה, שדותיה הם: מספר המבחן, מספר סידורי של שאלה והתשובה.

תרשים ERD-0 של מסד הנתונים:

שאלות

ערים

מותאמות ב

מותאמות ב

גר ב

שאלות לתכונות

שאלות בטופס

משתמשים

מופיעות ב

מופיעות ב

מבחן של

מופע של

מבחנים

טפסים

מופיעות ב

תשובות על

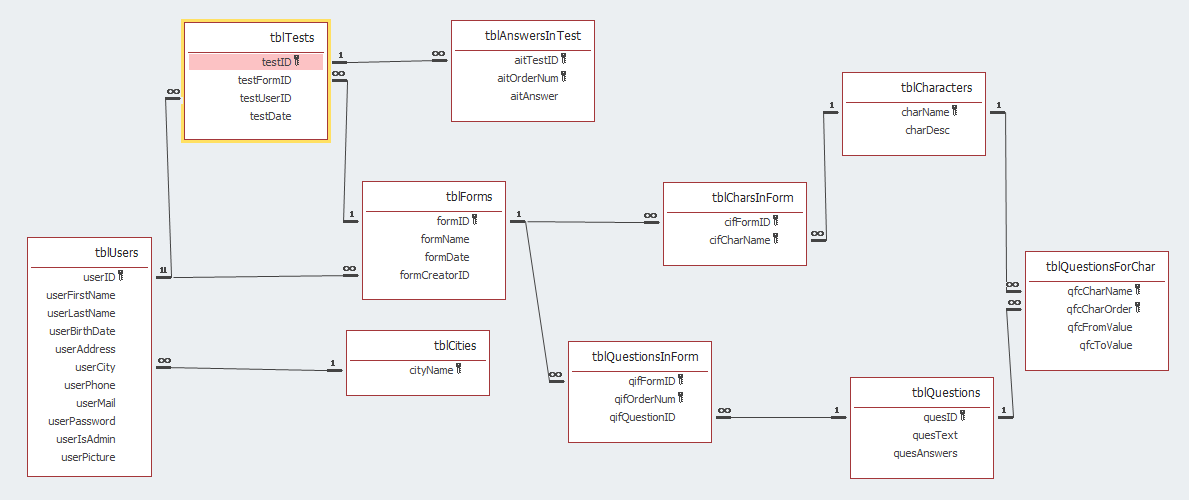
תכונות בטפסים

תשובות במבחנים

מותאמות ב

תכונות

תרשים קשרי הגומלין:



פירוט הטבלאות ושדותיהן:

|  |  |  |
| --- | --- | --- |
| **מבחן אמינות   ReliabilityTest** | | |
| **טבלת משתמשיםtblUsers** | | |
| **שדה** | **טיפוס** | **Field** |
| **ת"ז** | **מספר** | **userID** |
| שם פרטי | טקסט | userFirstName |
| שם משפחה | טקסט | userLastName |
| תאריך לידה | טקסט | userBirthDate |
| כתובת | טקסט | userAddress |
| עיר | טקסט | userCity |
| טלפון | טקסט | userPhone |
| מייל | טקסט | userMail |
| סיסמה | טקסט | userPassword |
| מנהל | כן/לא | userIsAdmin |
| תמונה | טקסט | userPicture |
|  | | |
| **טבלת עריםtblCities** | | |
| **שדה** | **טיפוס** | **Field** |
| **שם עיר** | **טקסט** | **cityName** |
|  | | |
| **טבלת שאלות  tblQuestions** | | |
| **שדה** | **טיפוס** | **Field** |
| **מספר שאלה** | **מספור אוטומטי** | **quesID** |
| טקסט שאלה | טקסט | quesText |
| מספר תשובות | מספר | quesAnswers |
|  | | |
| **טבלת תכונות  tblCharacters** | | |
| **שדה** | **טיפוס** | **Field** |
| **שם תכונה** | **טקסט** | **charName** |
| תיאור | טקסט | charDesc |

**מדריך למשתמש**

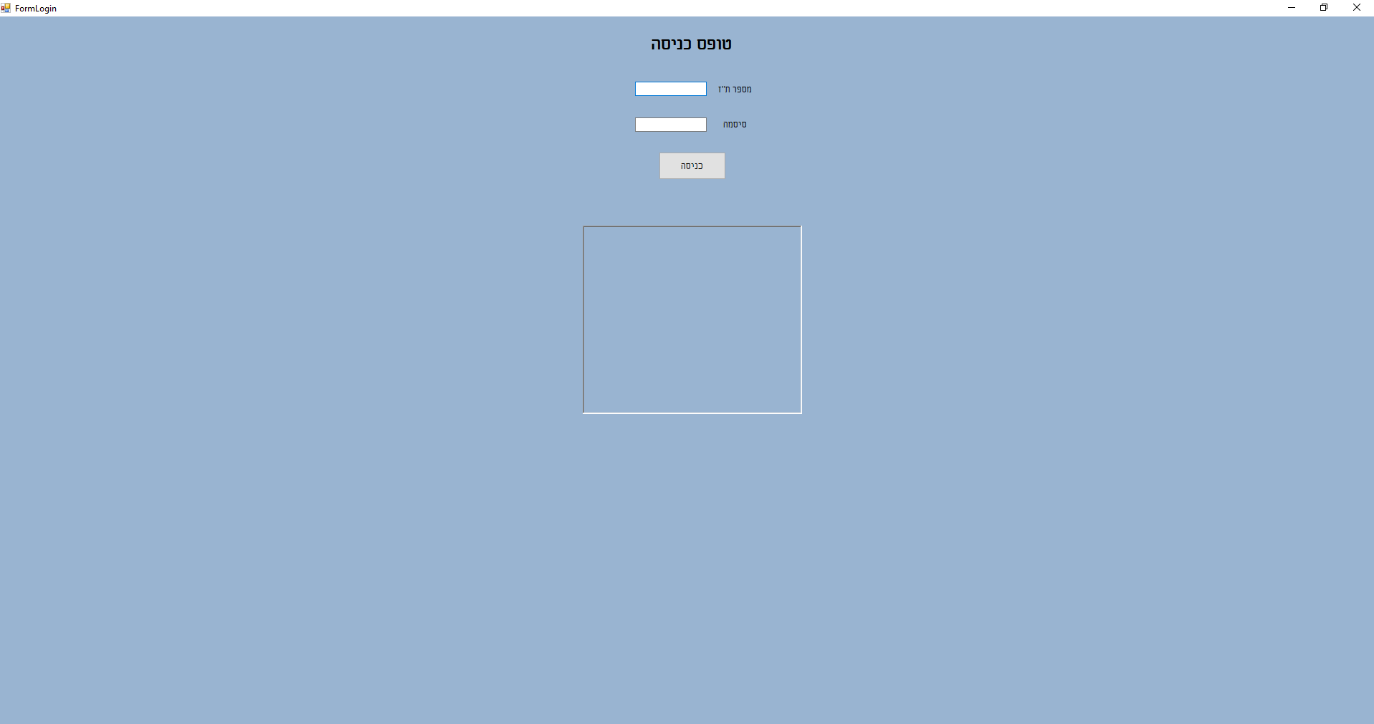
מדריך להפעלת תהליכי המערכת:

במדריך הבא אציג איך להשתמש בכל תהליכי המערכת בליווי תמונות הממחישות את הנאמר.

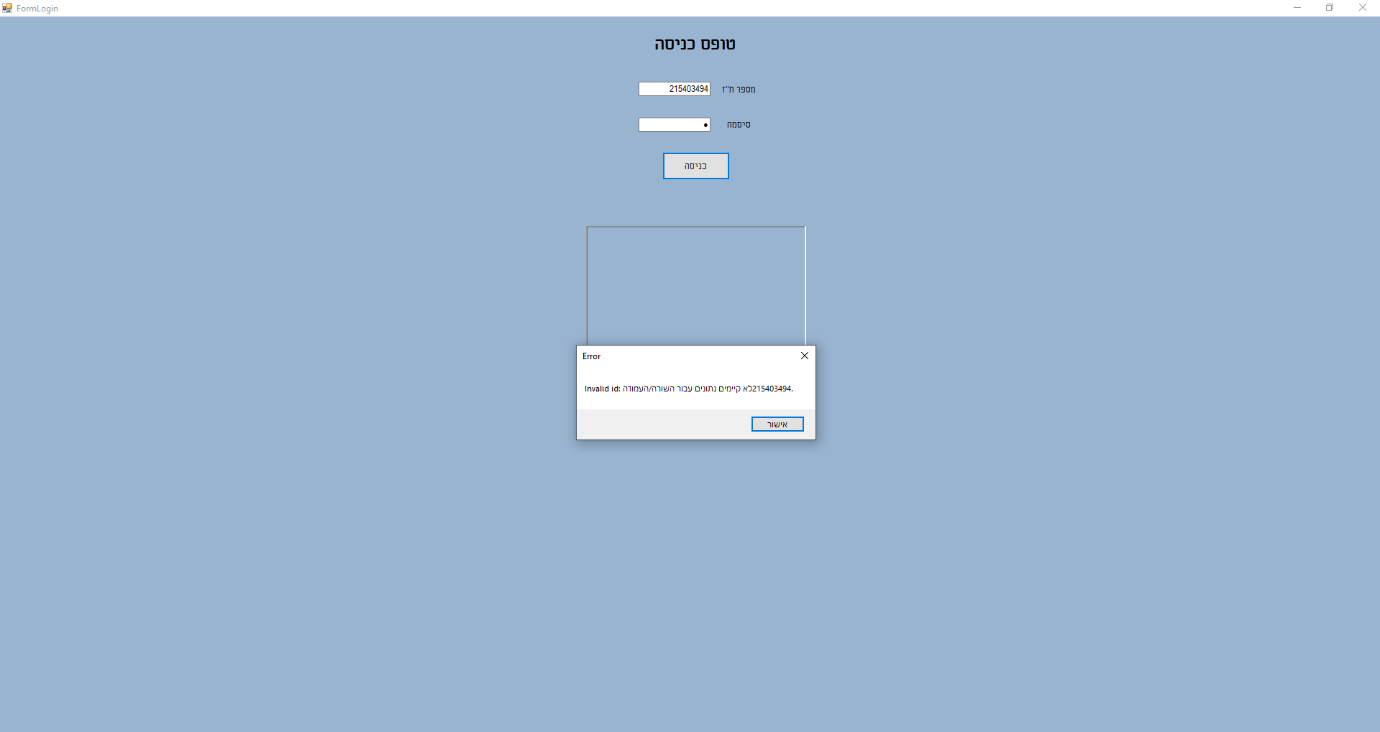
**התחברות למערכת:**

לאחר הכניסה למערכת מופיע מסך ההתחברות למערכת, שם על מנת להיכנס למערכת עליך להכניס את תעודת הזהות שלך ואת הסיסמא שלך, לאחר מכן יש ללחוץ על כפתור ה"כניסה", אם הפרטים שהכנסת נכונים וקיימים במערכת תתקבל הודעת "ברוך הבא" עם השם שלך, ותוכל להיכנס למערכת עם כפתור ה"המשך", במידה והכנסת נתונים לא נכונים שלא קיימים במערכת תחזור אליך הודעת שגיאה ולא תוכל להיכנס עד שתכניס פרטים נכונים.

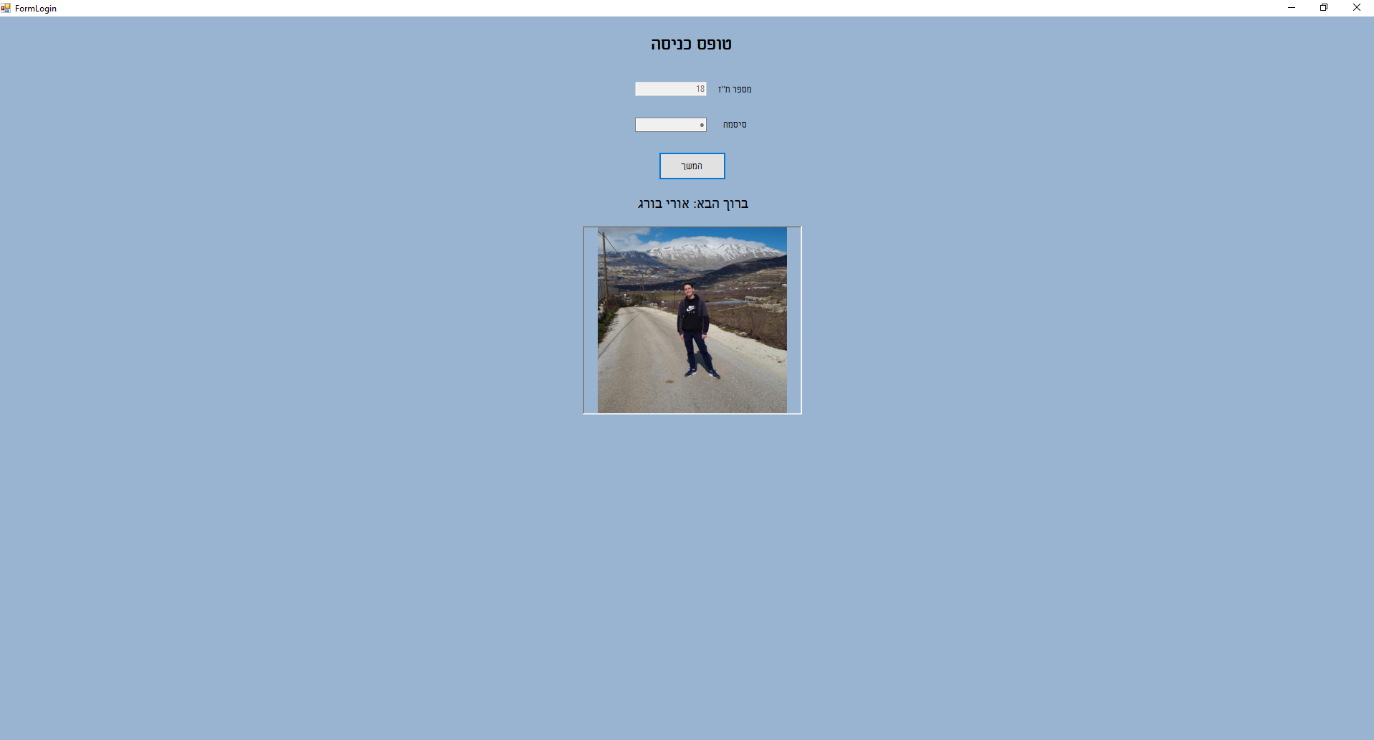
מסך הכניסה:



במידה של הכנסת פרטים שגוים תופיע שגיאה:



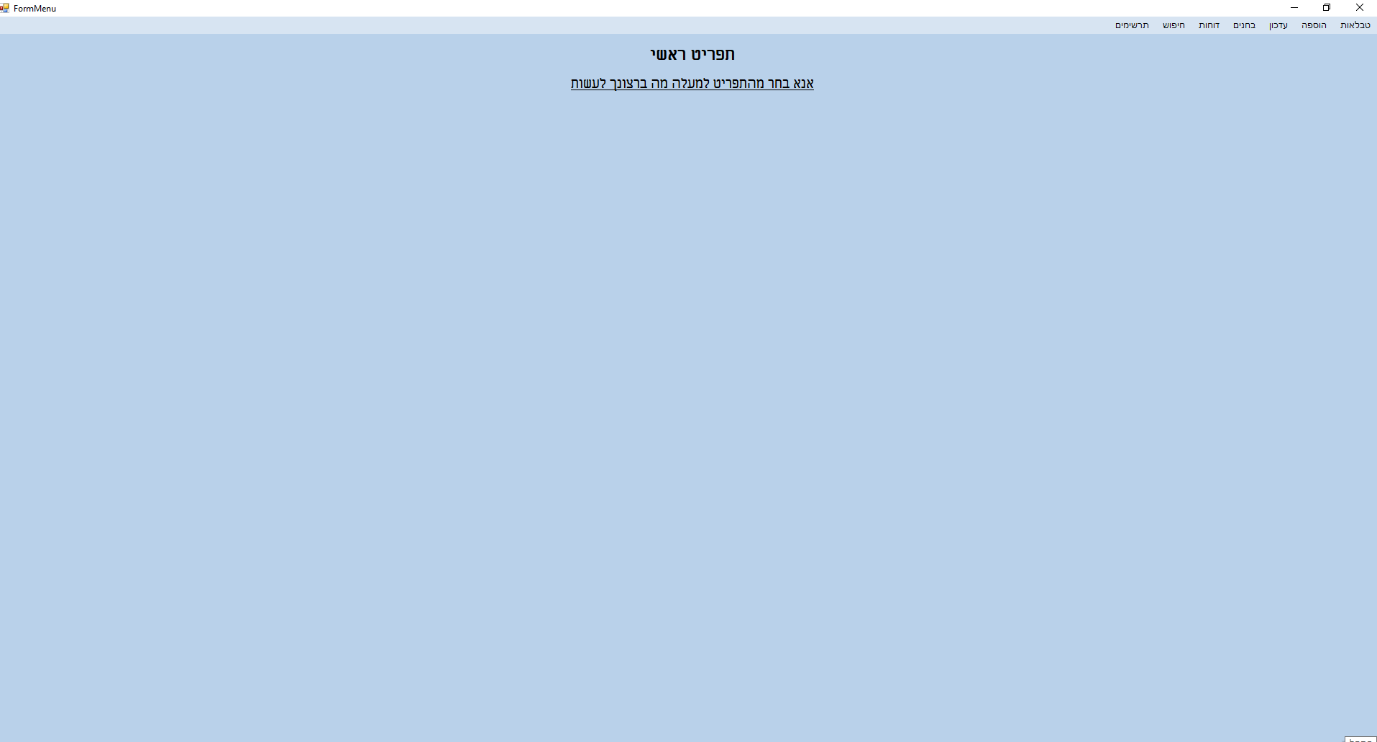
במידה שהפרטים נכונים תופיע "ברוך הבא" וכפתור המשך:



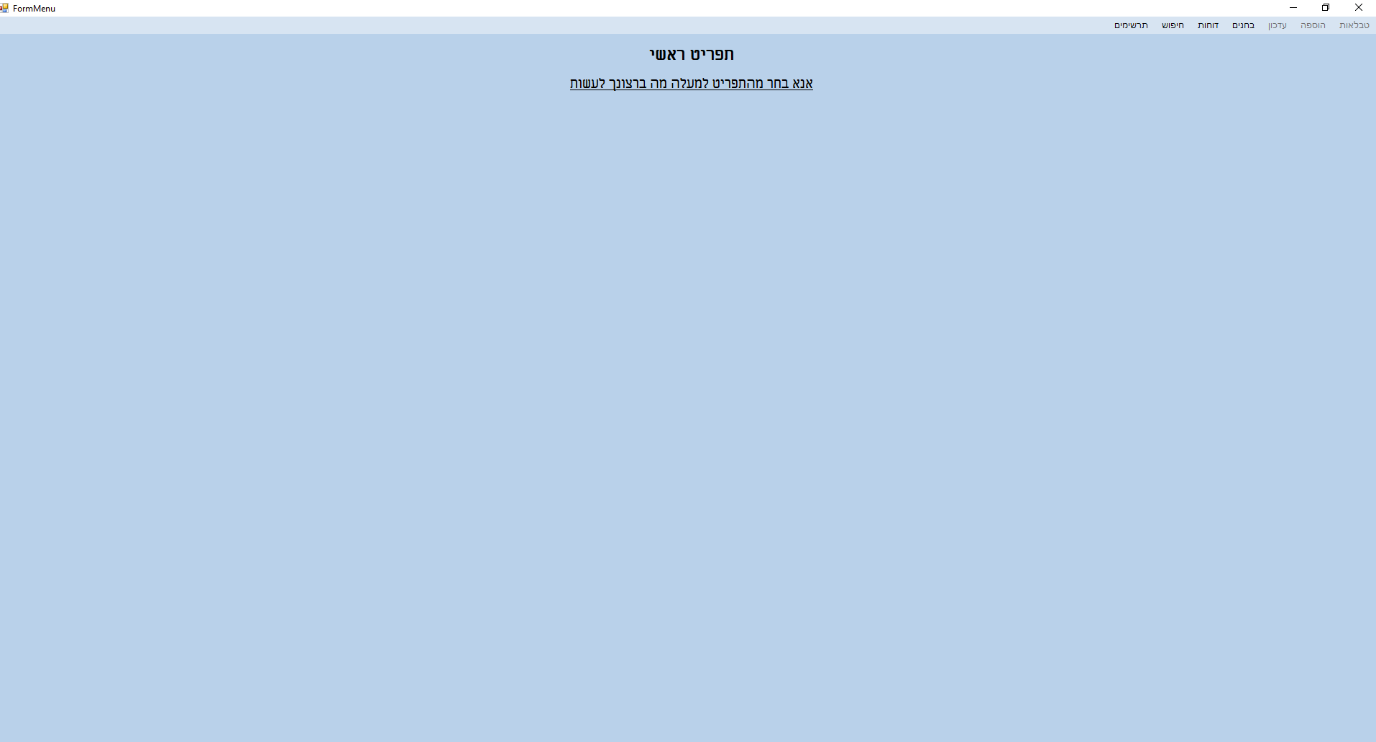
**התפריט הראשי:**

לאחר התחברות למערכת הינך מובל אל התפריט הראשי, מימנו בעזרת סרגל התפריטים שנמצא בחלק העליון של המסך תוכל להתחבר לכל תהליכי המערכת, במידה ויש לך גישת מנהל תוכל לגשת גם אל תהליכי ניהול המידע במערכת, תפריטי- "טבלות", "הוספה" ו"עדכון", במידה ואין לך גישה תוכל לגשת לשאר התפריטים והתהליכים ללא התפריטים הללו, ניתן להיכנס לכל תהליך על ידי לחיצה על התפריט הרצוי ובחירה בתהליך הרצוי מתוך התפריט שנפתח.

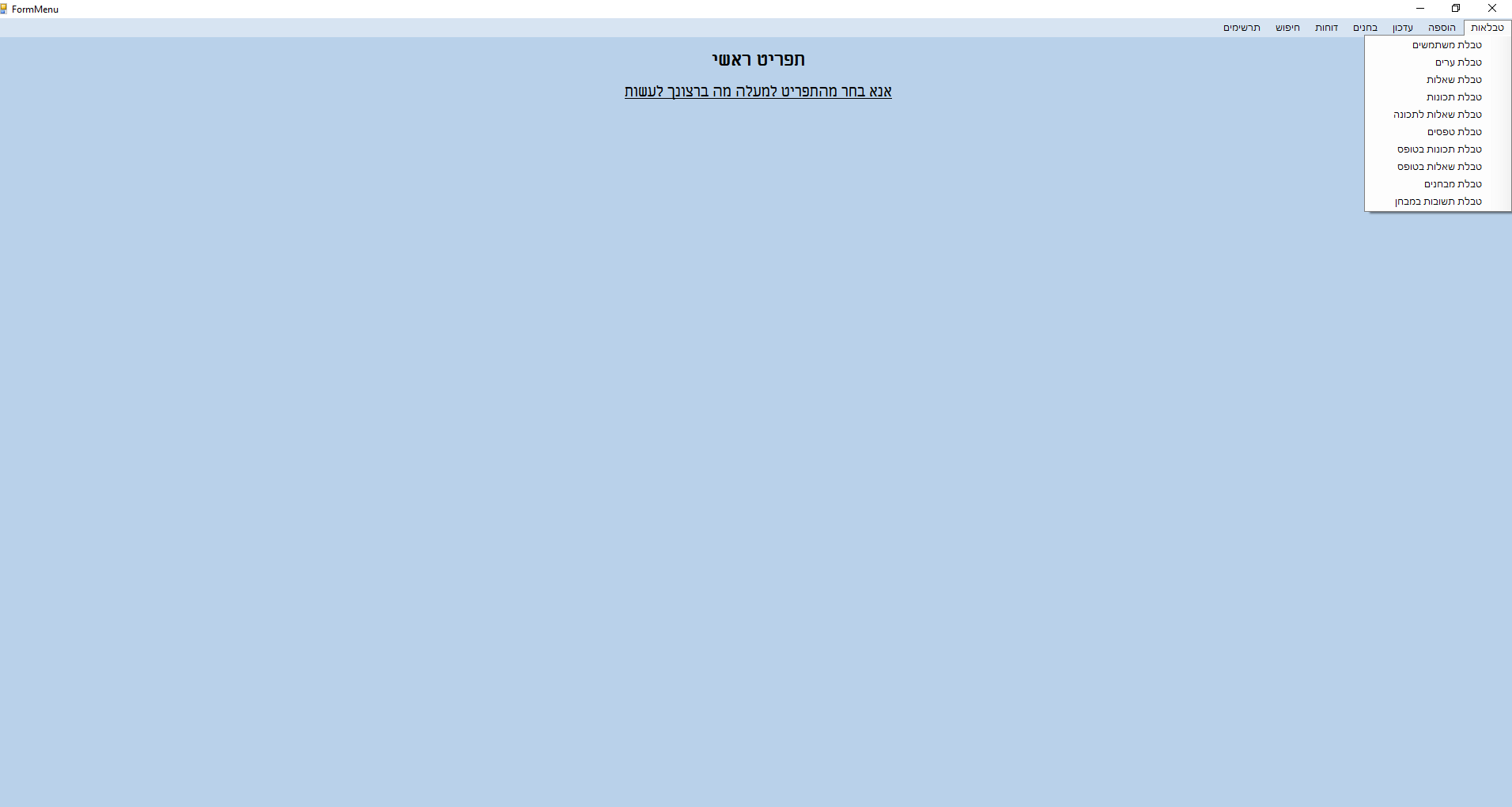
כניסה לתפריט הראשי עם גישת מנהל:



כניסה לתפריט הראשי ללא גישת מנהל:

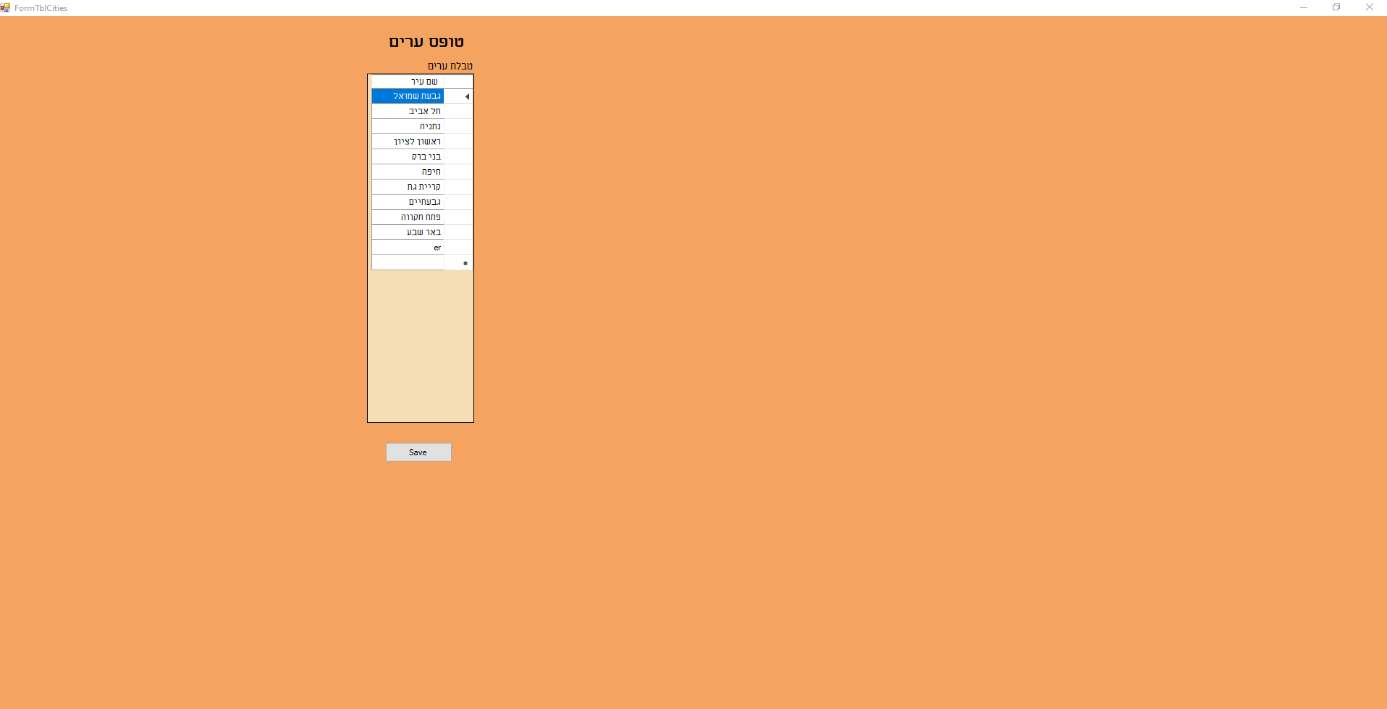
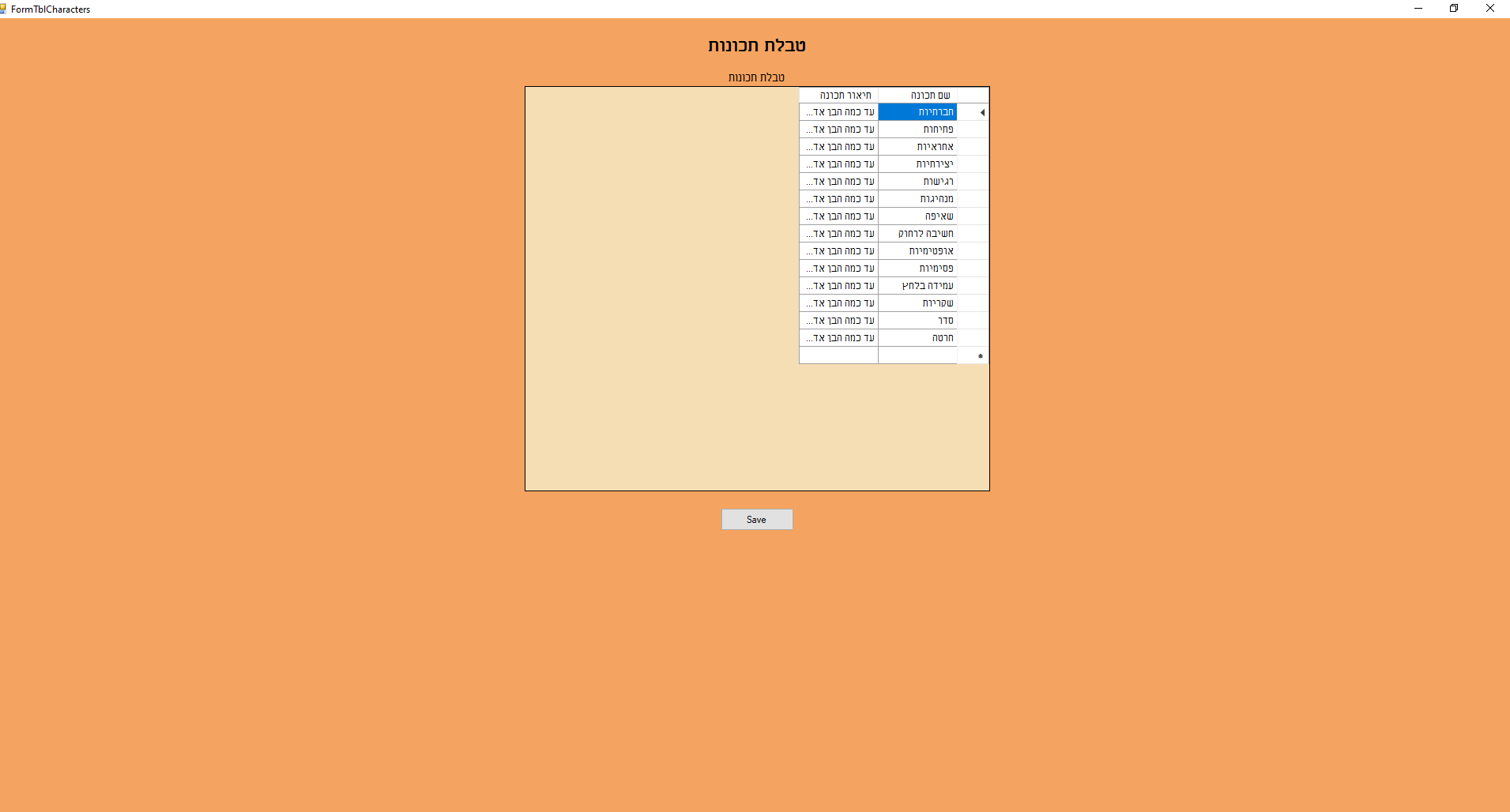
****

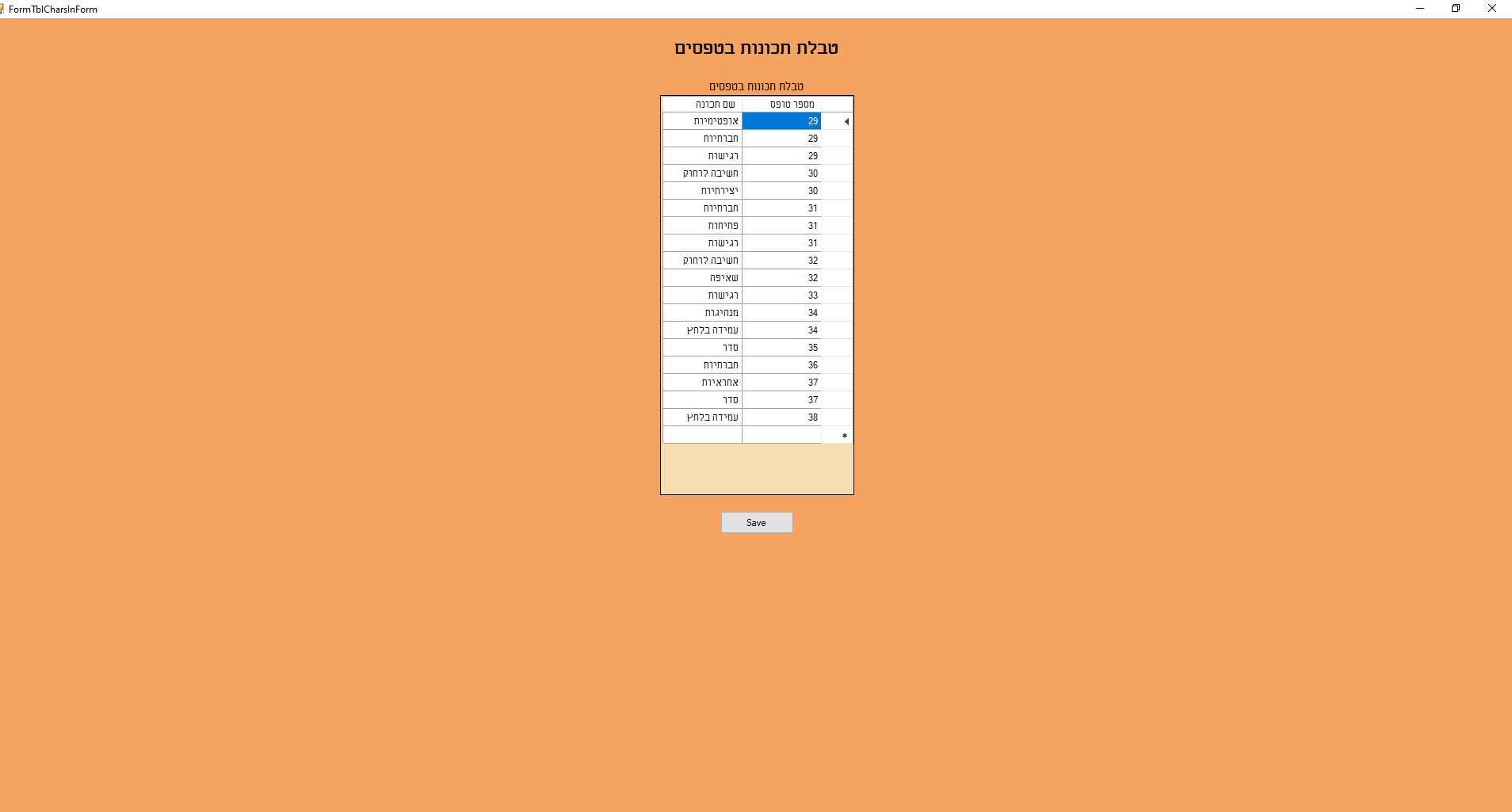
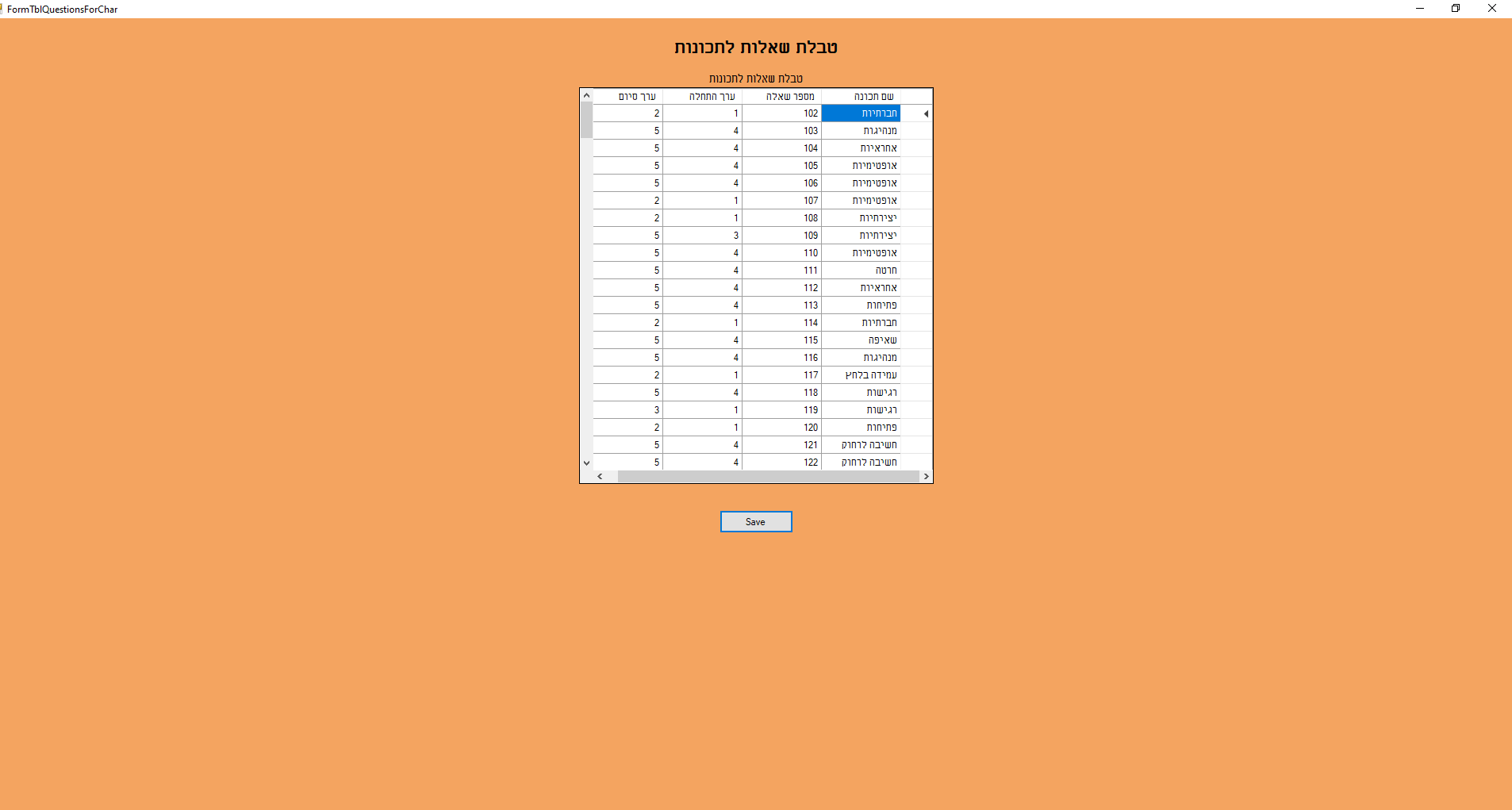
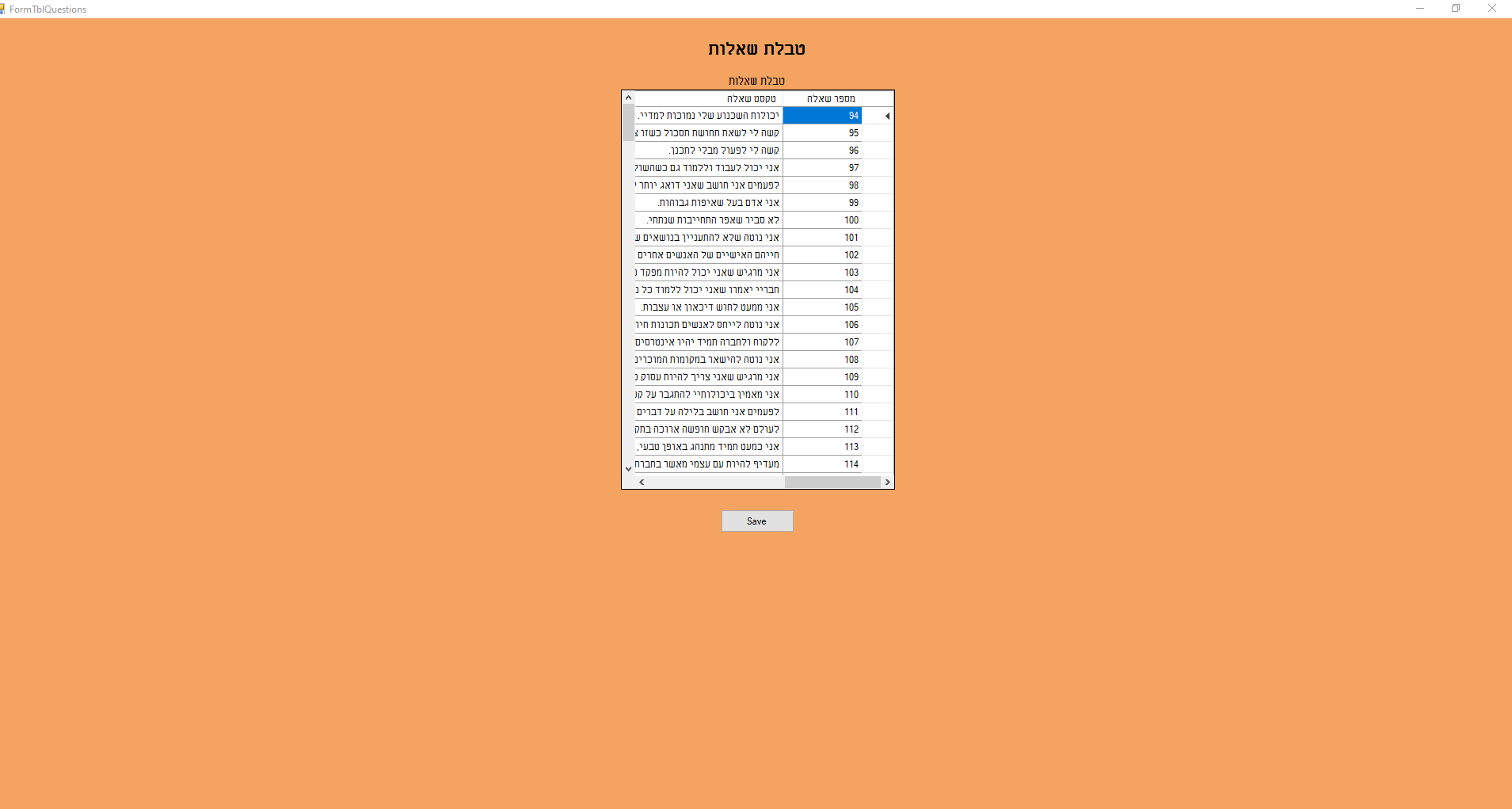
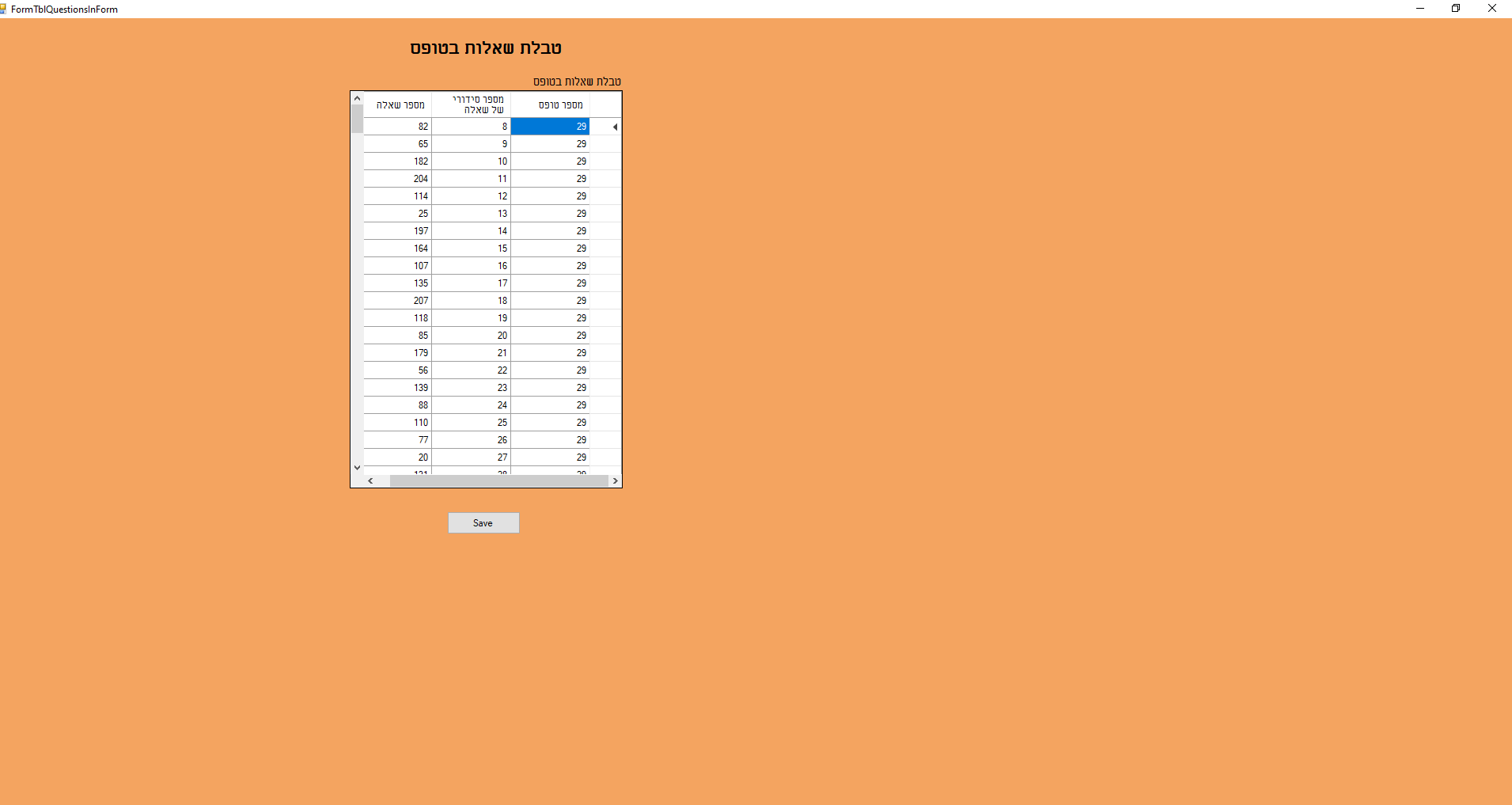
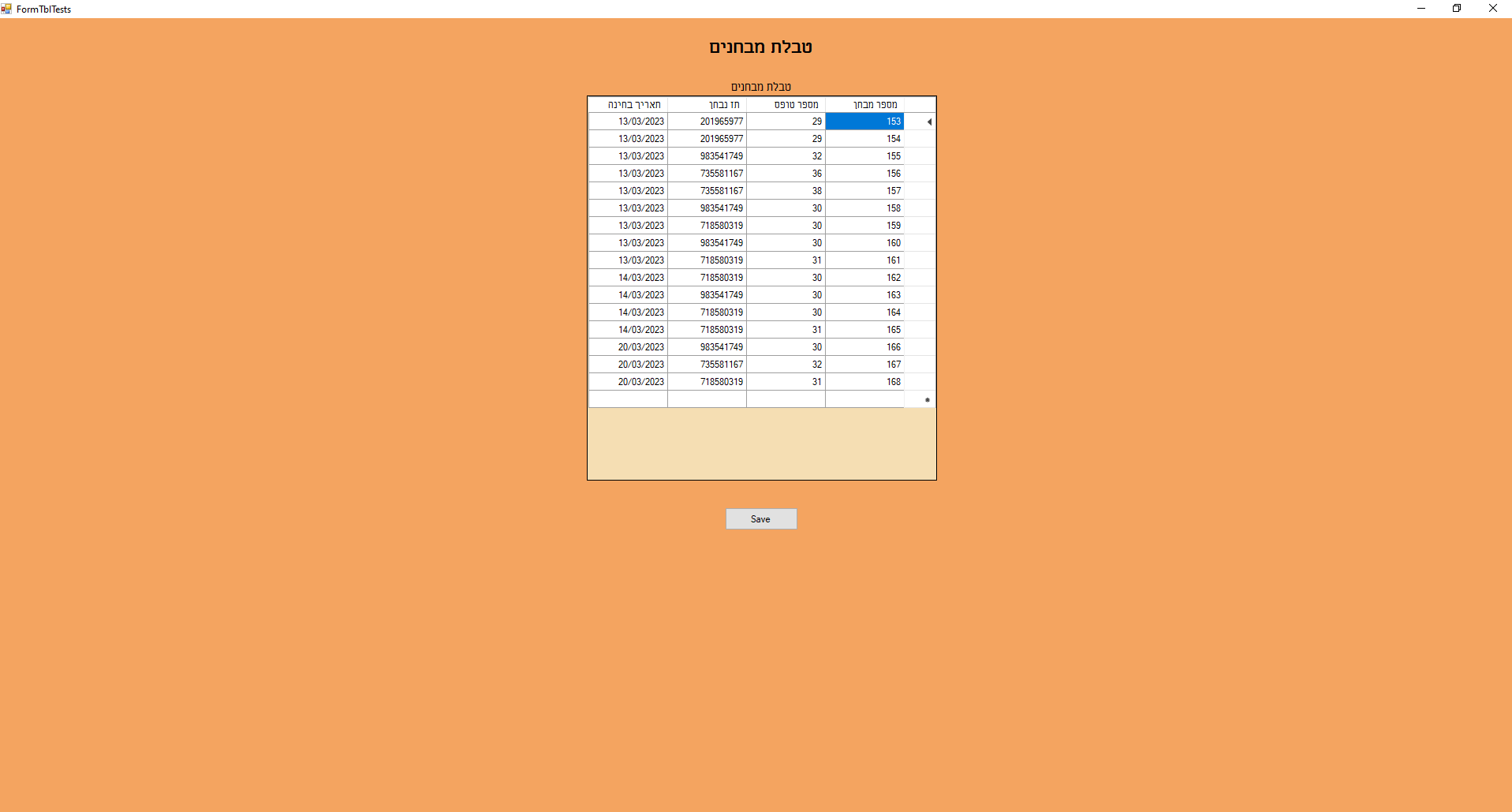
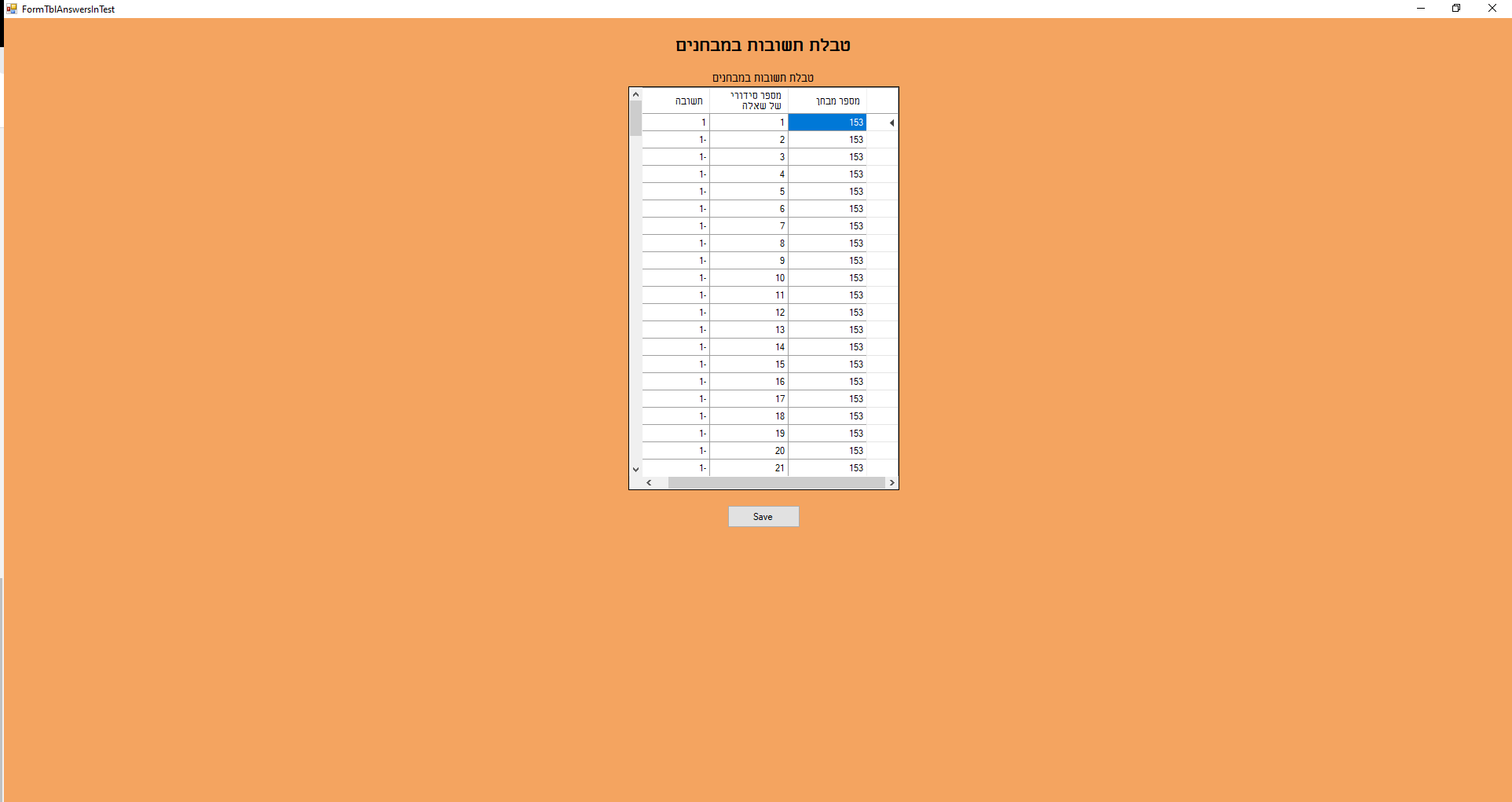
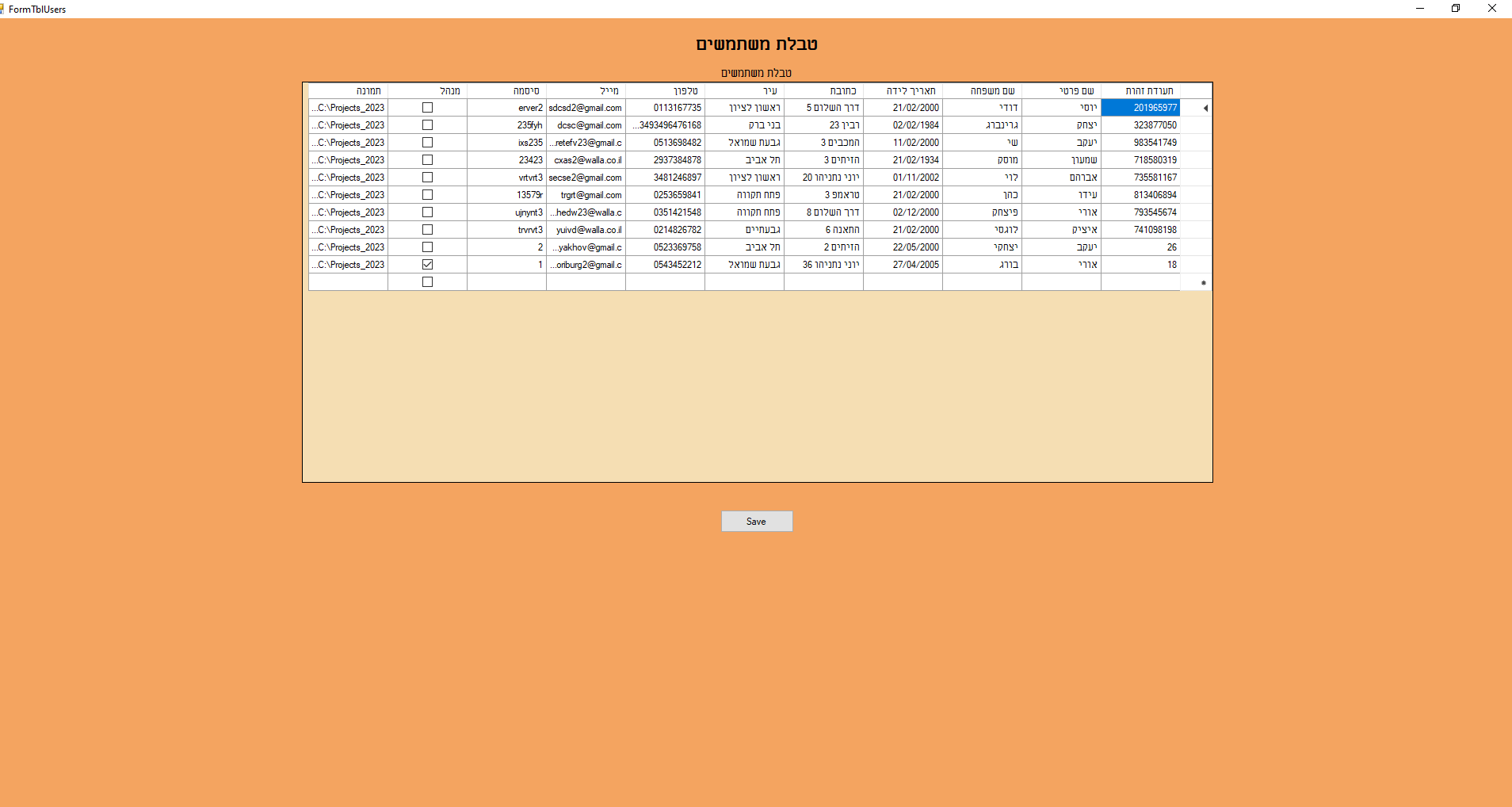
כניסה לאחד מהתפריטים בסרגל התפריטים בתפריט הראשי:

****

**טבלאות:**

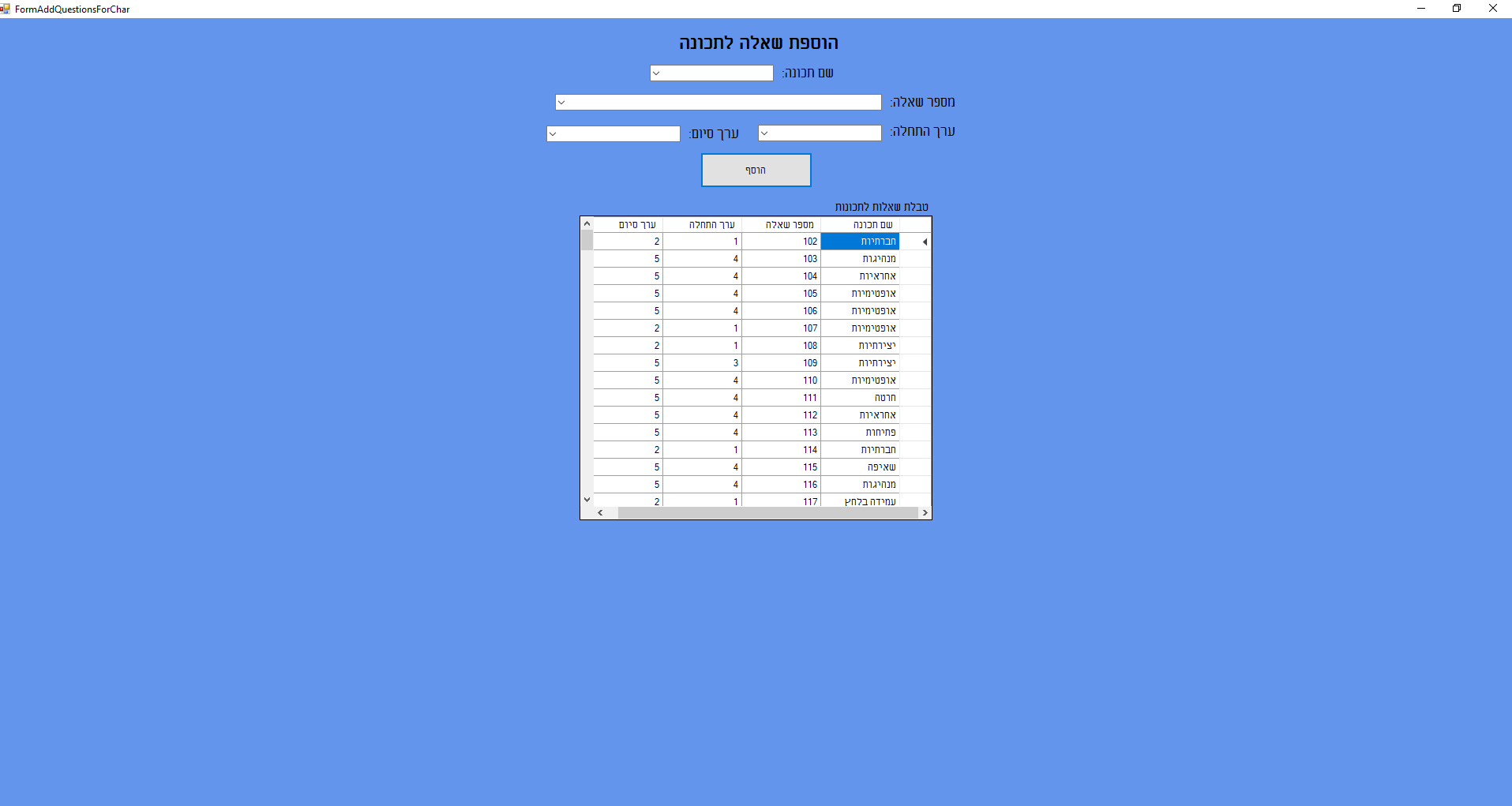
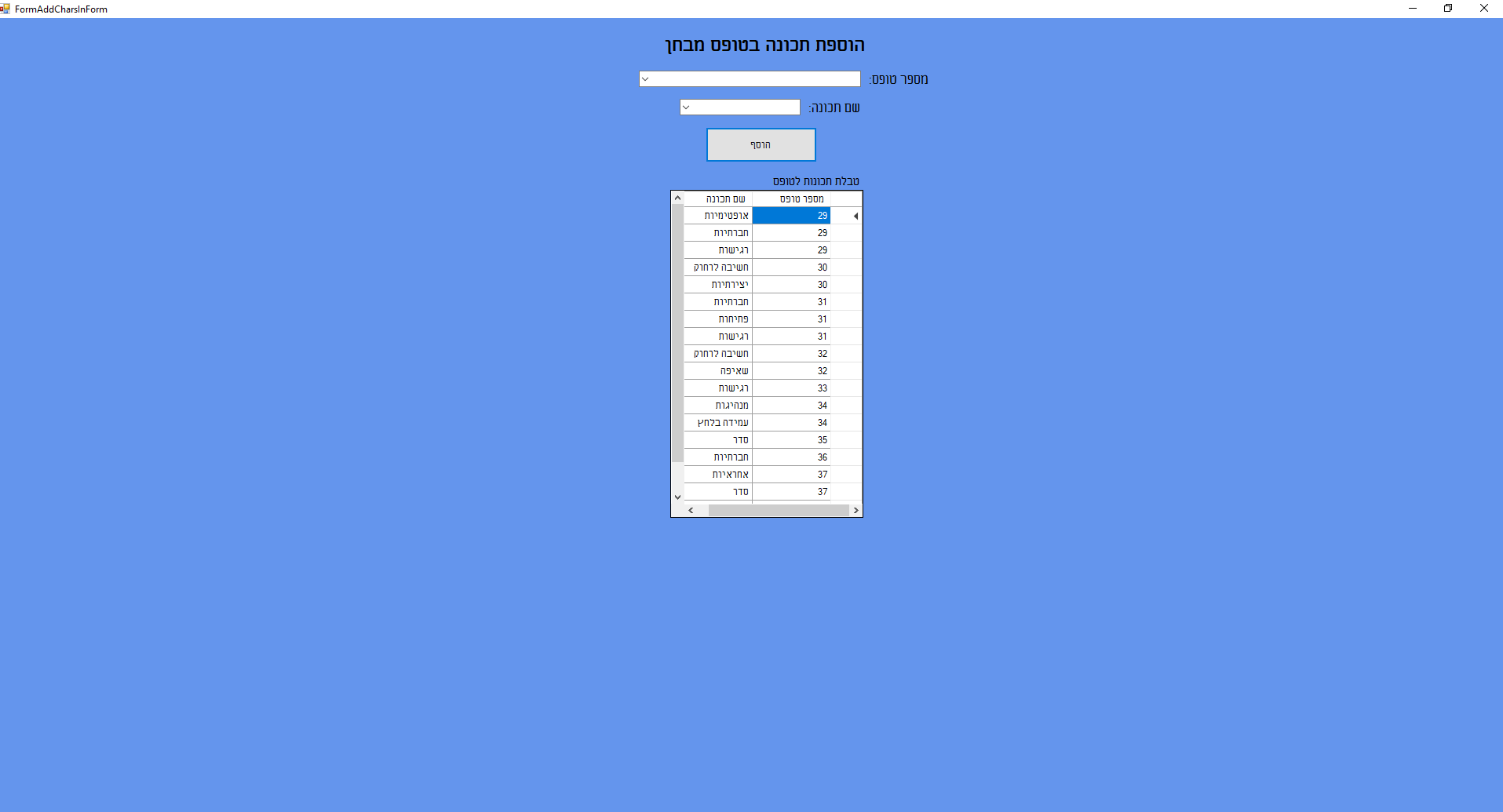
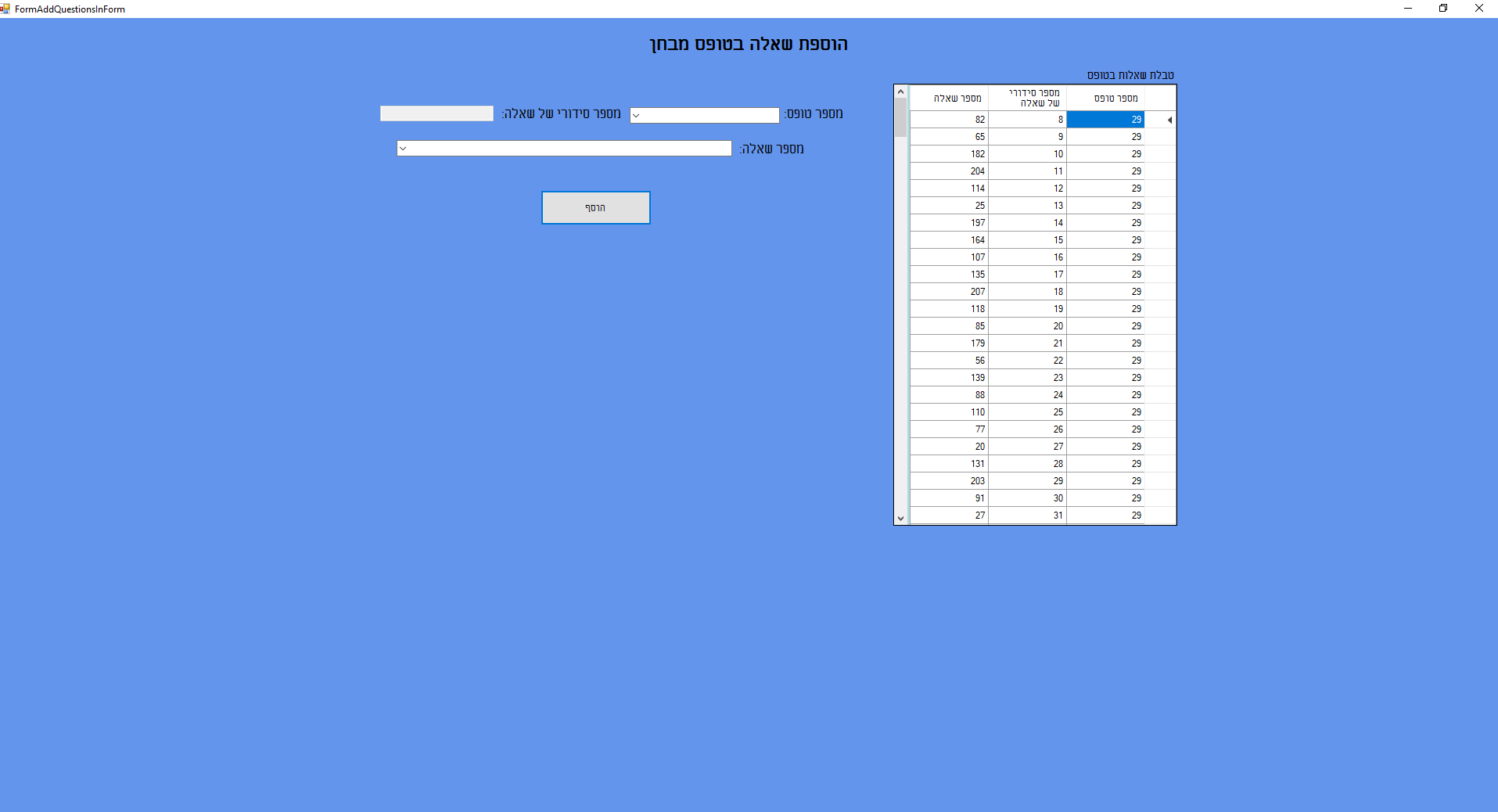
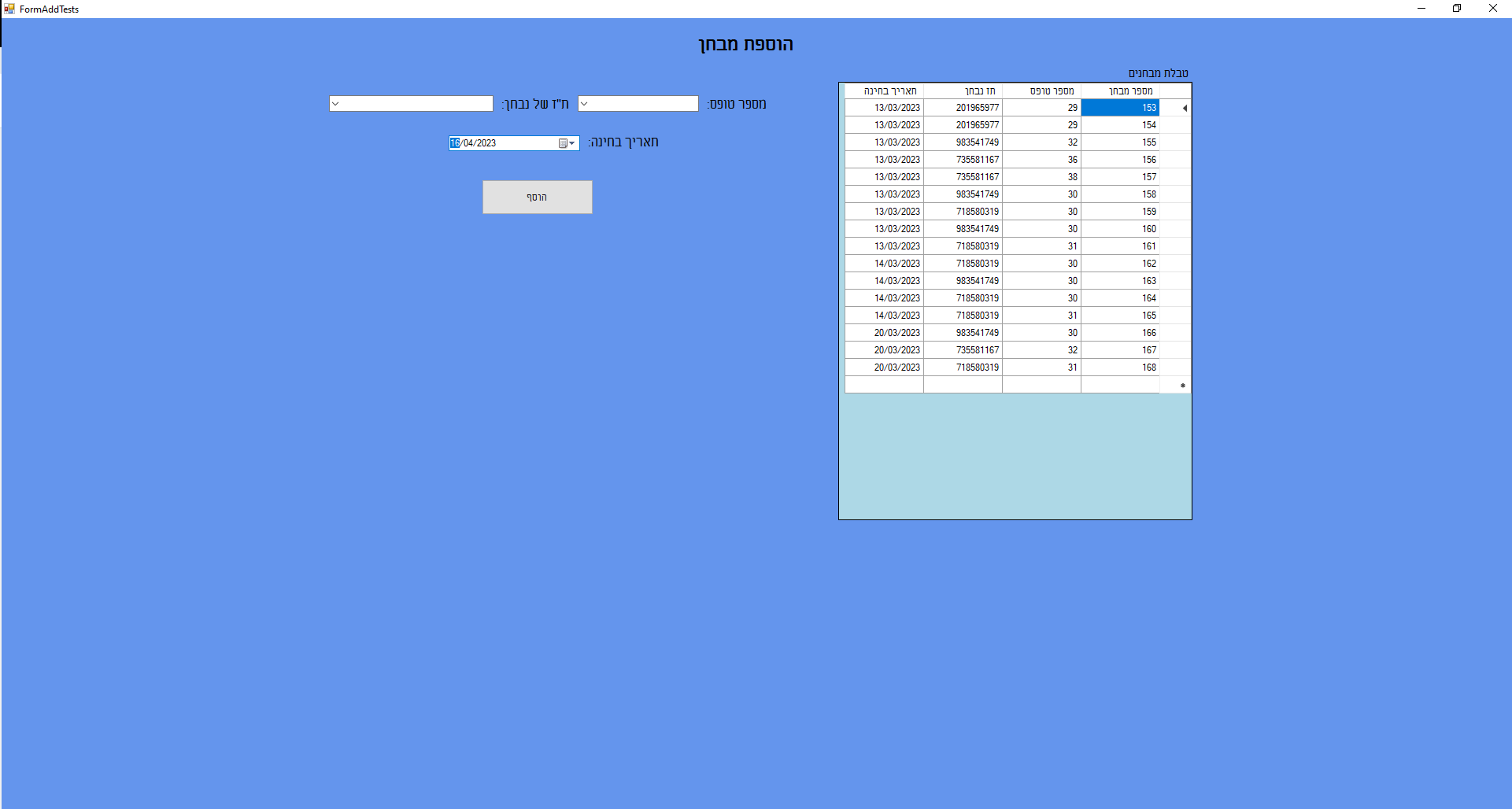
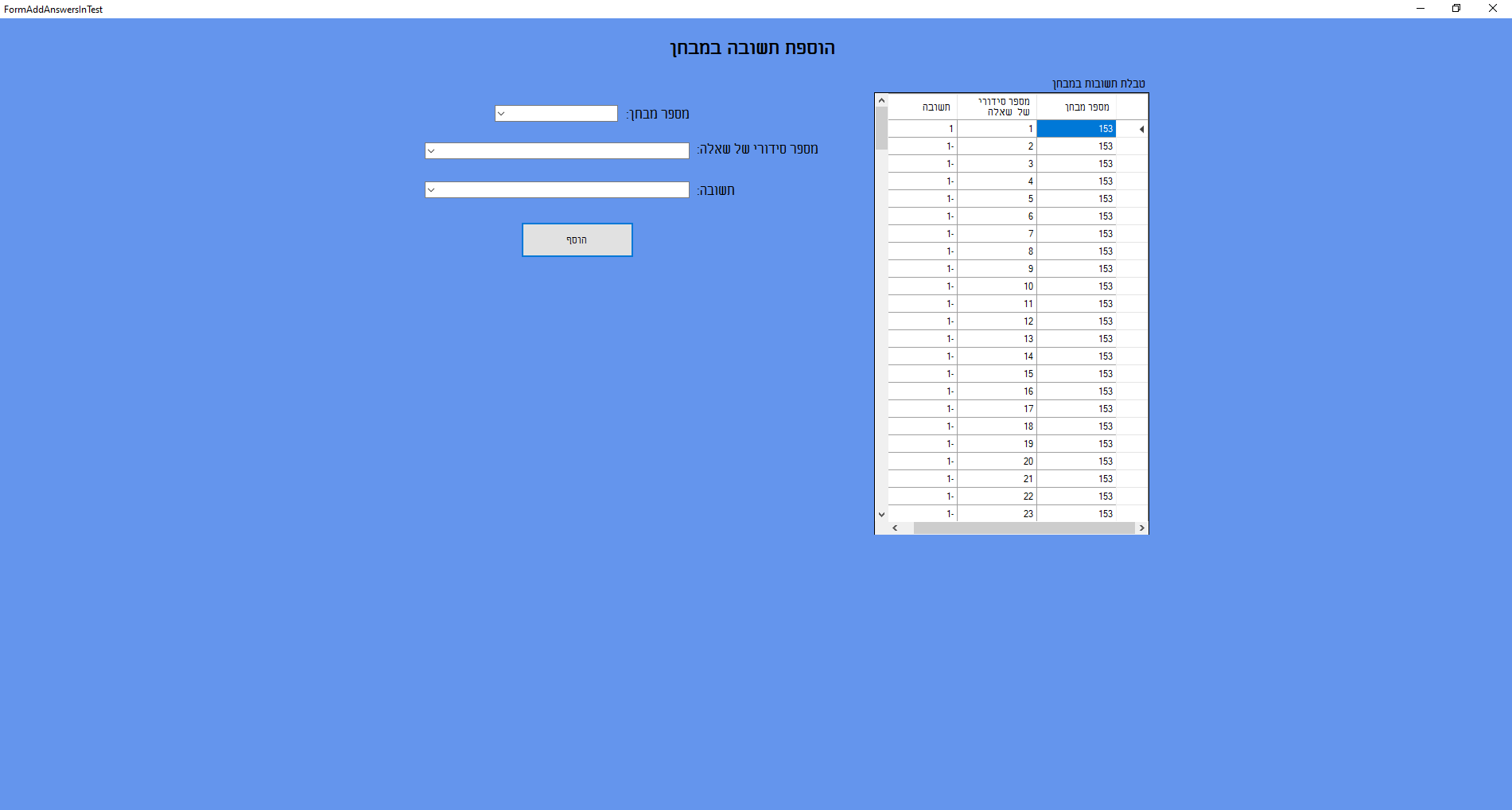
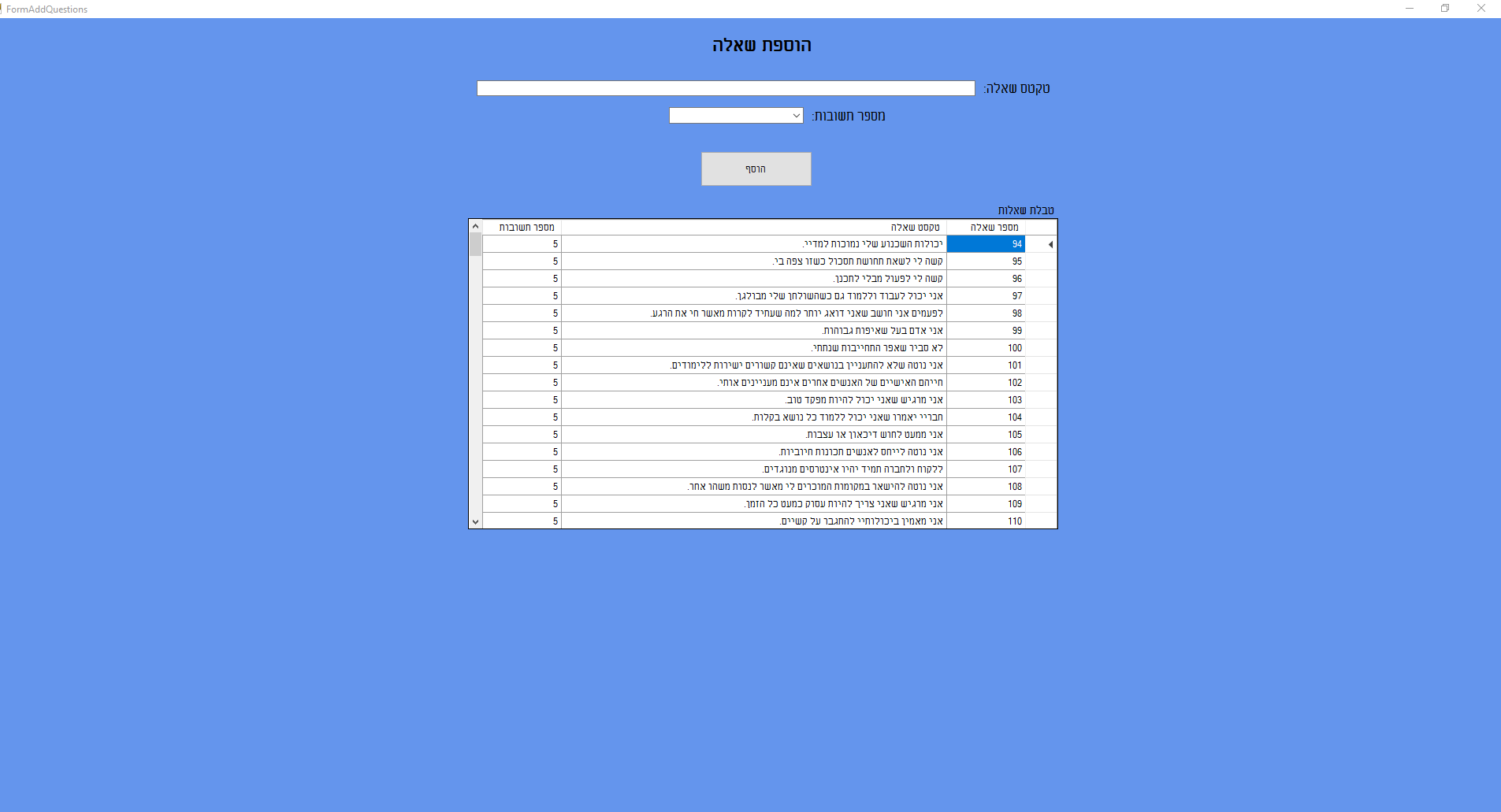
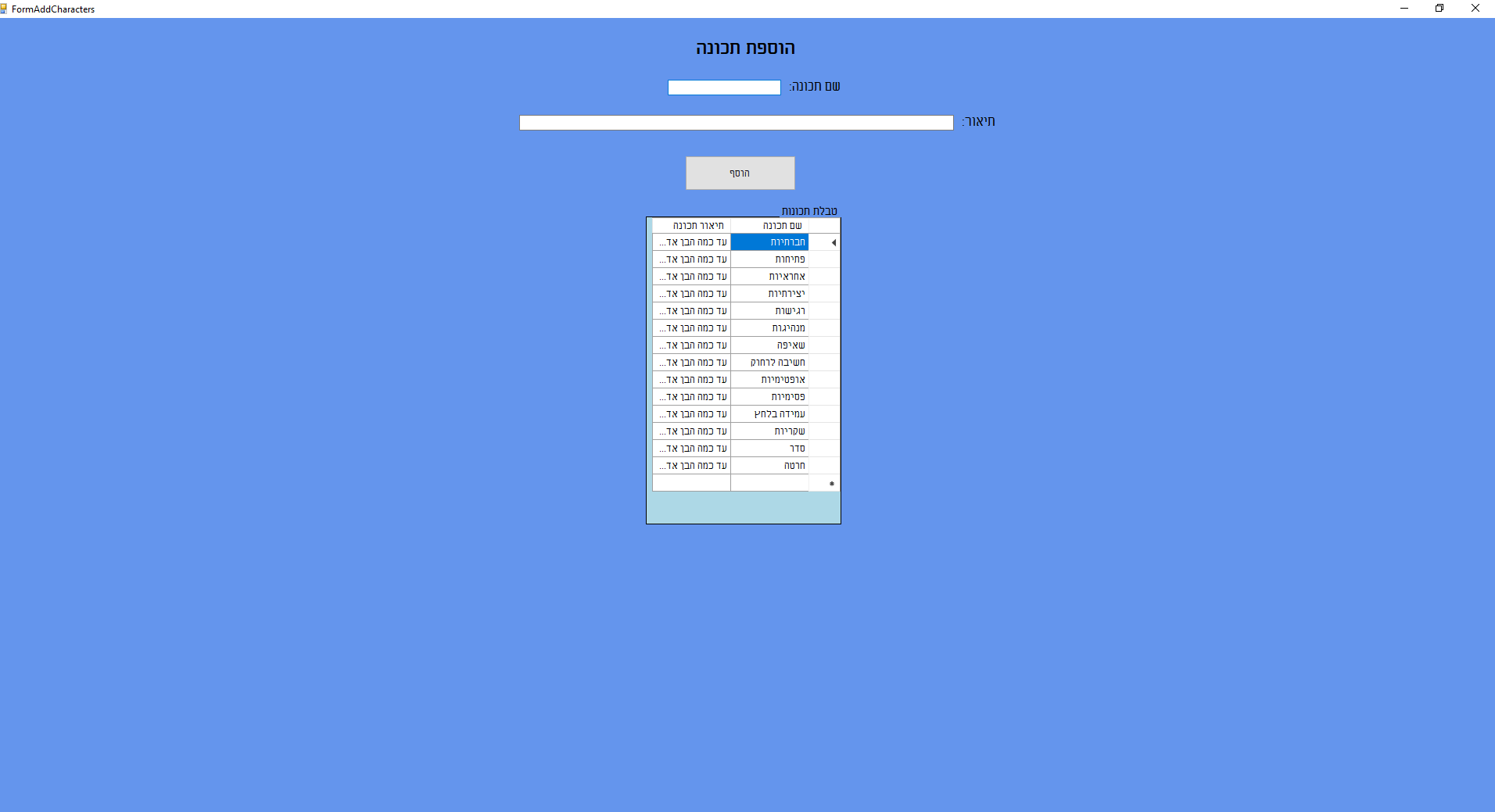
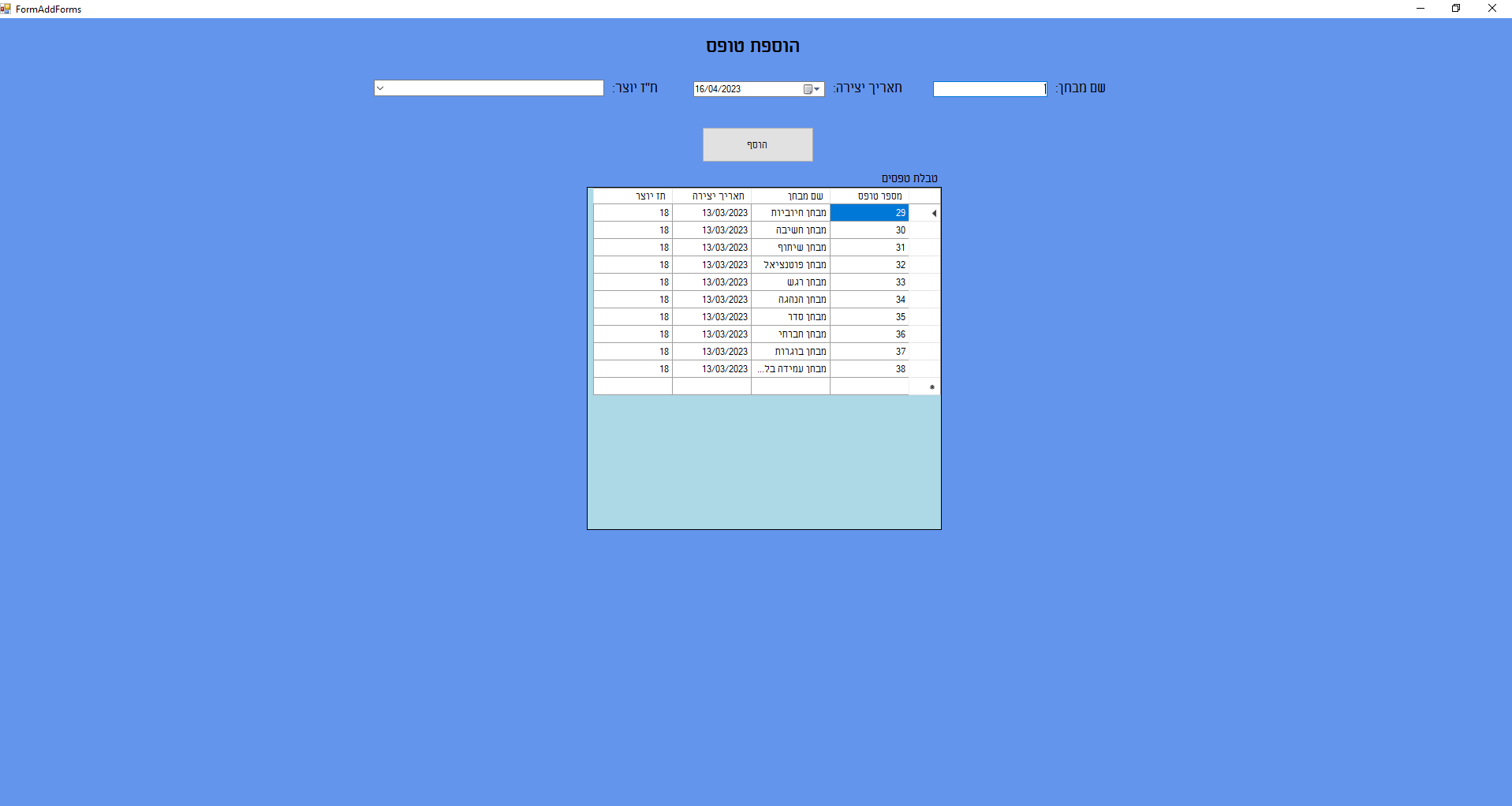
בלחיצה בסרגל התפריטים על "טבלאות" יפתחו תהליכי הטבלאות במערכת, מתוכם משתמש עם גישת מנהל יכול באופן ישיר לשנות כל מידע שירצה בממסד הנתונים באופן נגיש, לאחר שבוצע השינוי הרצוי ניתן ללחוץ על "save" על מנת לשמור את השינויים בטבלה.

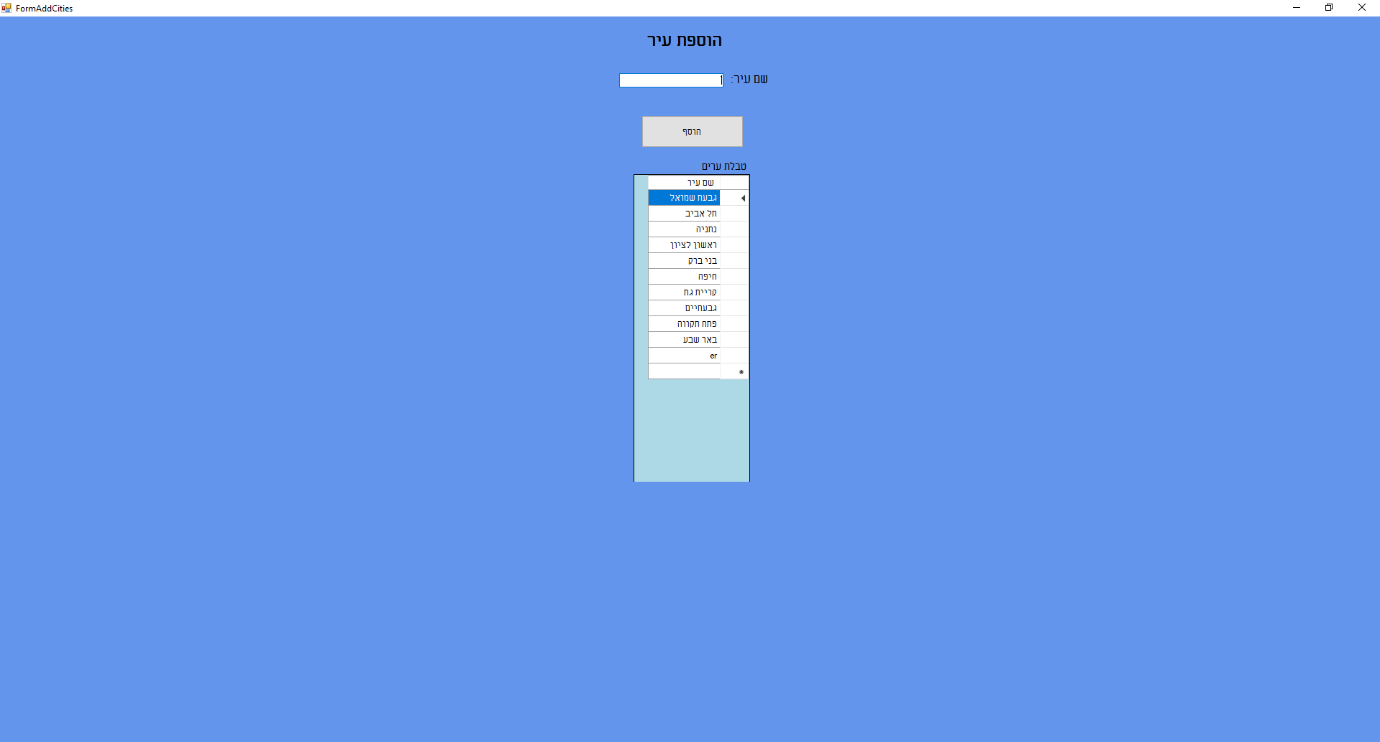
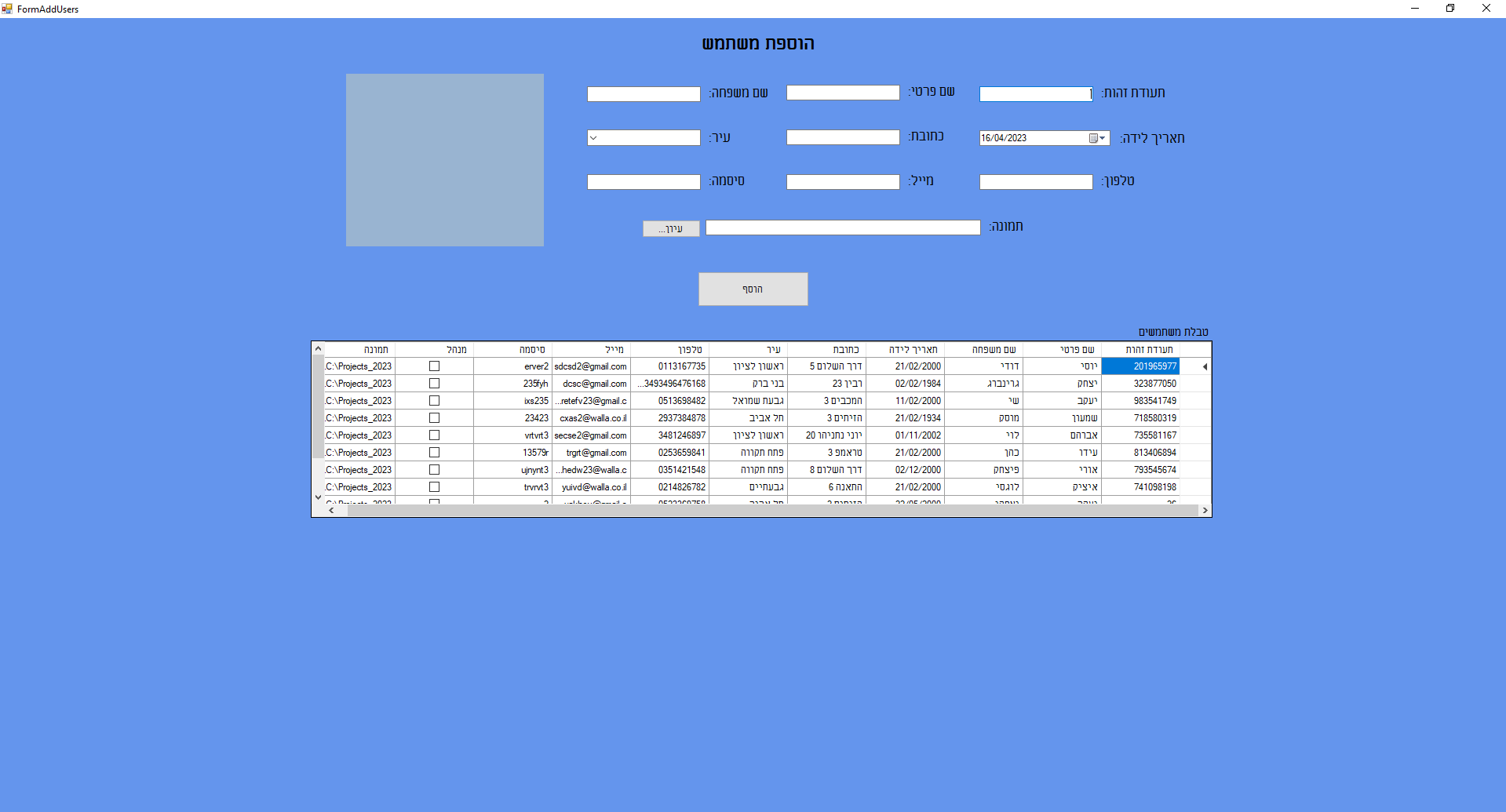
טבלאות המערכת:



**הוספה:**

בלחיצה בסרגל התפריטים על "הוספה" יפתחו תהליכי ההוספה במערכת, מתוכם משתמש עם גישת מנהל יכול להוסיף מידע לכל אחד מטבלאות המערכת בצורה פשוטה, וללא צורך הבנת המערכת לעומק, חלק מתהליכי ההוספה נותנים לך אפשרות בחירה מתוך אופציות, לדוגמא כשיש צורך בשאלה אז בחירה מתוך השאלות הקימות, וחלק ישנם פשוט שורות ריקות למילוי חופשי, לאחר המילוי הרצוי ניתן ללחוץ על "הוסף" והנתונים יוספו למערכת.

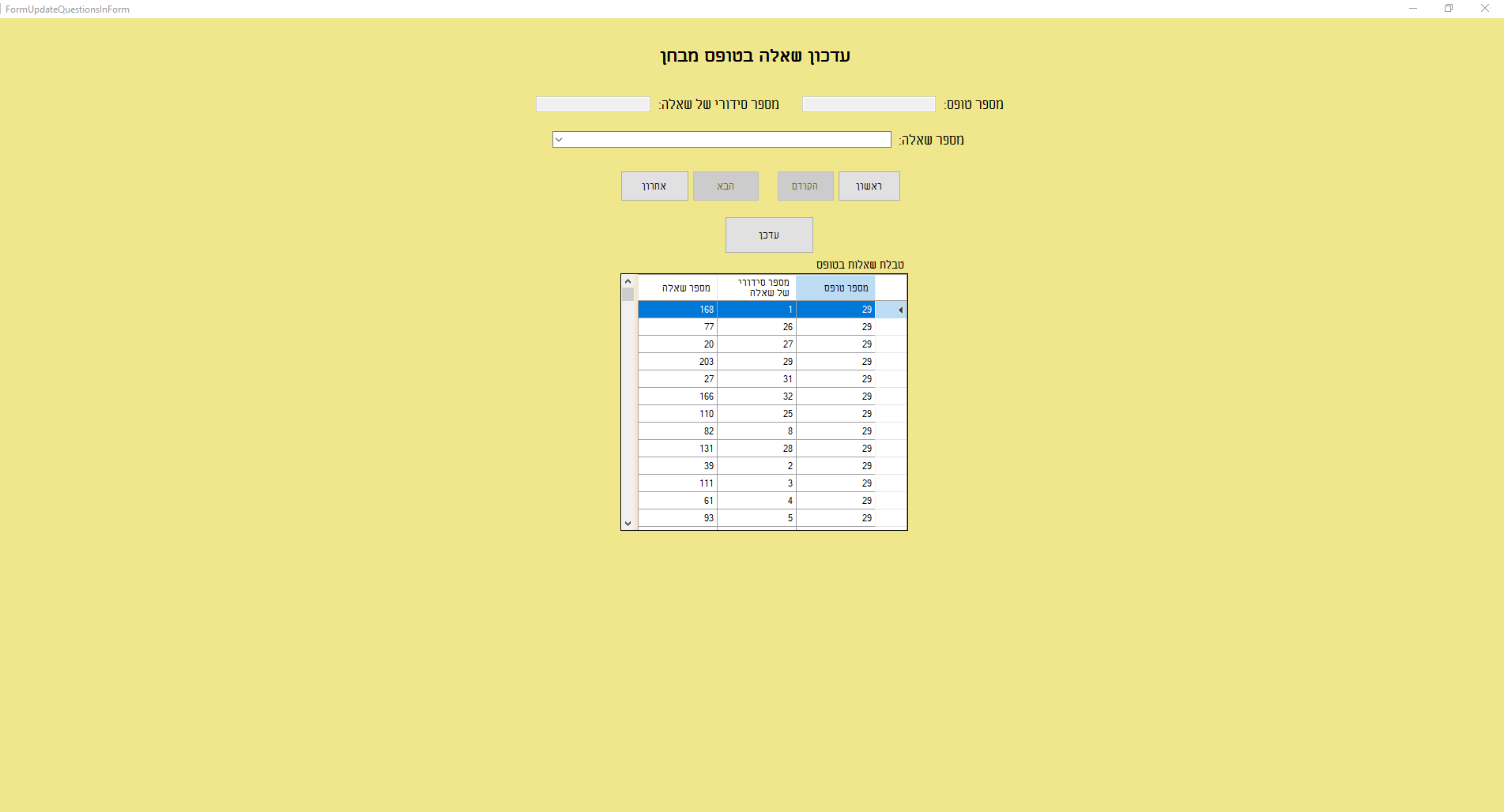
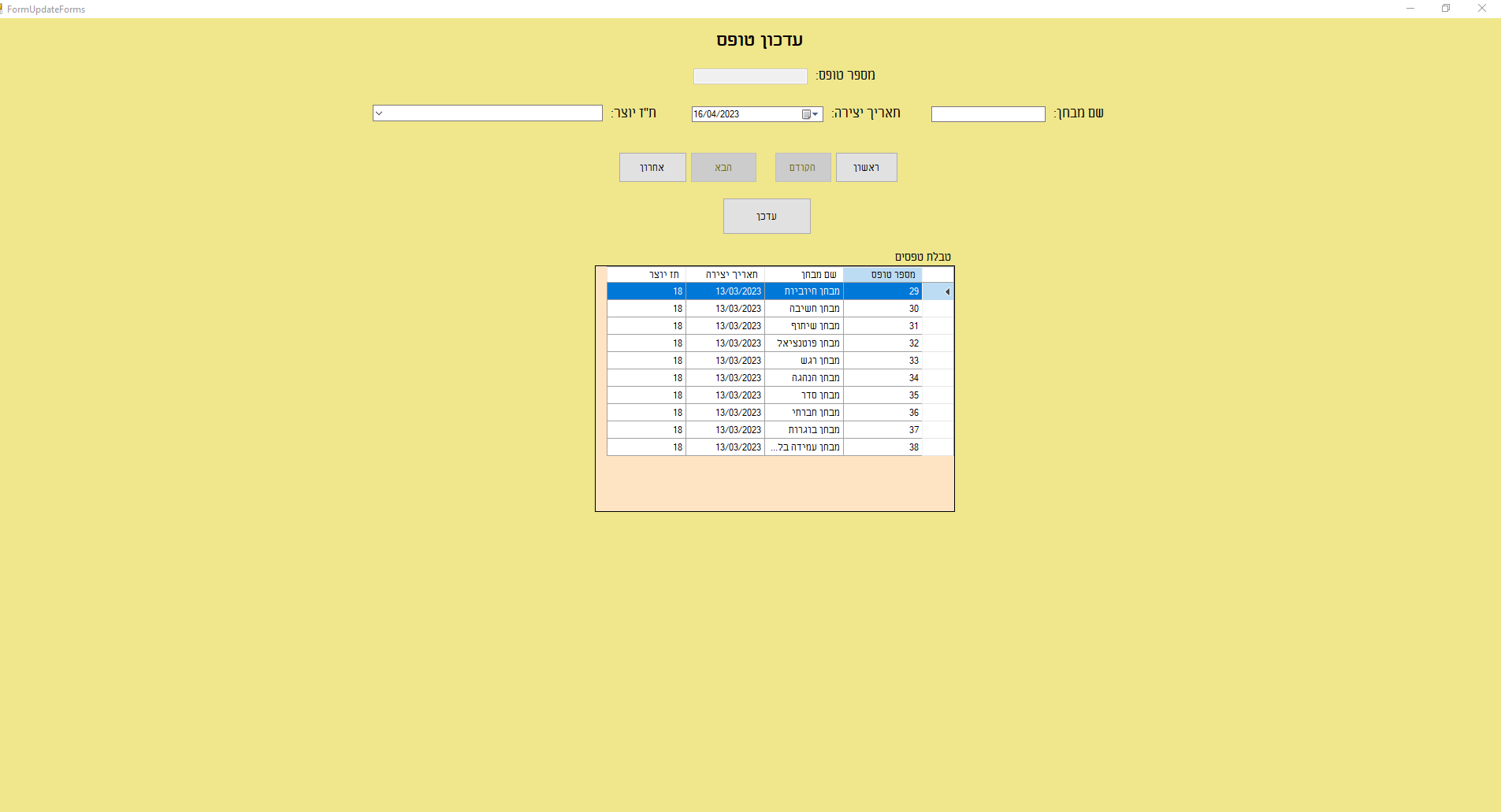
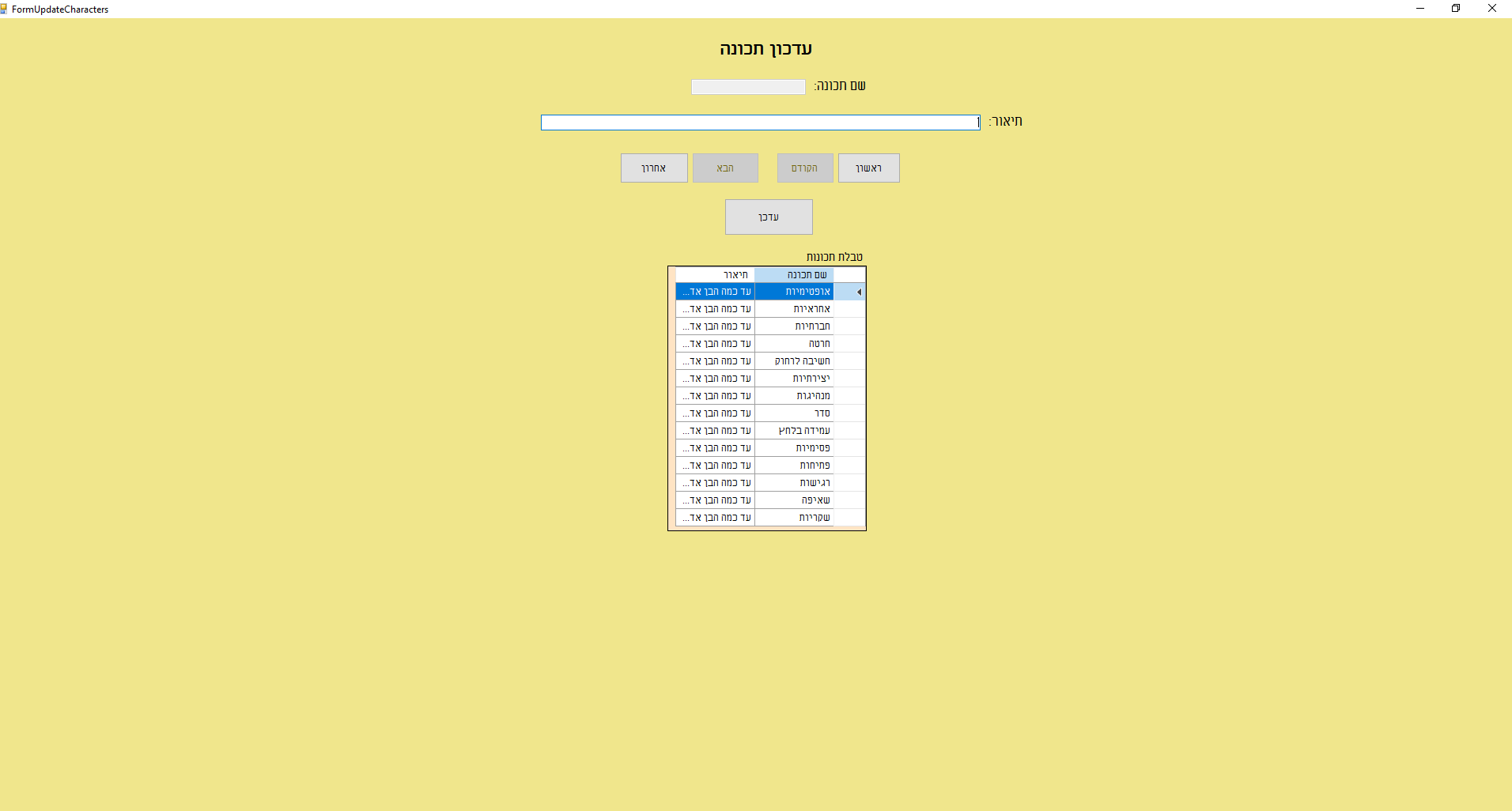
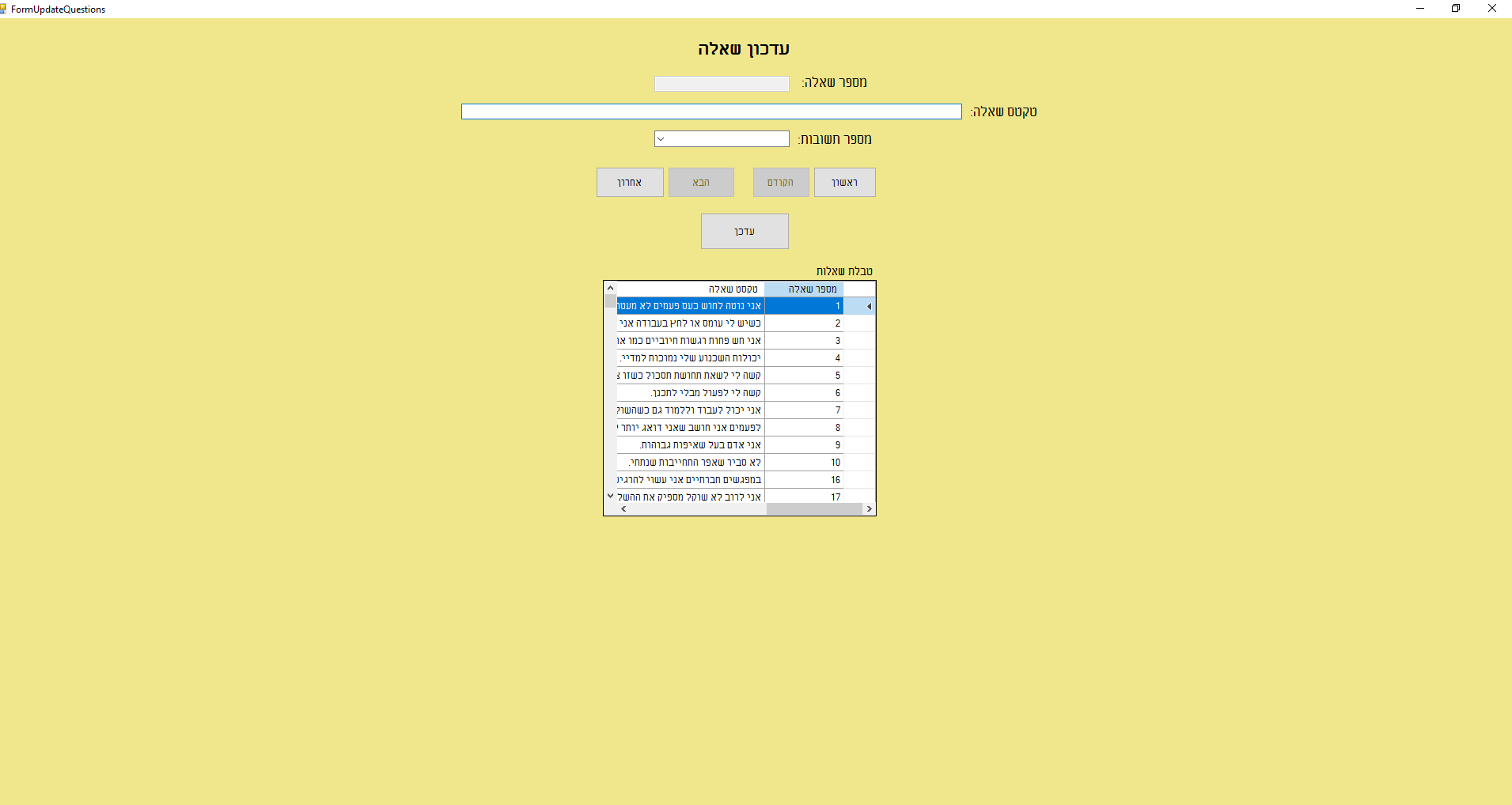
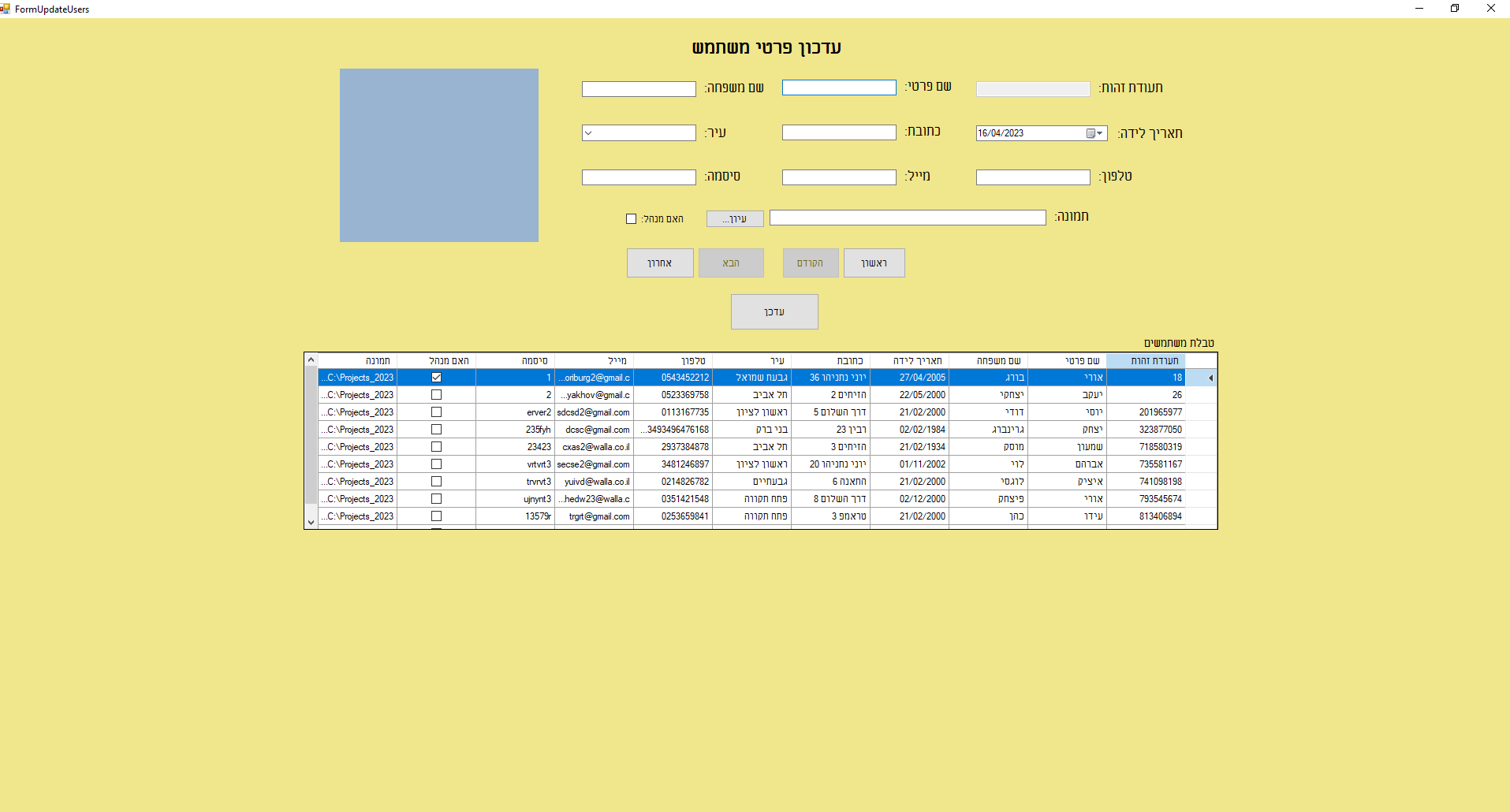
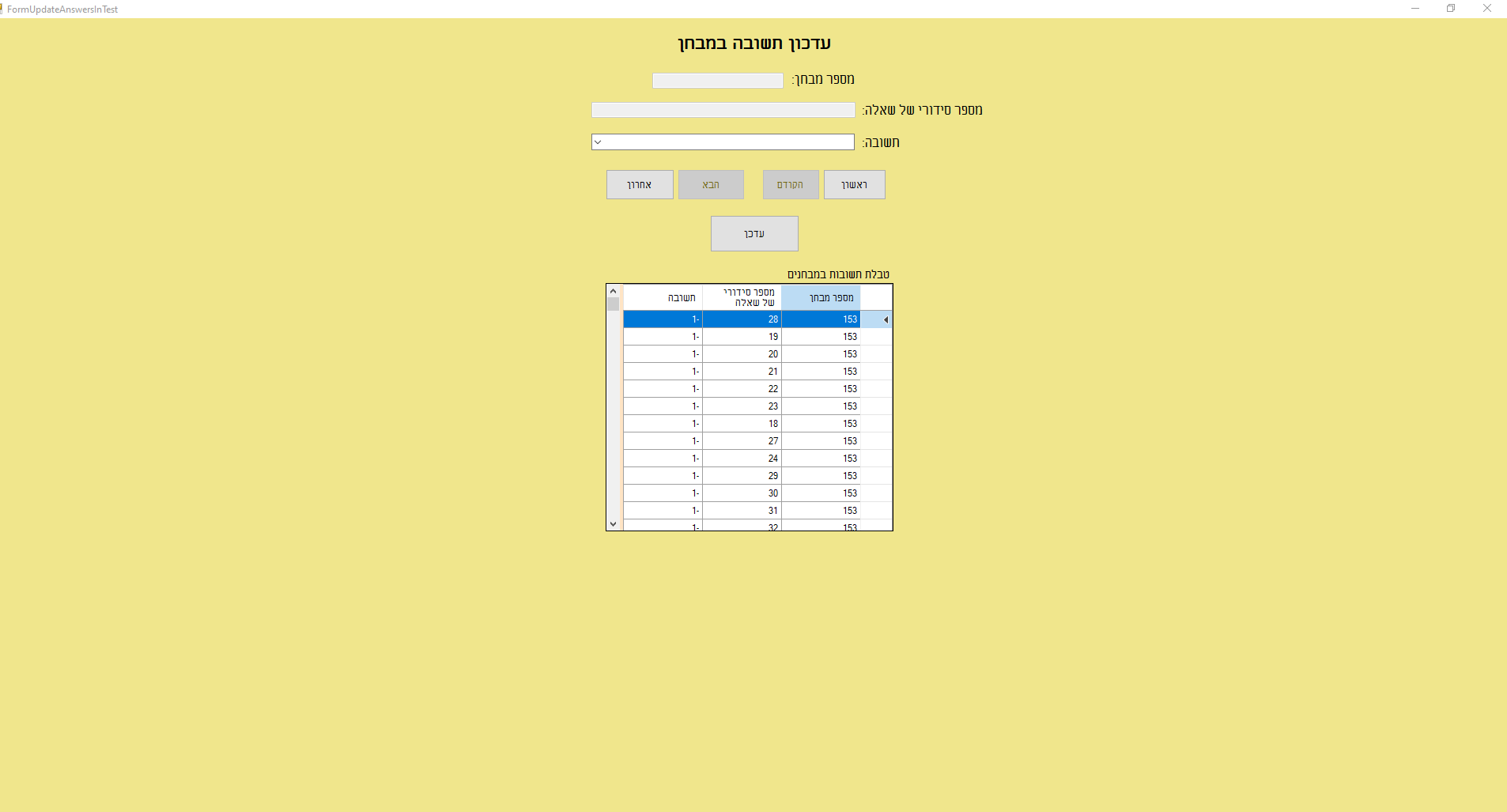
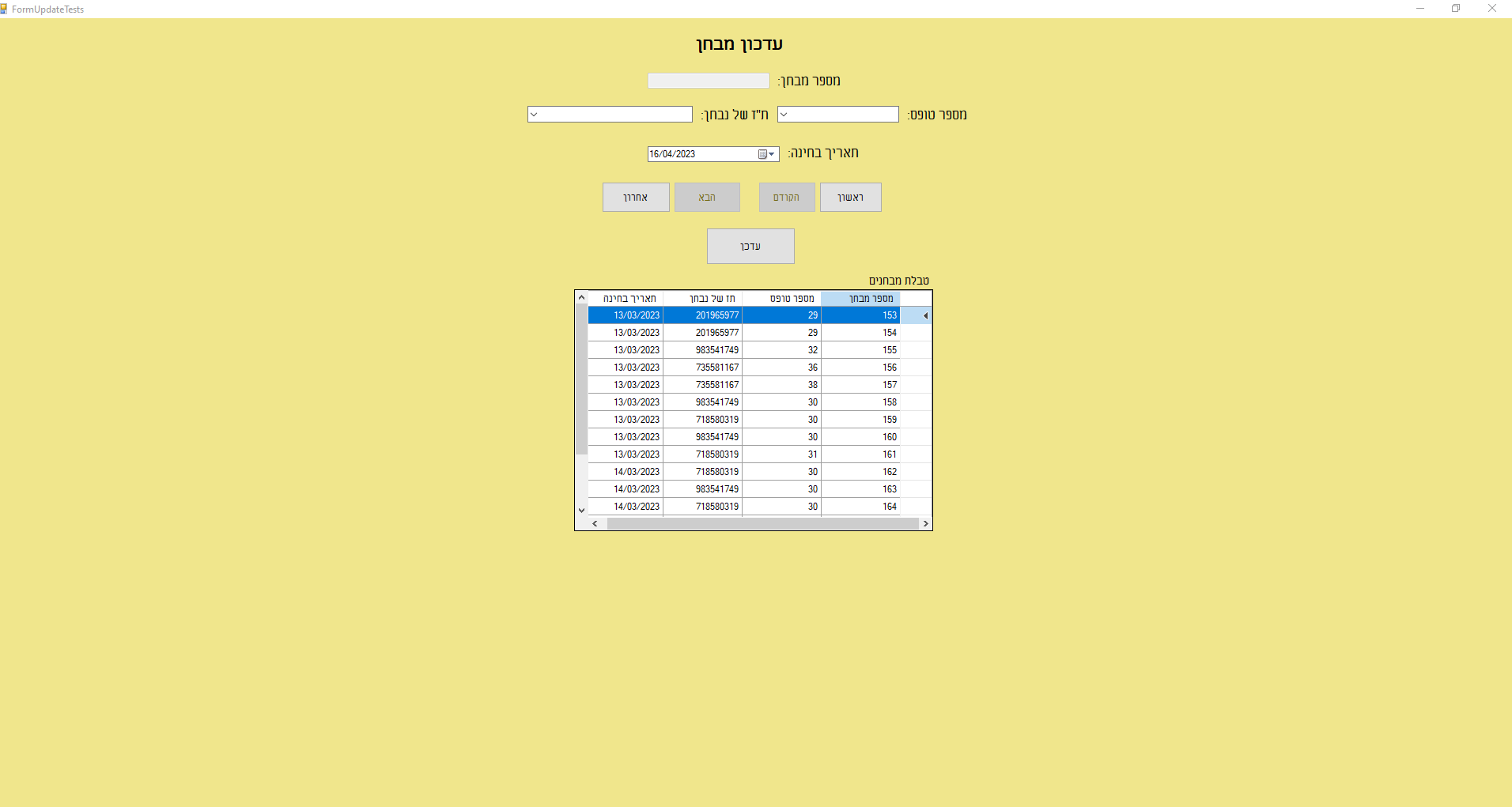
תהליכי ההוספה של המערכת::****

****

**עדכון:**

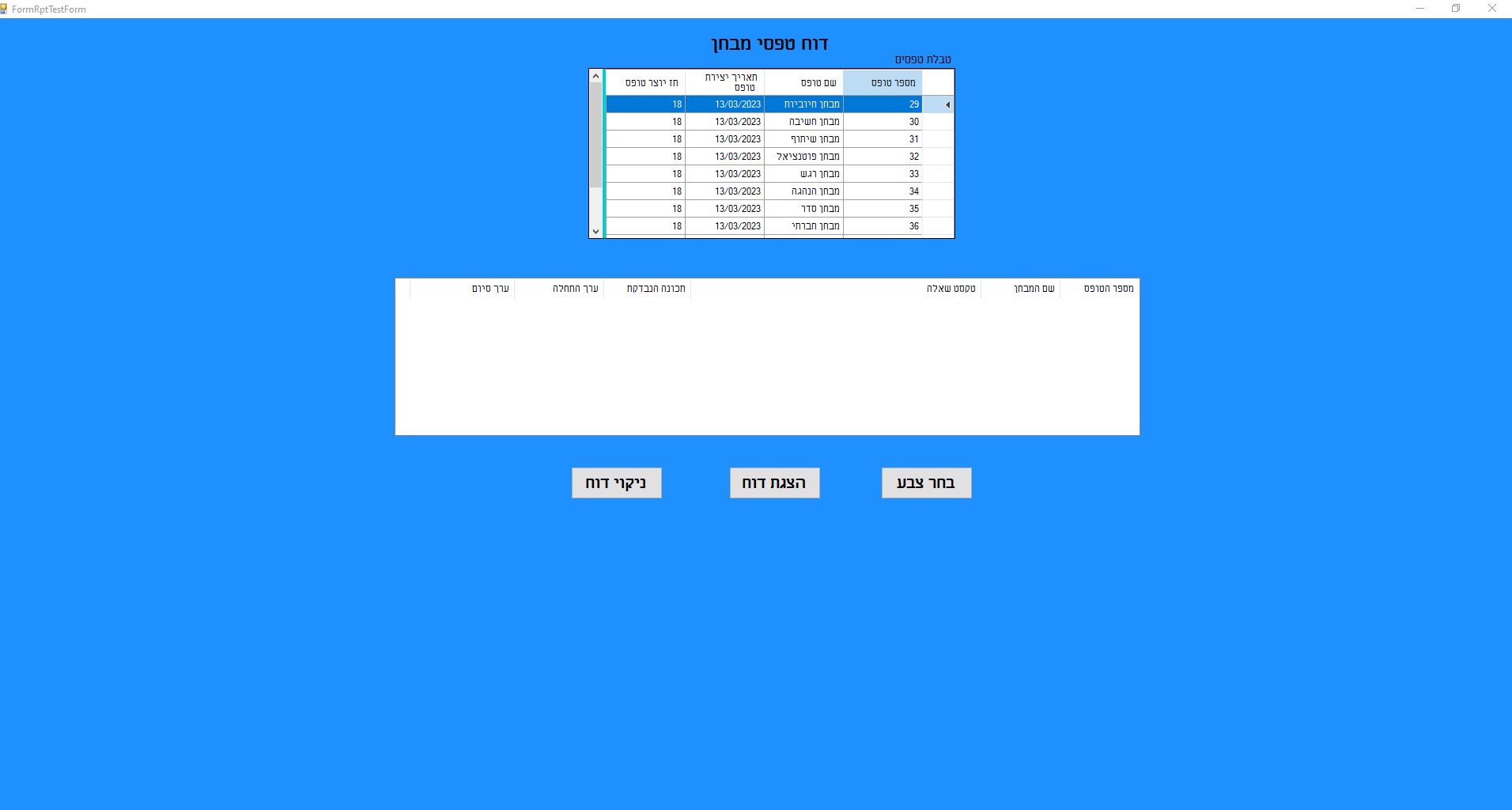
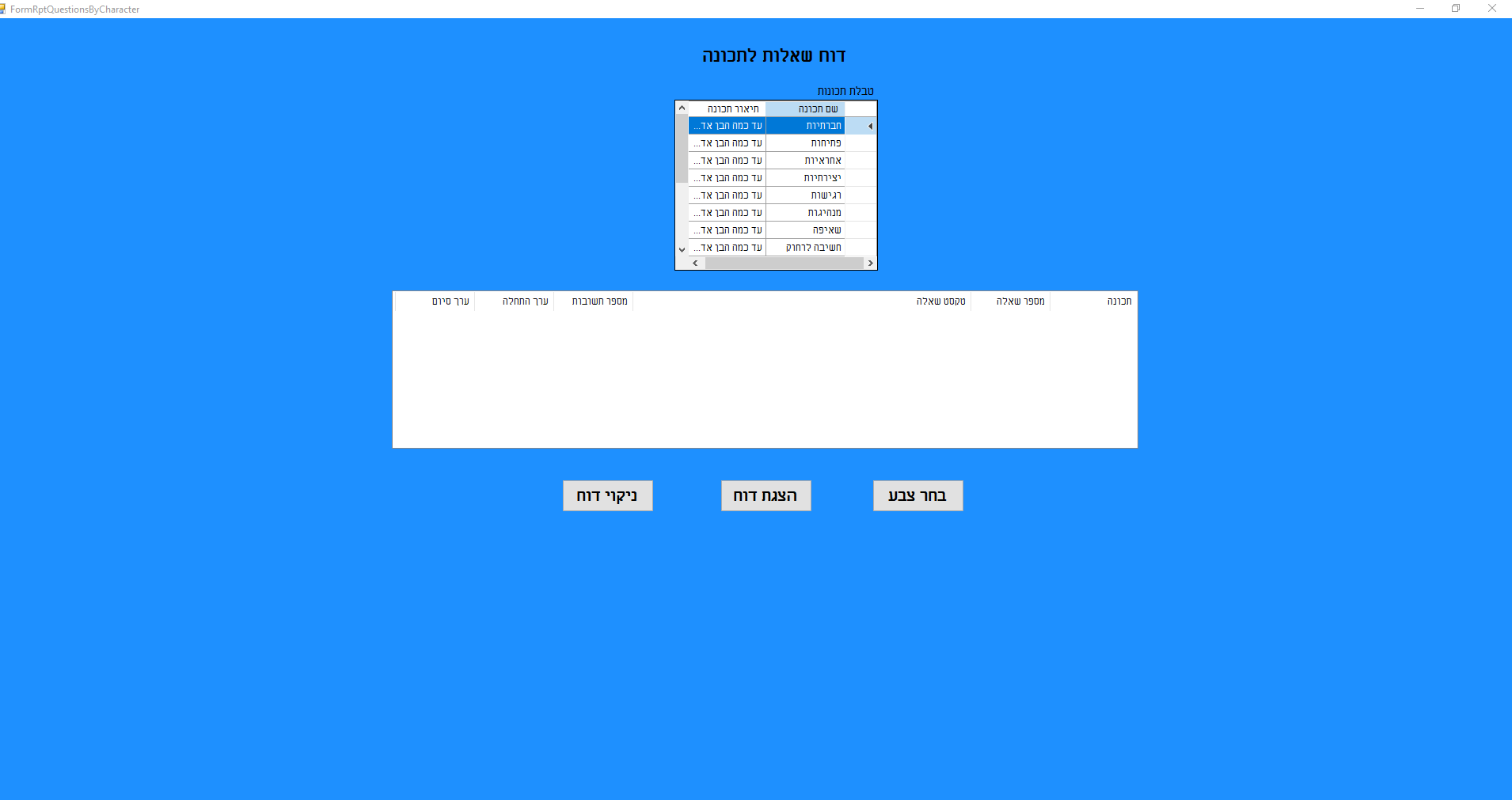
בלחיצה בסרגל התפריטים על "עדכון" יפתחו תהליכי העדכון במערכת, מתוכם משתמש עם גישת מנהל יכול לעדכן מידע לכל אחד מטבלאות המערכת (חוץ מ2 שבהם לא ניתן לשנות משום שהם מכילים רק שדות ייחודים) בצורה פשוטה, וללא צורך הבנת המערכת לעומק, חלק מתהליכי העדכון נותנים לך אפשרות בחירה מתוך אופציות, לדוגמא כשיש צורך בשאלה אז בחירה מתוך השאלות הקימות, וחלק ישנם פשוט שורות ריקות למילוי חופשי, לאחר בחירת העמדה בנתונים שתרצה לשנות בעזרת לחיצה בטבלה או עם כפתורי הניווט, ניתן לאחר מילוי הנתונים ללחוץ "עדכן" והנתונים יתעדכנו.

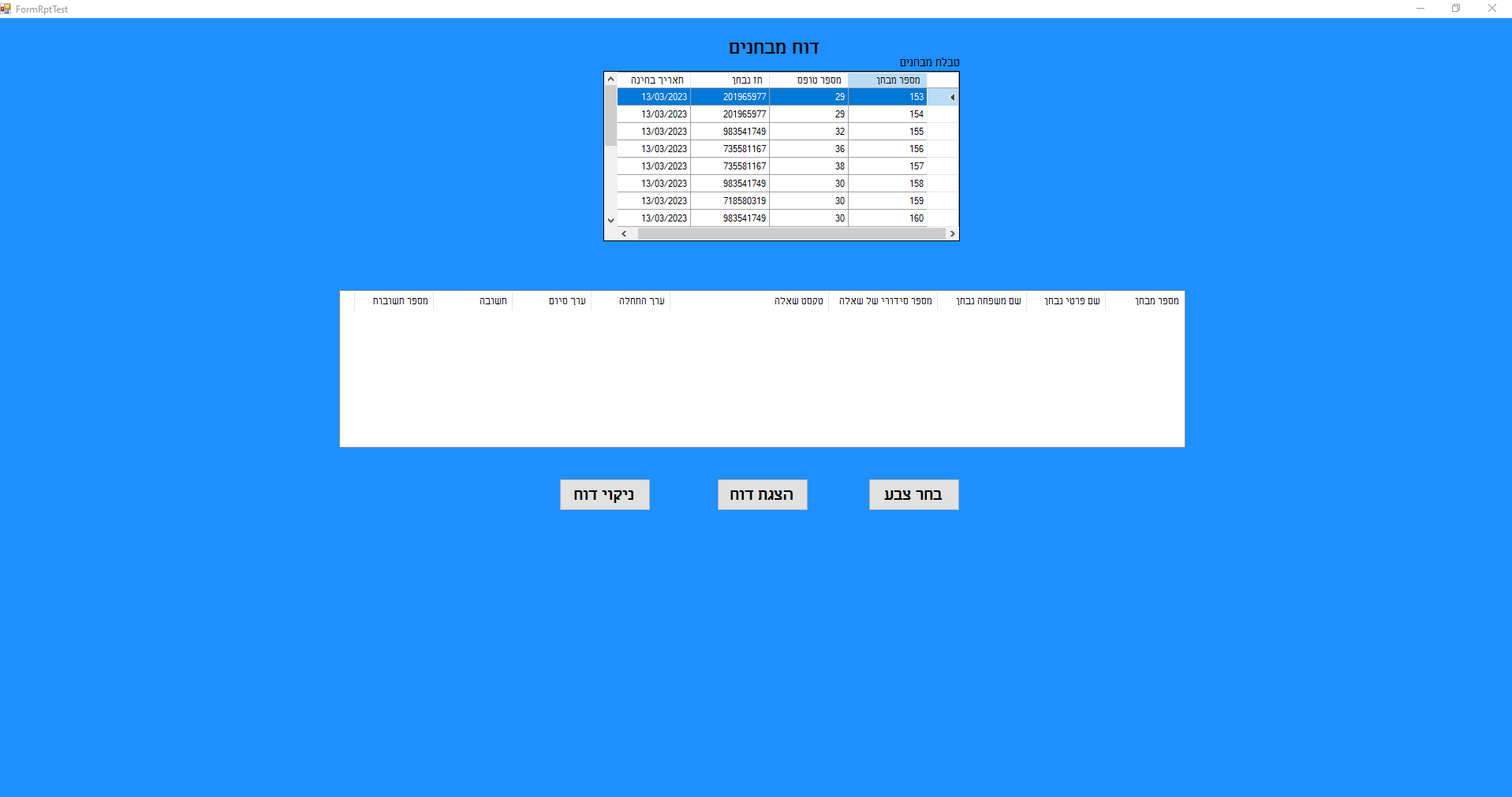
תהליכי העדכון של המערכת:

****

**דוחות:**

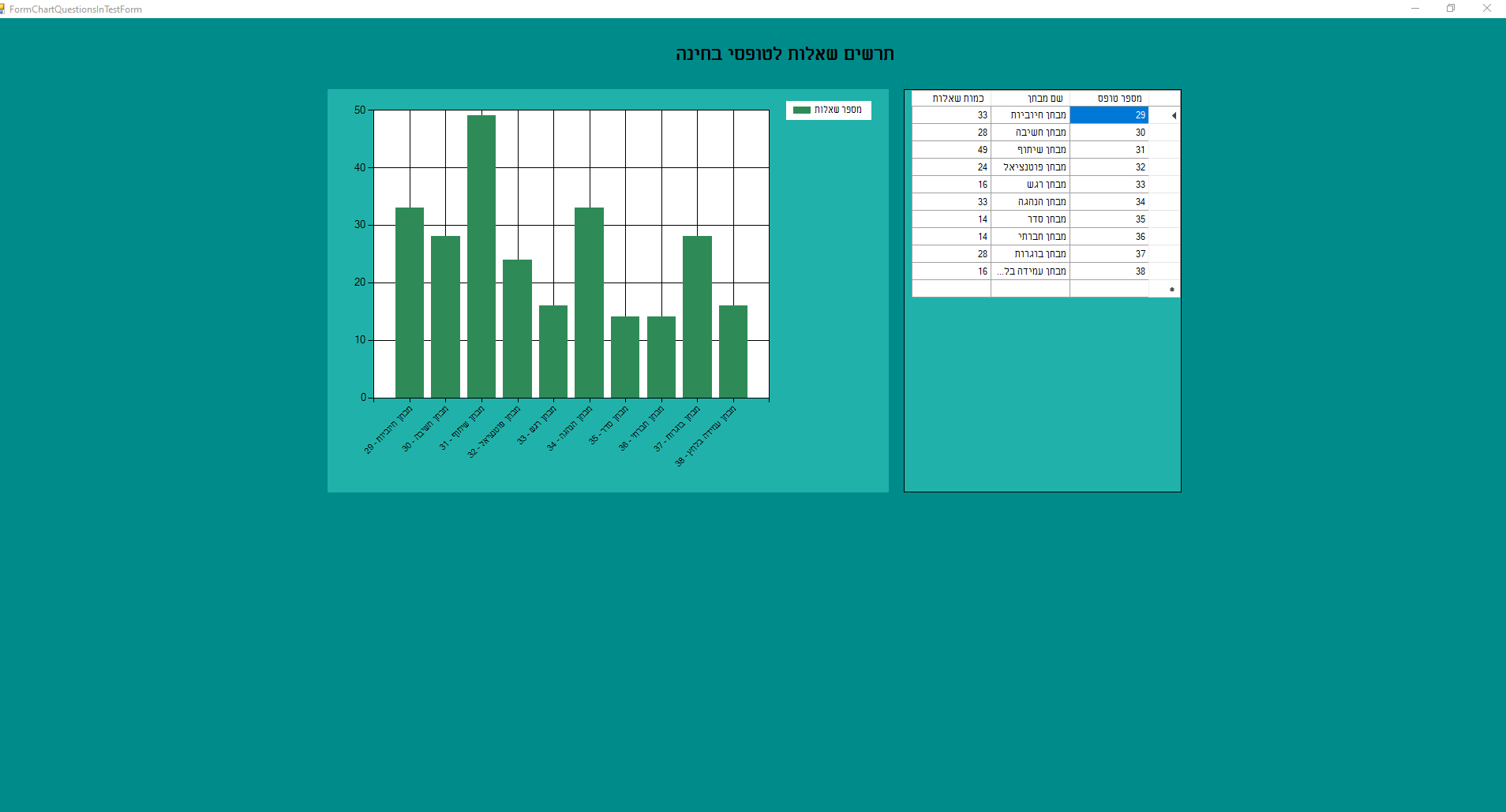
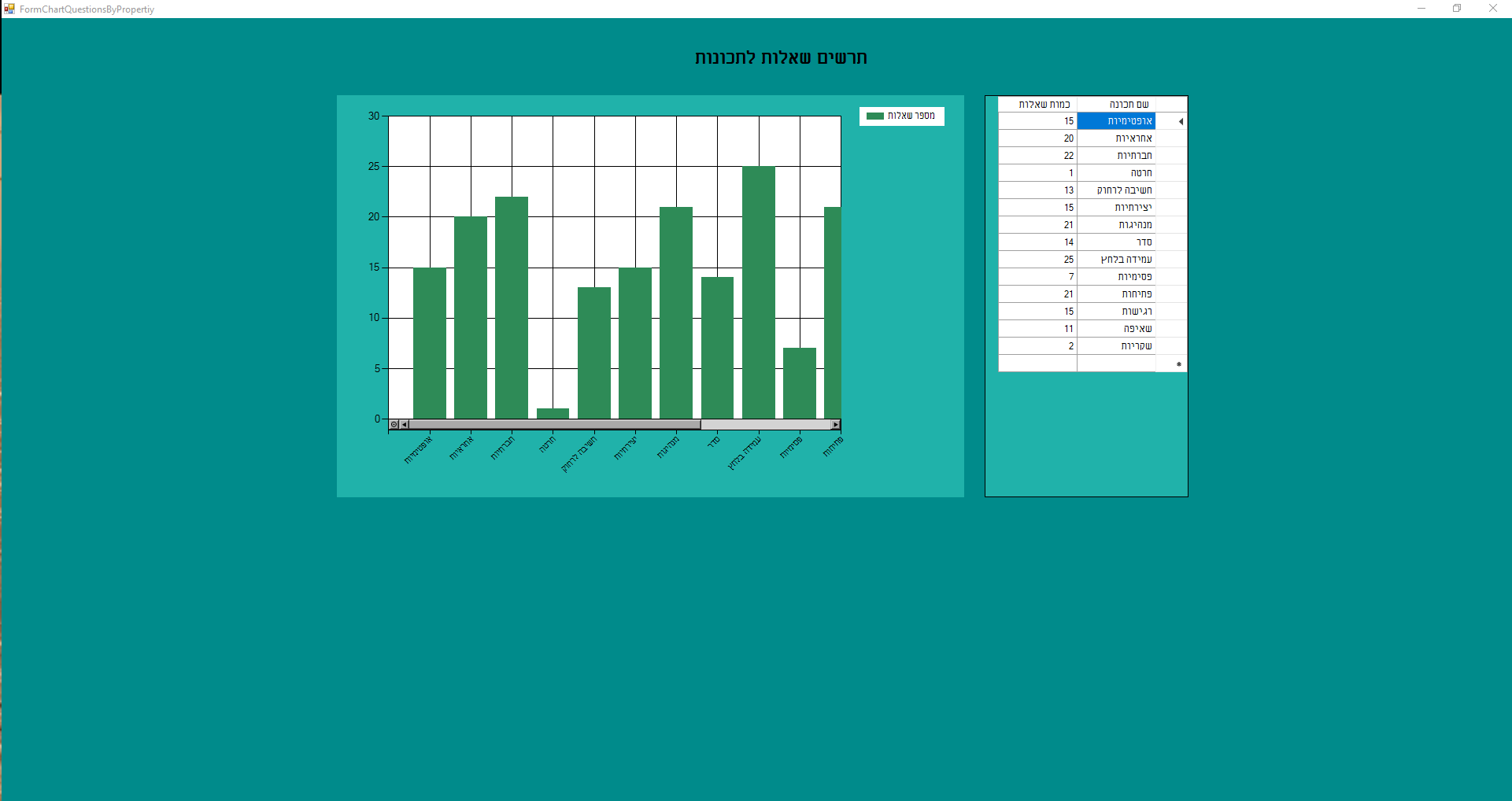
בלחיצה בסרגל התפריטים על "דוחות" יפתחו הדוחות של המערכת, מתוכם משתמש יכול לראות מידע במערכת בצורה נוחה ומובנת, כגון על טפסי המבחן או על המבחנים, ניתן לראות את הנתונים על ידי בחירת העמדה הרצויה לצפייה מהטבלה, ולחיצה על הכפתור "הצגת דוח", ניתן גם לשנות את צבע הדוח על ידי הכפתור "בחר צבע" וגם לנקות את הדוח על ידי הכפתור "ניקוי דוח".

הדוחות של המערכת:****

****

**תרשימים:**

בלחיצה בסרגל התפריטים על "תרשימים" יפתחו התרשימים של המערכת, מתוכם משתמש יכול לראות מידע במערכת בצורה נוחה ומובנת, כגון על כמות שאלות שיש לכל תכונה או על כמות שאלות שיש בטפסי מבחן, בצורה נוחה וויזואלית בעזרת דיאגרמה, ניתן לראות את הנתונים גם בטבלה הנמצאת בטופס וגם בעזרת הדיאגרמה המראה באופן מופשט את המידע.

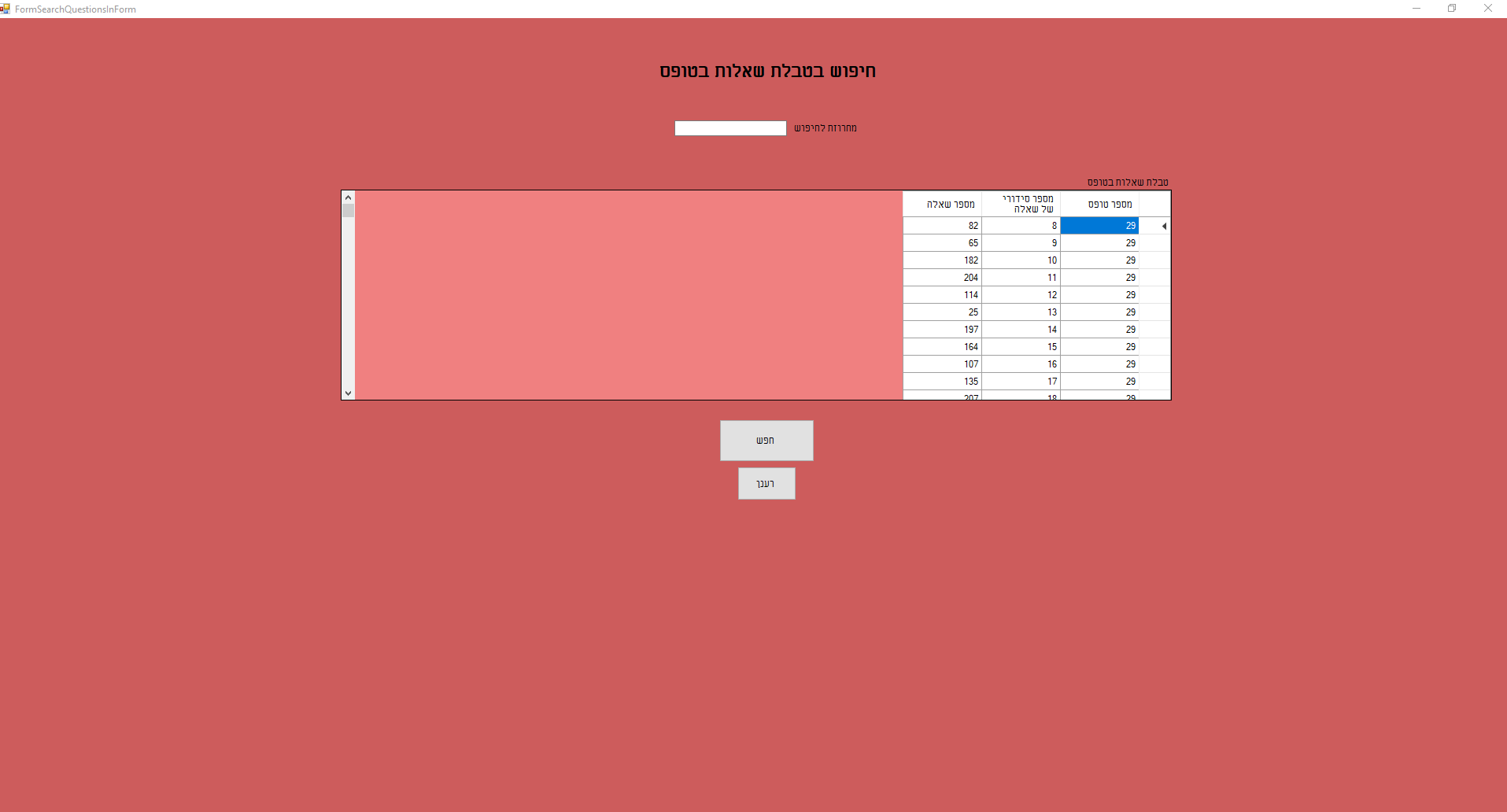
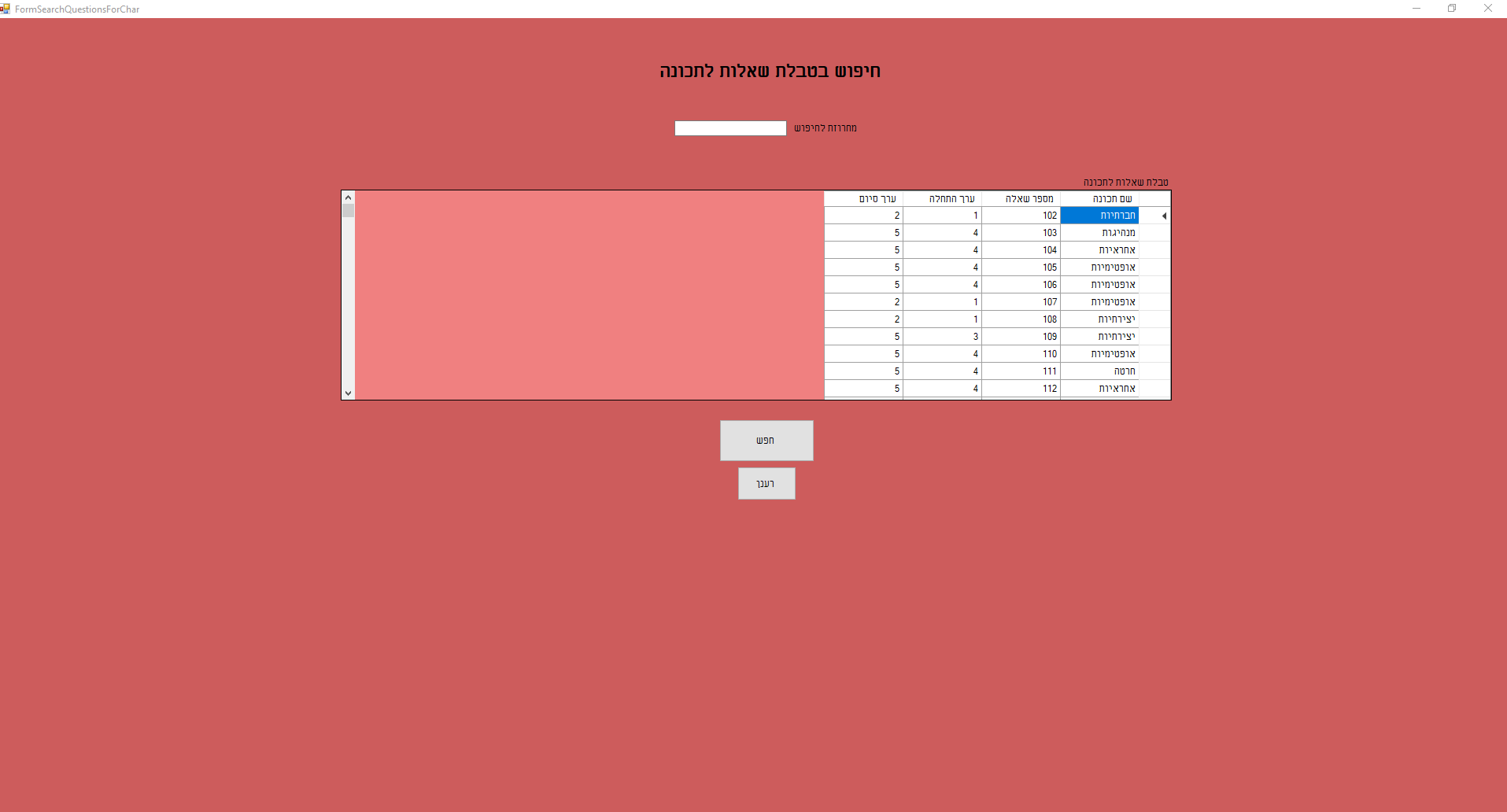
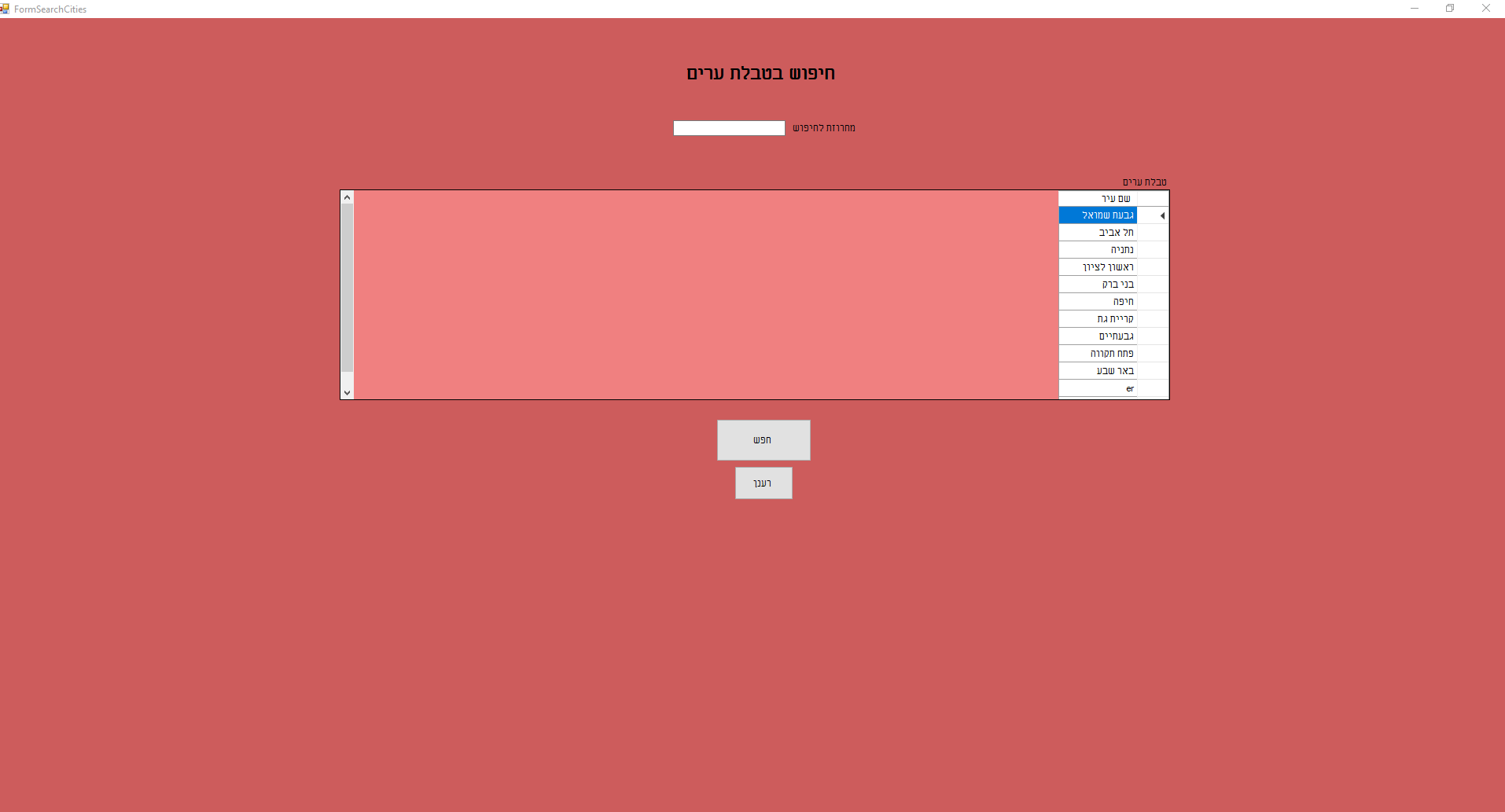
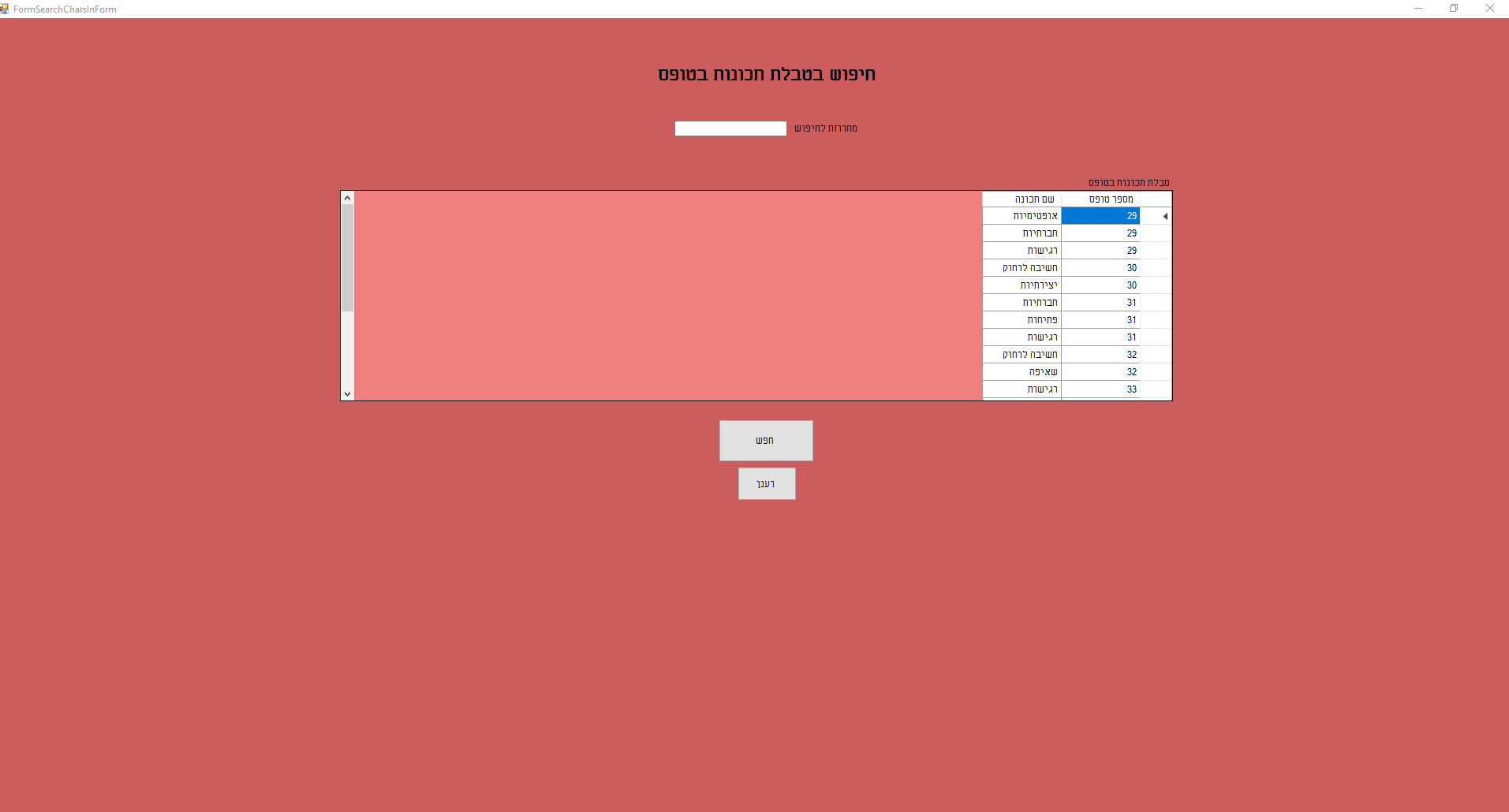
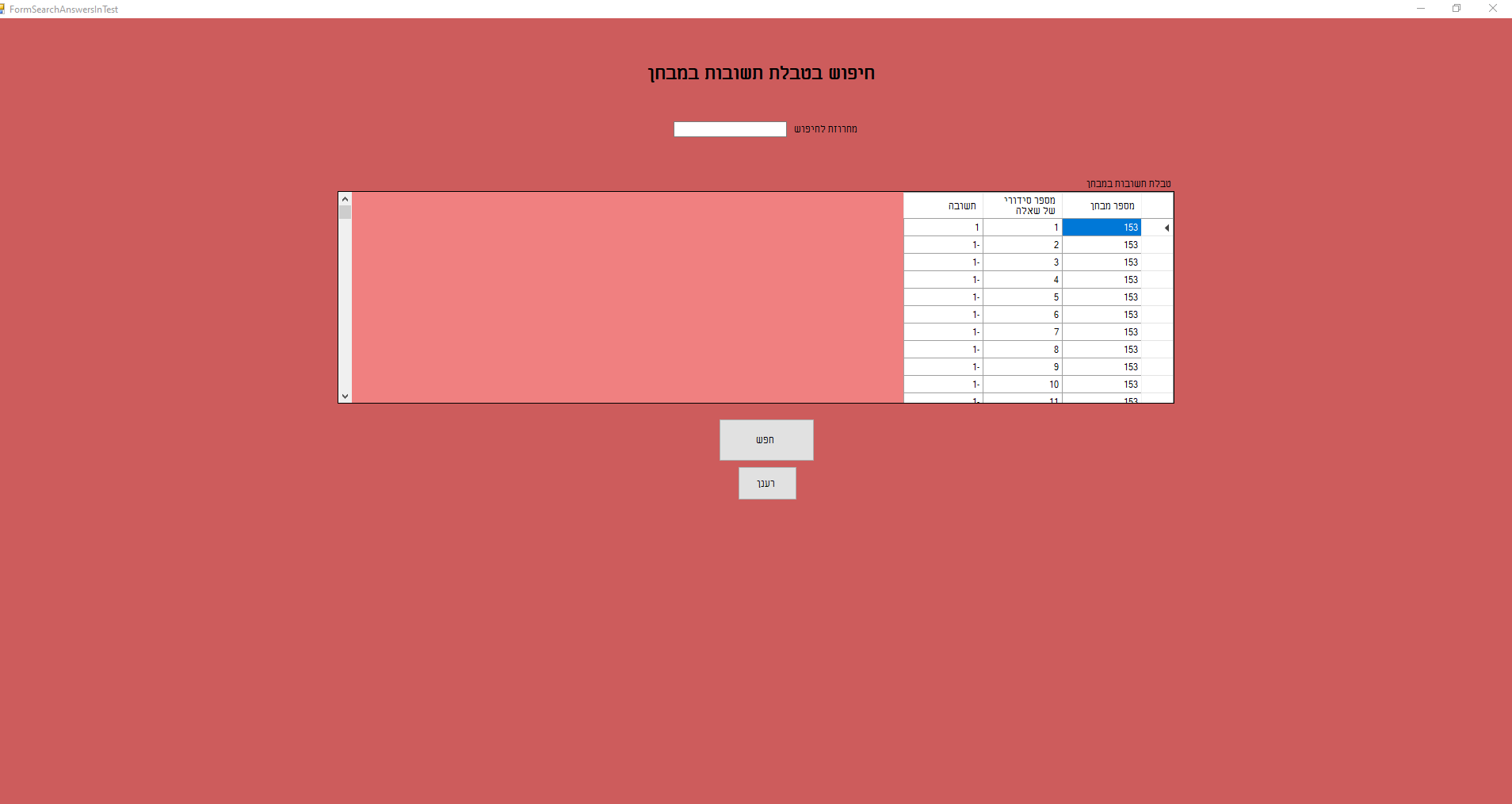
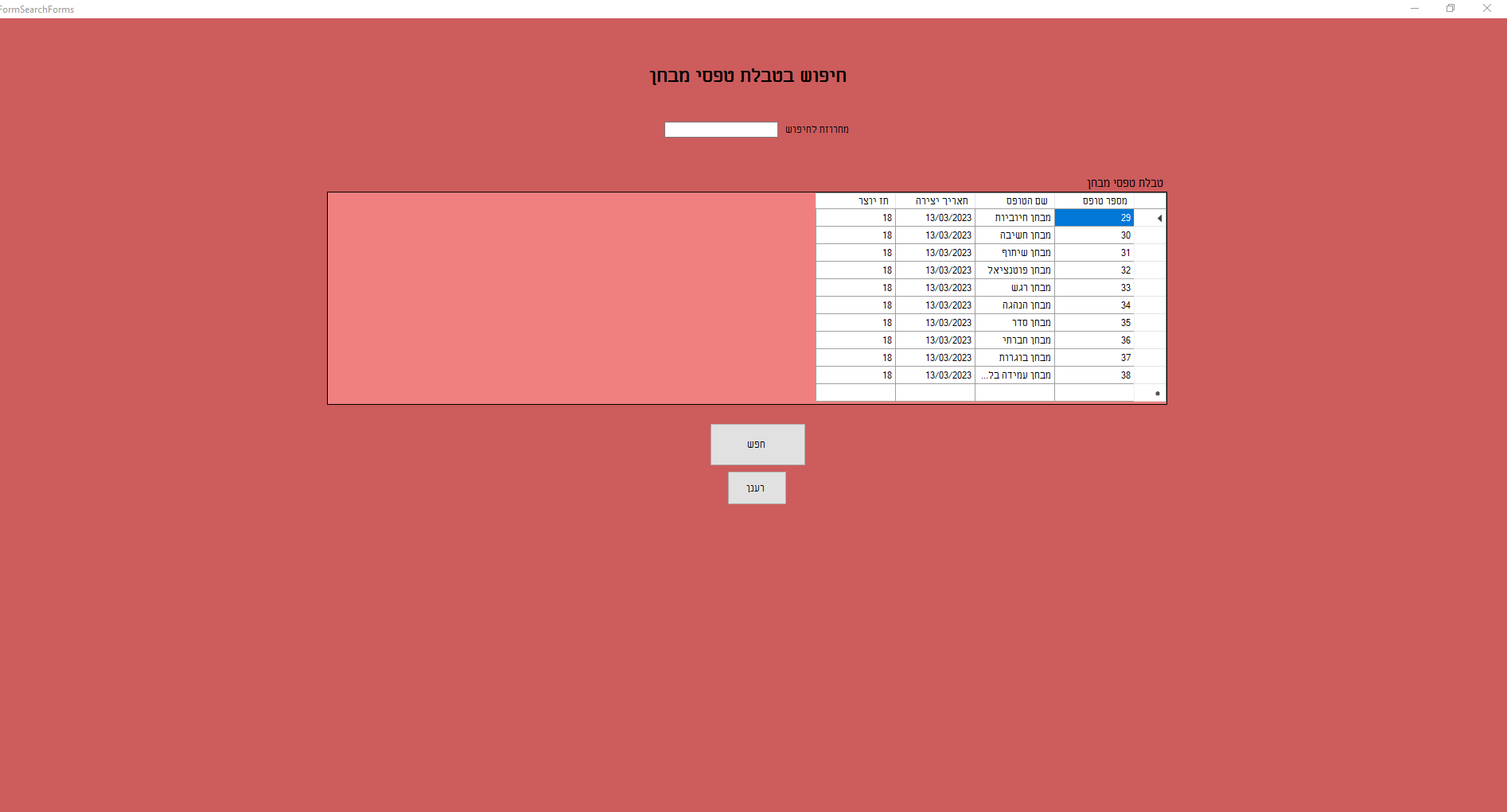
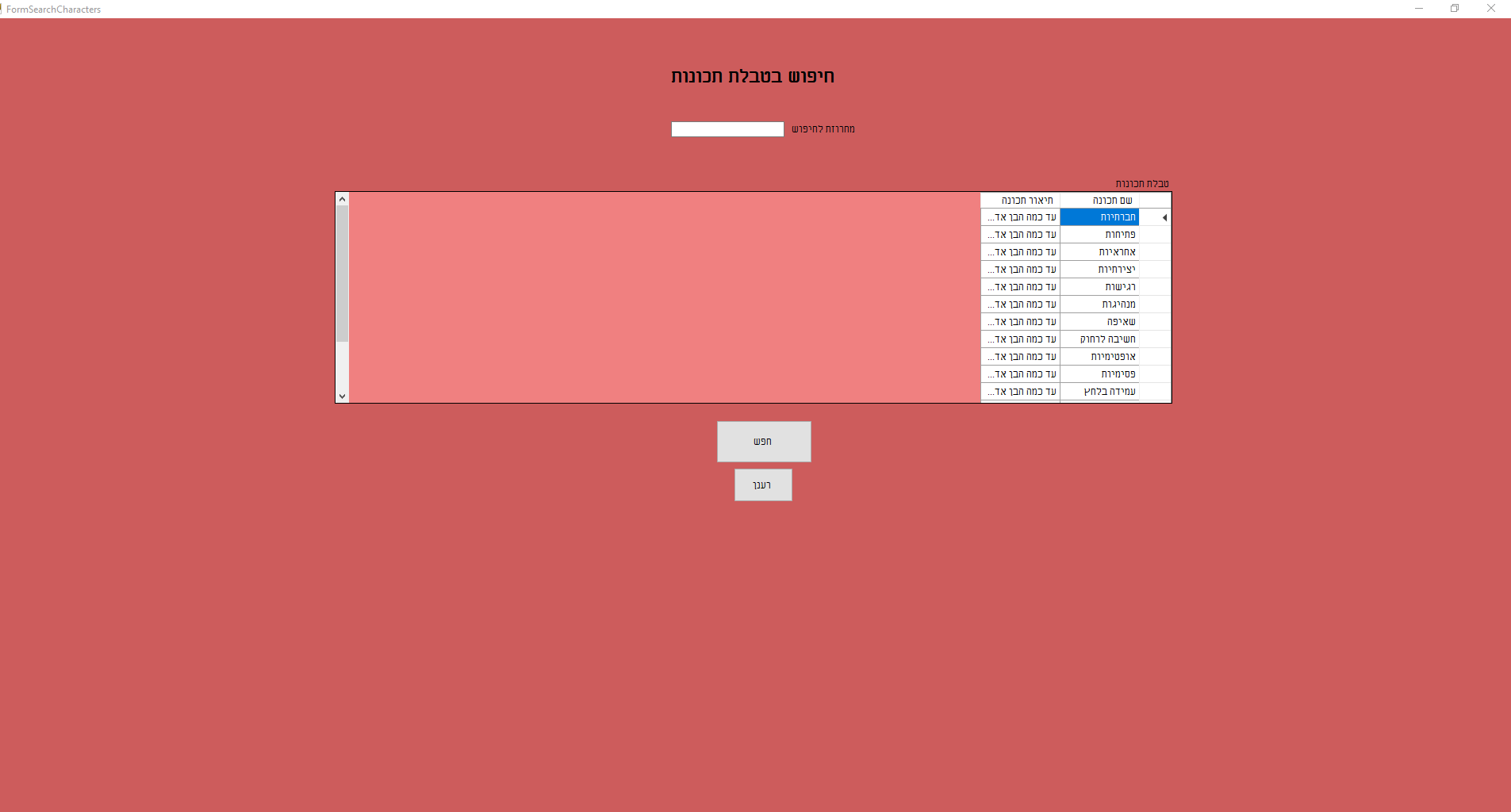
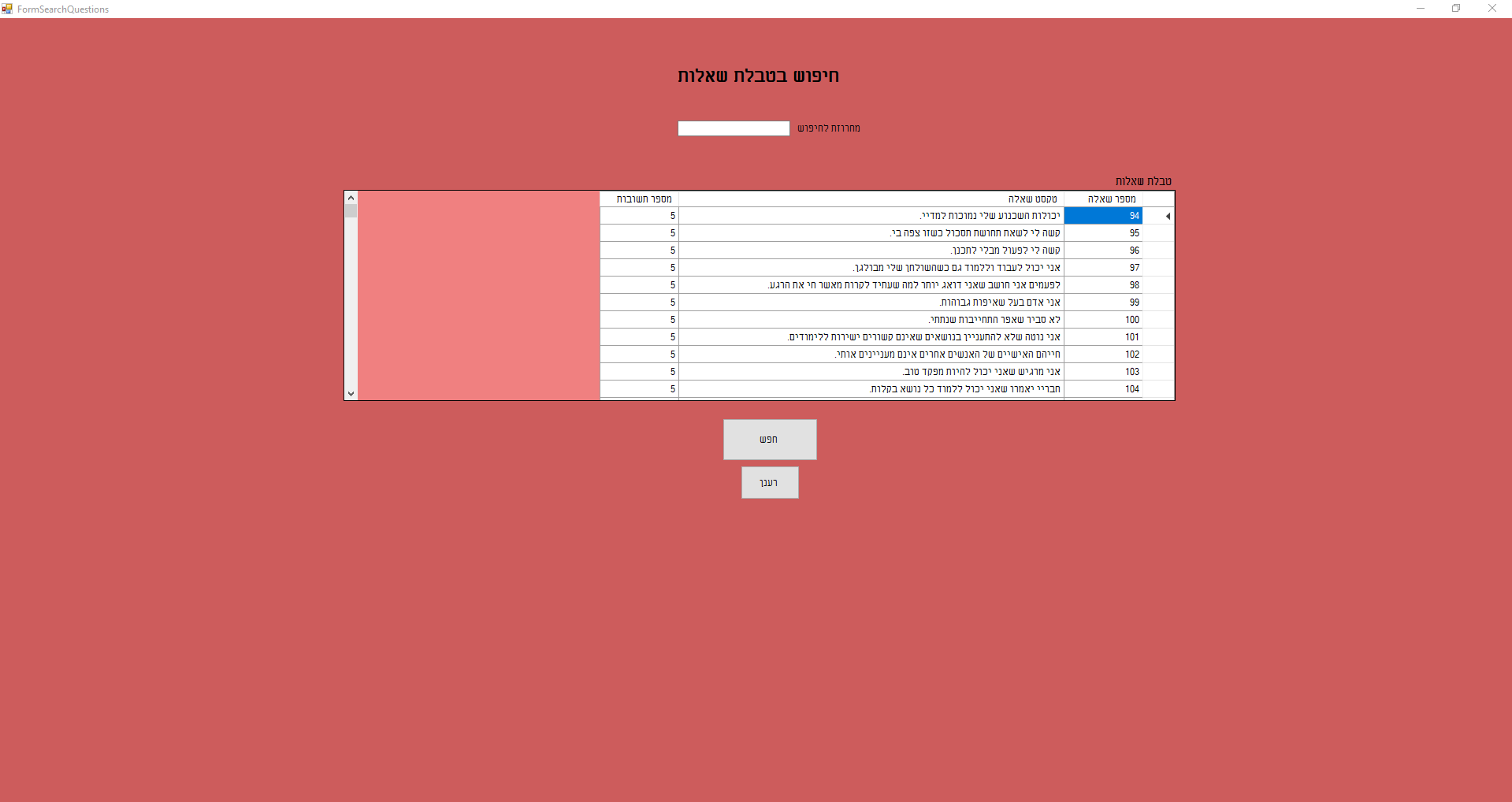
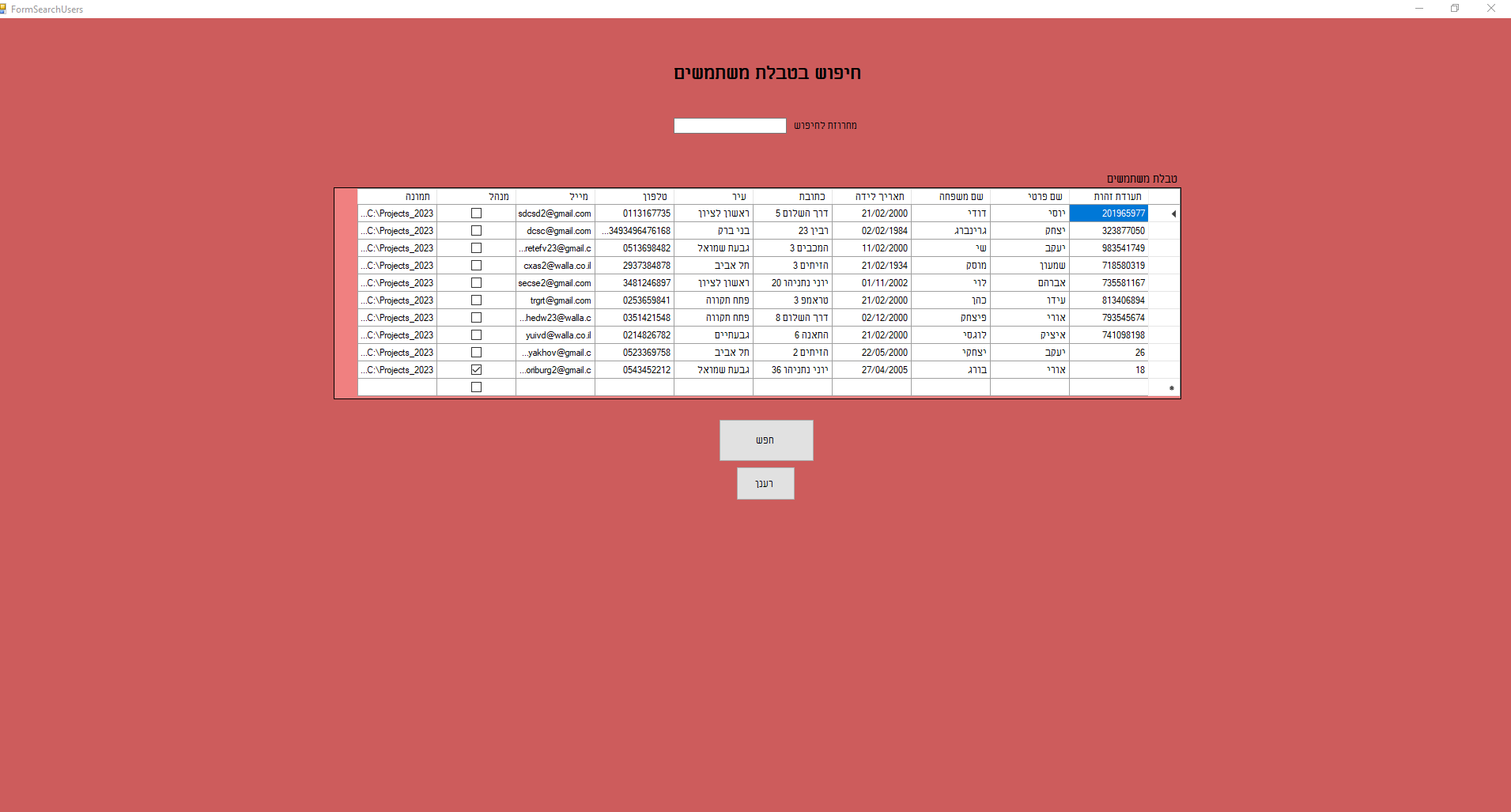
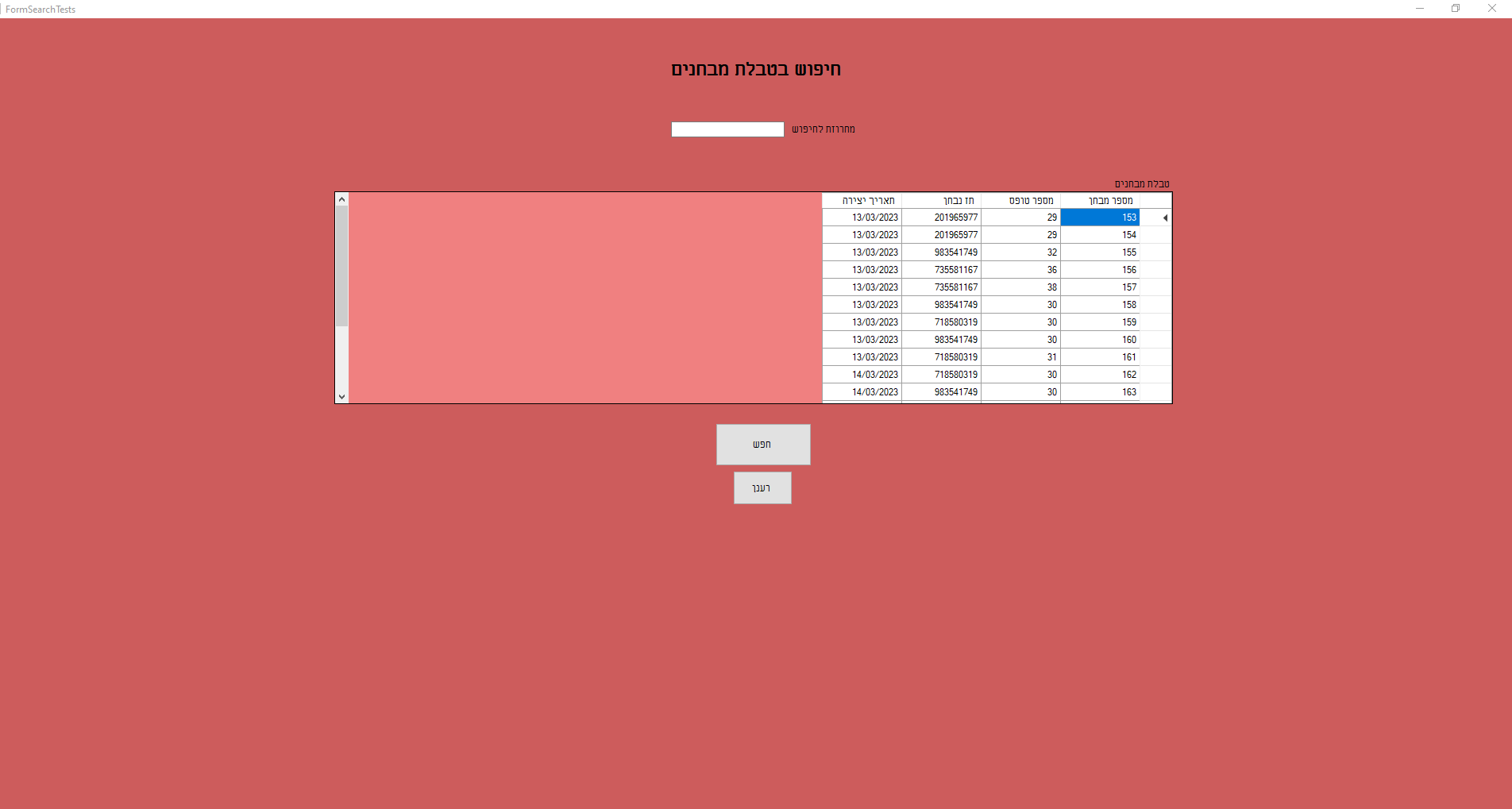
התרשימים של המערכת:****

****

**חיפוש:**

בלחיצה בסרגל התפריטים על "חיפוש" יפתחו תהליכי החיפוש של המערכת, מתוכם משתמש יכול לחפש בכל אחת מהטבלאות של המערכת מידע שהוא מחפש על ידי כתיבת אחד מהנתונים שהוא יודע, לדוגמא חיפוש משתמש על ידי כתיבת כתובת המייל שלו, ניתן לחפש על ידי כתיבת המחרוזת לחיפוש בתיבת הטקסט הנמצאת לכך, ולחיצה על כפתור "חפש", ניתן גם ללחוץ על כפתור "רענן" בשביל לראות את כל הטבלה ללא סינון החיפוש המחרוזת שהוכנסה.

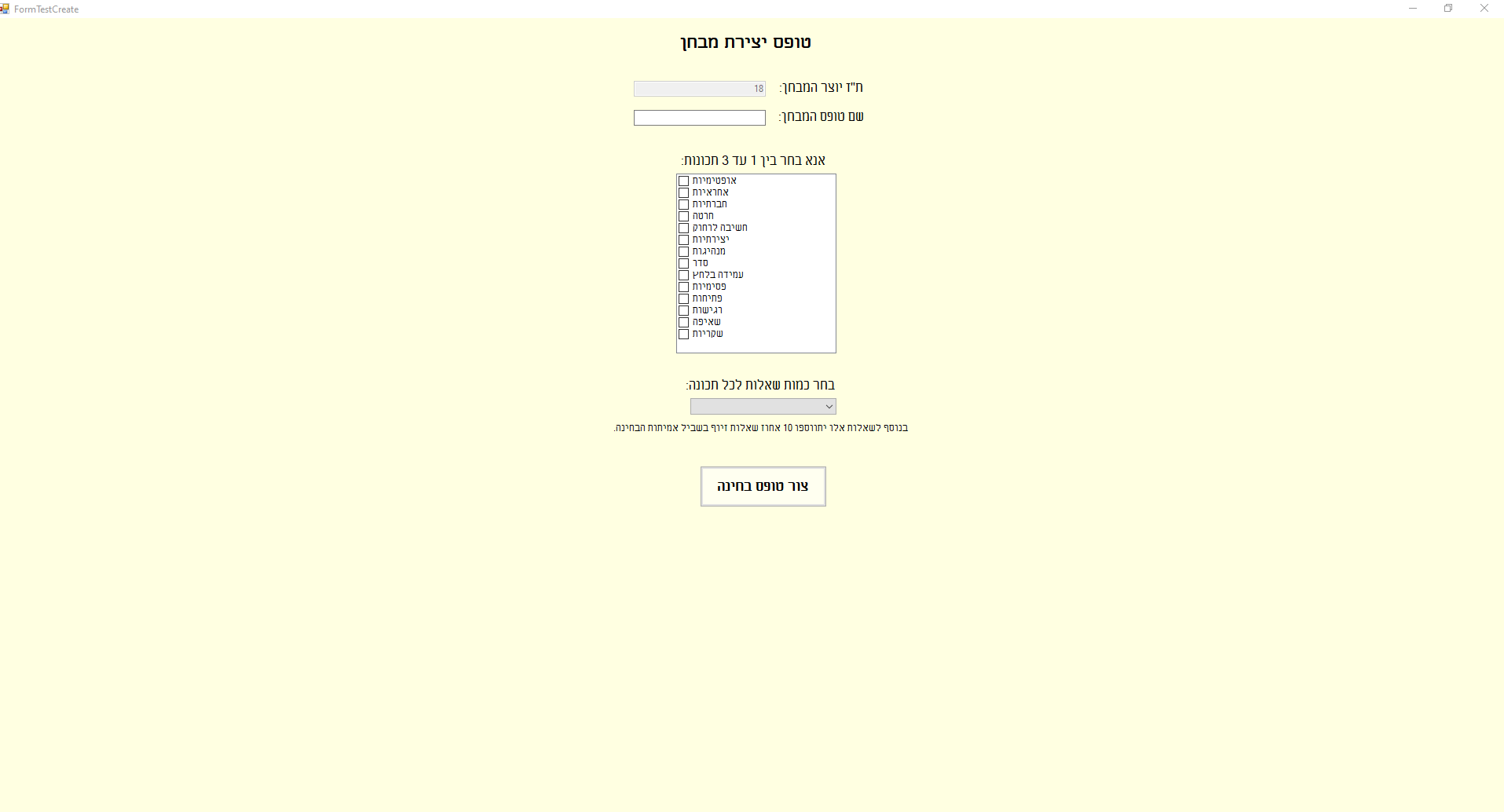
תהליכי החיפוש של המערכת:

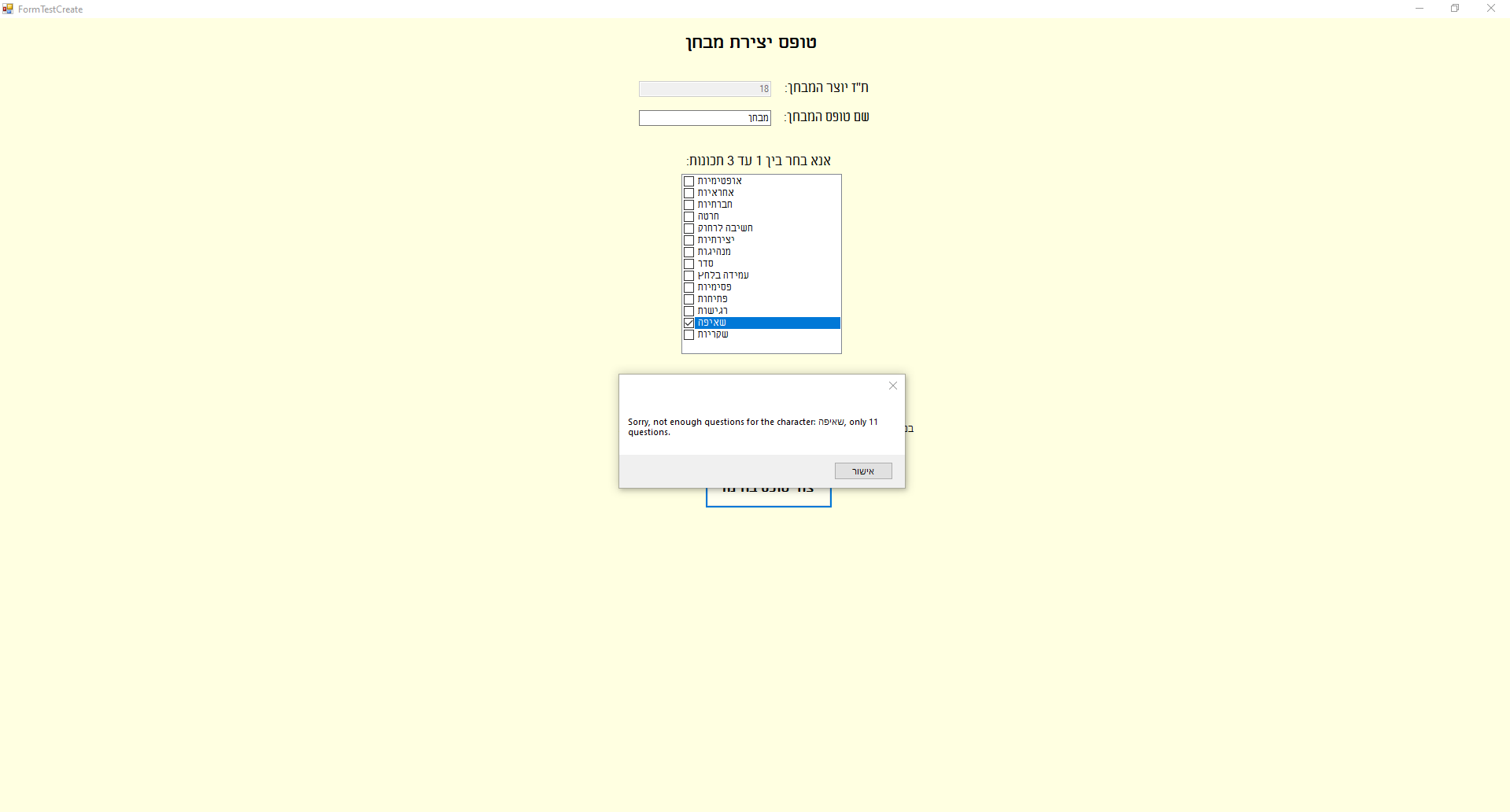
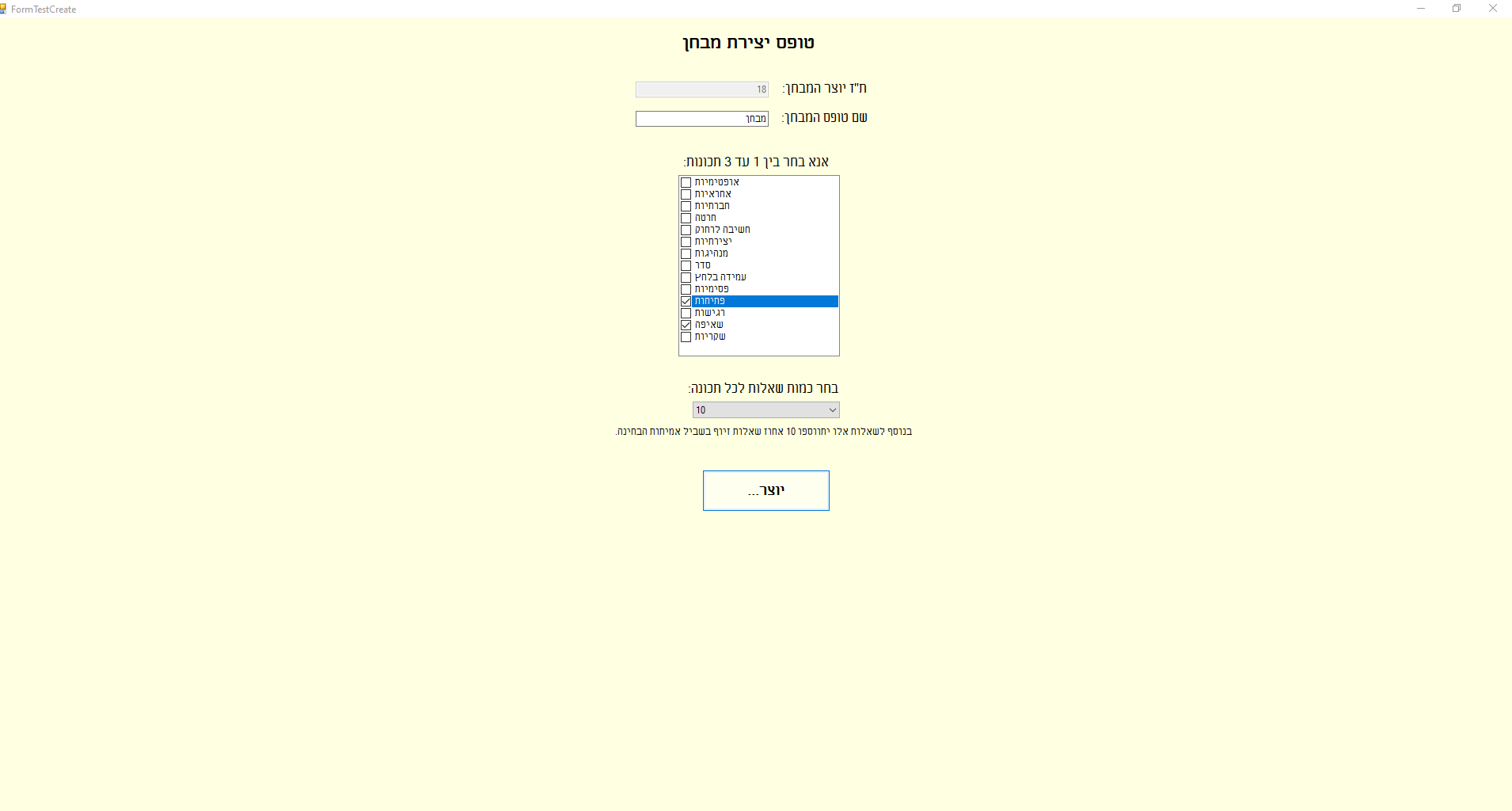


**יצירת מבחן:**

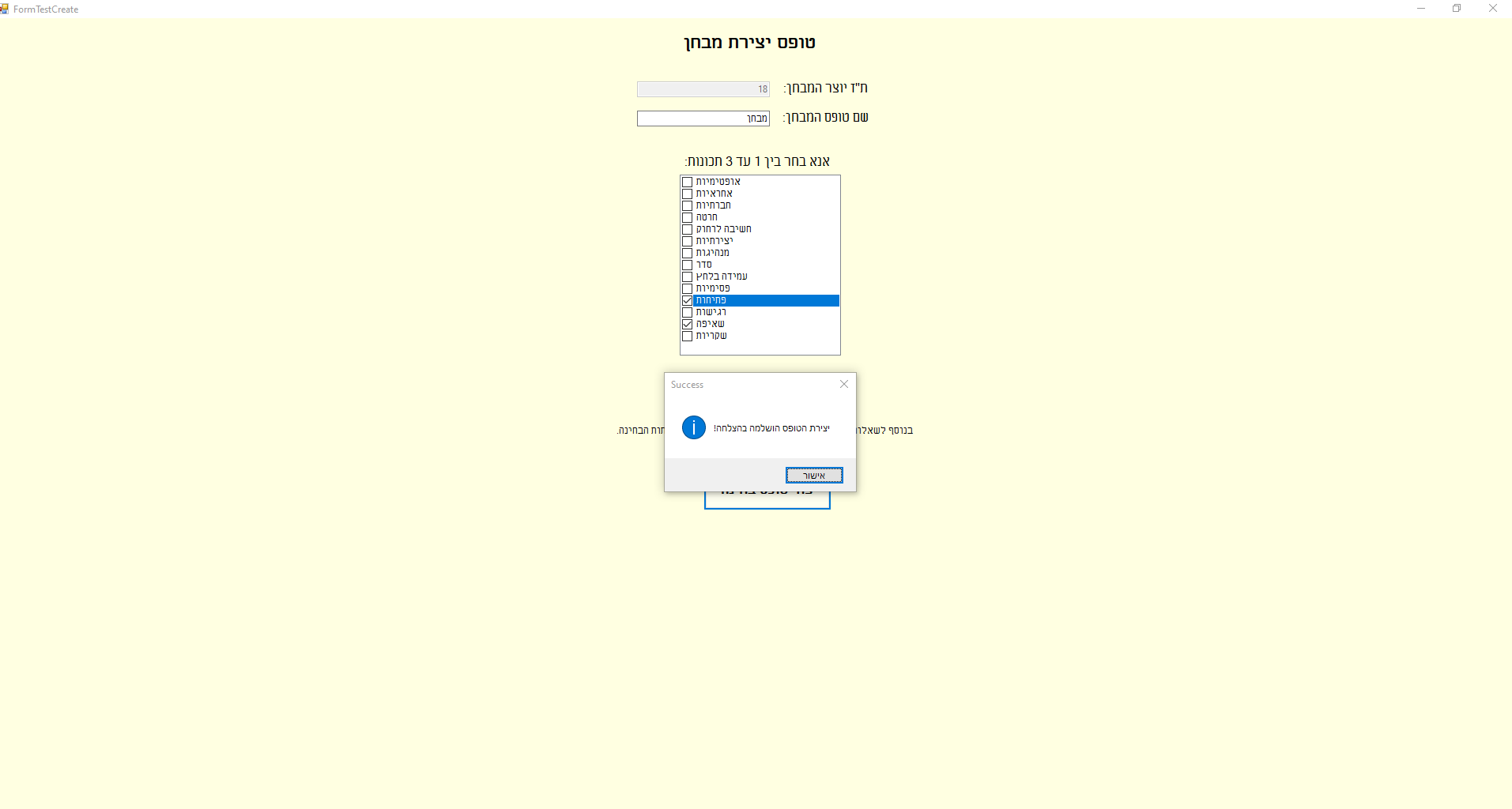
בלחיצה בסרגל התפריטים על "בחנים" ואז לחיצה על "יצירת מבחן" יפתח תהליך יצירת טופס מבחן, בעזרתו ניתן בפשטות ליצור מבחן חדש במערכת על ידי קריטריונים שהמשתמש היוצר בוחר, ניתן ליצור את המבחן על ידי מילוי שם טופס המבחן, ולאחר מכן לבחור בין 1 ל3 תכונות מהתכונות הקימות במערכת שאותם המבחן יבדוק, ובנוסף לבחור כמות שאלות שתהיה לכל תכונה, לכמות הזו תוסף 10 אחוז שאלות מזויפות על מנת להביא יותר אמיתות לבחינה (השאלות המזויפות הלא קשורות אינם יוכללו בציון), ואז ניתן ליצור את טופס המבחן על ידי כפתור "צור טופס בחינה", במידה ואין מספיק שאלות לתכונה מסוימת יהיה הודעה על כך ותצטרך לבחור או פחות שאלות לפי הכמות הקיימת במערכת, להוסיף עוד שאלות ואז לחזור, או לבחור תכונה אחרת עם יותר שאלות שקימות לה במערכת.

טופס המבחן שייווצר על ידי המערכת יכיל שאלות רנדומליות בסדר רנדומלי לפי הקריטריונים שהמשתמש ביקש.

מסך תהליך יצירת המבחן של המערכת:****

במידה ואין מספיק שאלות לאחת מהתכונות הנבחרות תופיע הודעה כזו ולא ייווצר טופס:**** בזמן יצירת המבחן הכפתור ישנה שם ל"יוצר..." ויהיה המסך הבא: ****

לאחר יצירת הטופס ההודעה הבאה תופיע:

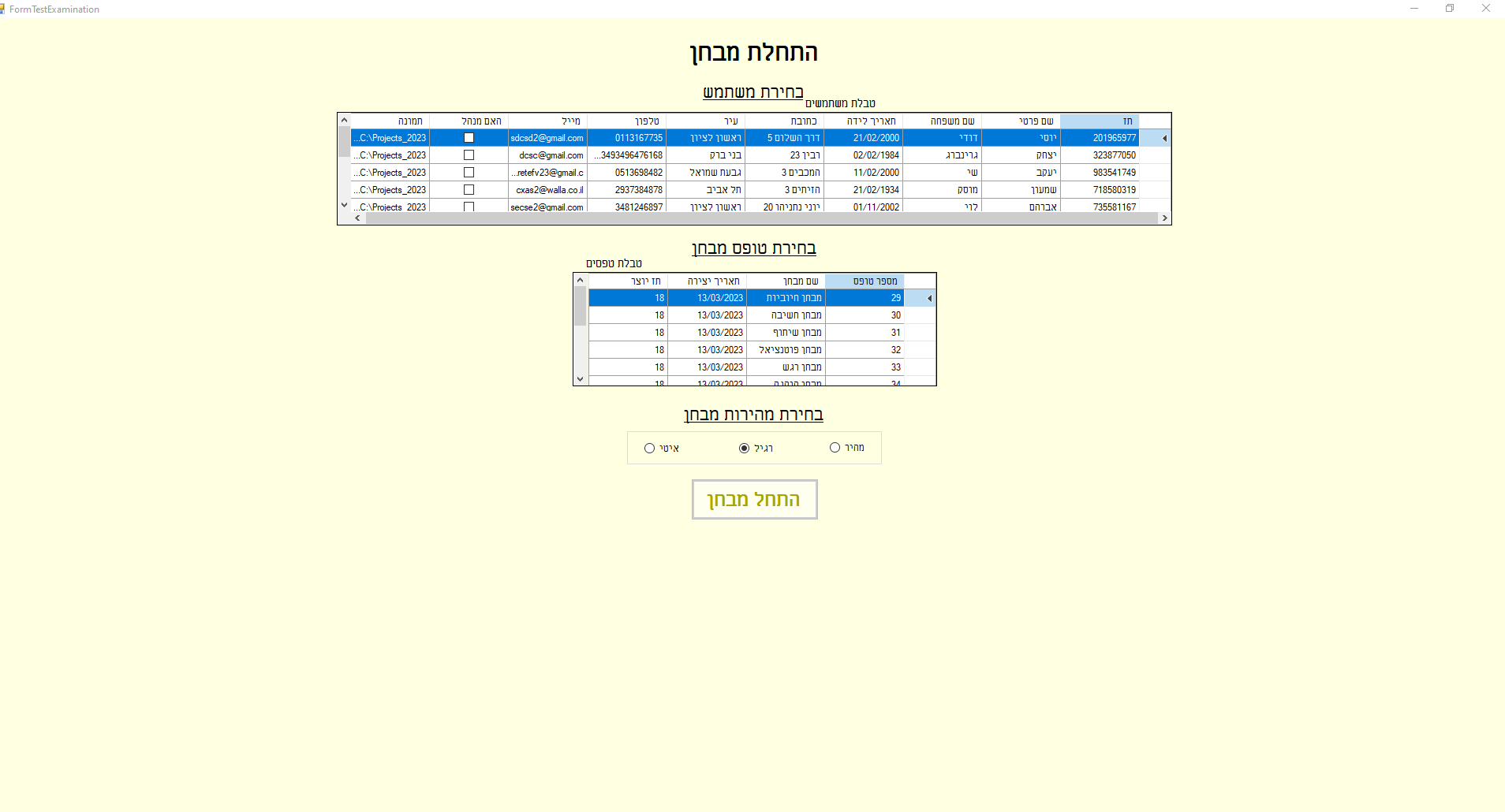
****

**התחלת מבחן:**

בלחיצה בסרגל התפריטים על "בחנים" ואז לחיצה על "עשיית מבחן" יפתח תהליך התחלת המבחן, בו על מנת להתחיל את המבחן המשתמש צריך לבחור את עצמו מתוך טבלת המשתמשים ולאחר מכן את טופס המבחן אותו ירצה לבצע מתוך טבלת הטפסים, ולבסוף את מהירות המבחן שבה ירצה לבצע את המבחן, שתקבע כמה מהר צריך לענות על כל שאלה בין האופציות- מהיר, רגיל ואיטי.

לאחר מכן יש ללחוץ על כפתור "התחל מבחן" ומבחן חדש ייווצר והמסך יעבור אל תהליך ביצוע המבחן.

מסך התחלת המבחן:

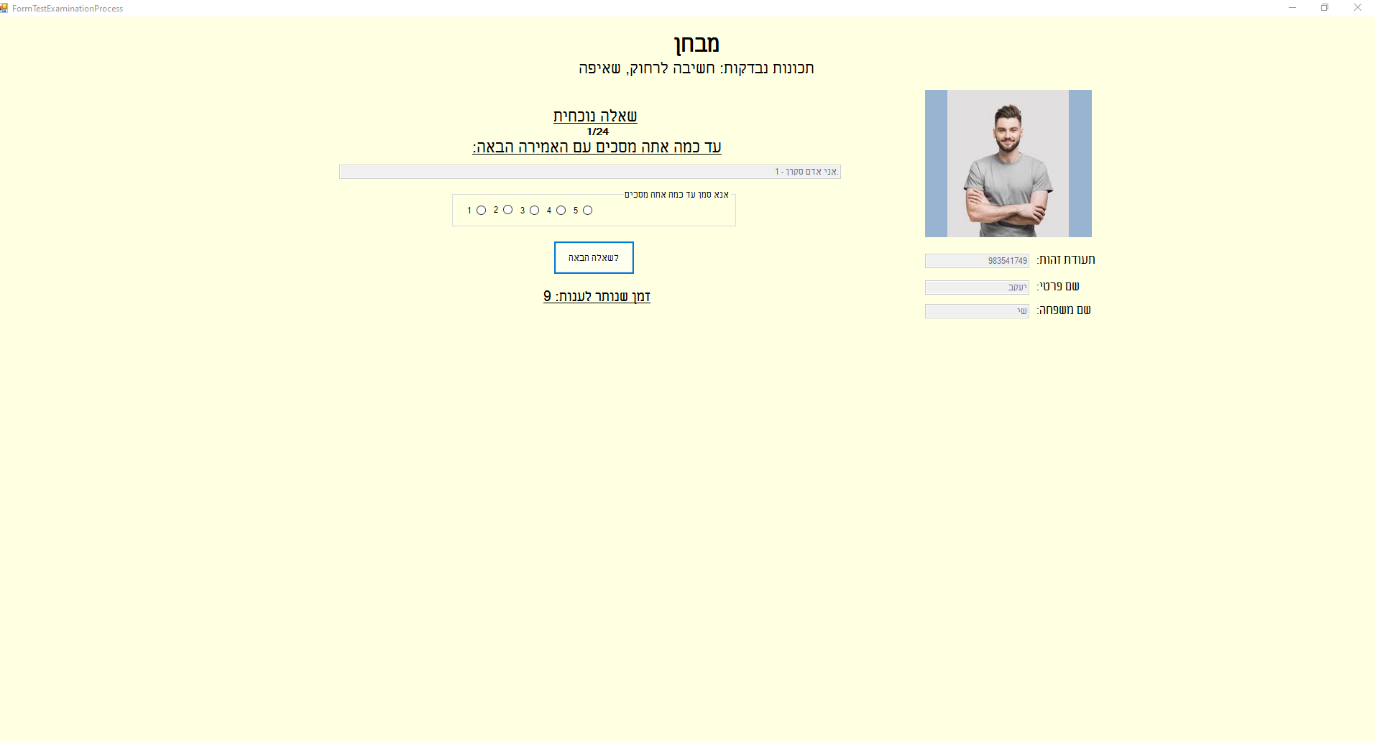


**ביצוע מבחן:**

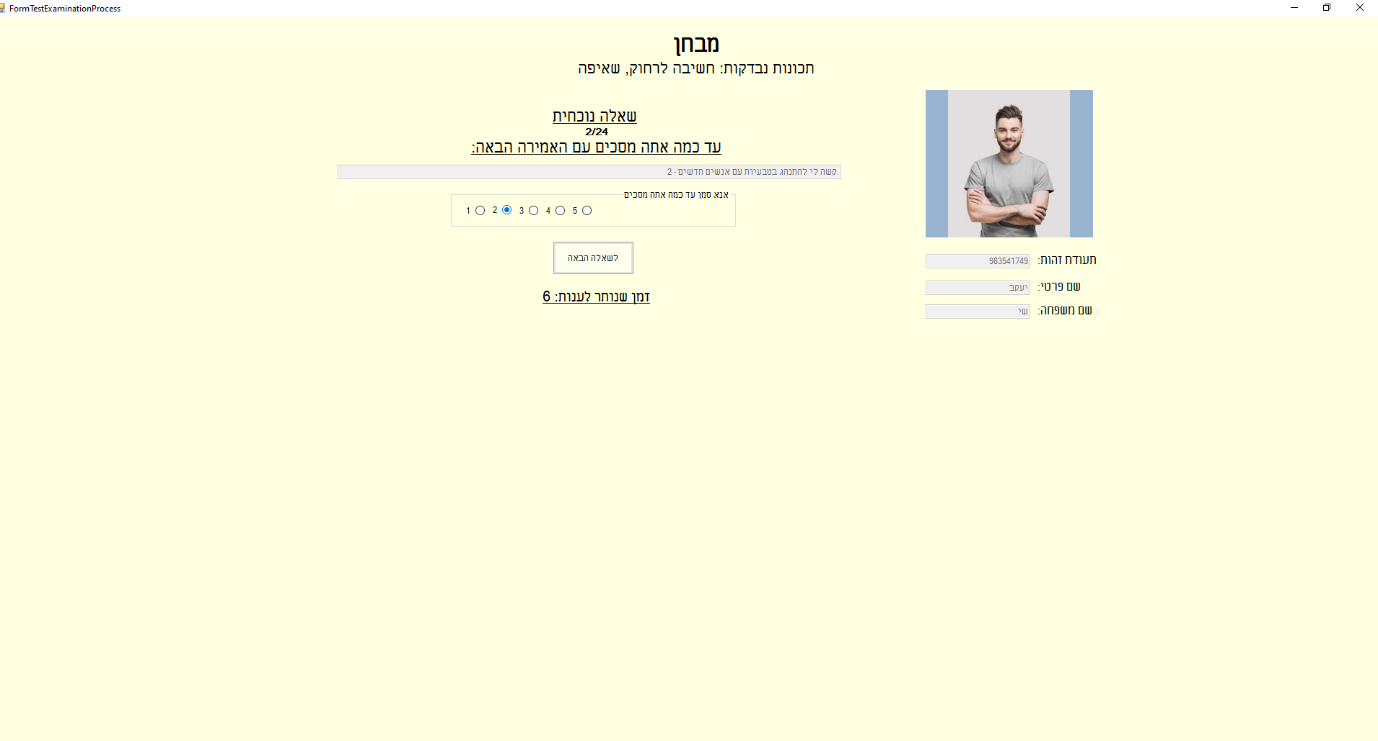
בלחיצה בסרגל התפריטים על "בחנים" ואז לחיצה על "עשיית מבחן" יפתח תהליך התחלת המבחן (עמוד 52), לאחר מילוי הפרטים ולחיצה על "התחלת מבחן" המסך יעבור אל תהליך ביצוע המבחן, בתהליך ביצוע המבחן יתבצע המבחן לפי טופס המבחן הנבחר, פרטי מבצע המבחן יופיעו בצד הימיני של המסך עם תמונה שלו, והשאלה הנוכחית תופיע בתיבת טקסט מתחת לכתוביות- "שאלה נוכחית" ומתחת למספר השאלות שבוצעו עד כה וכמה שאלות נותרו, בשביל לענות על השאלה המשתמש צריך לבחור עד כמה הוא מסכים עם האמירה שמופיעה בין המספרים הנמצאים ב"גרופ-בוקס" עם הכיתוב- "אנא סמן עד כמה אתה מסכים", לאחר מילוי התשובה צריך המשתמש ללחוץ על כפתור "לשאלה הבאה" בשביל שמירת התשובה והמשך לשאלה הבאה, המשתמש צריך לענות לפני שיגמר הזמן הנמצא בכיתוב מתחת לכפתור וללחוץ על הכפתור לשמירת התשובה, אחרת התשובה תיחשב כפסולה ותוריד נקודות.

לאחר מילוי כל שאלות הבחינה, המסך ישתנה ויופיע הכיתוב: "יפה מאוד! סיימת את המבחן!", מתחתיו הכיתוב- "ציונך הוא:" ומתחתיו את הציון שקיבל המשתמש, במצב זה נשמרו כל התשובות שהמשתמש הכניס ולפי התשובות שלו והתאמתו לפיהם לתכונות שהטופס בחינה בודק יחושב הציון, במצב זה ניתן לאחר צפייה בציון לסגור את תהליך ביצוע המבחן.

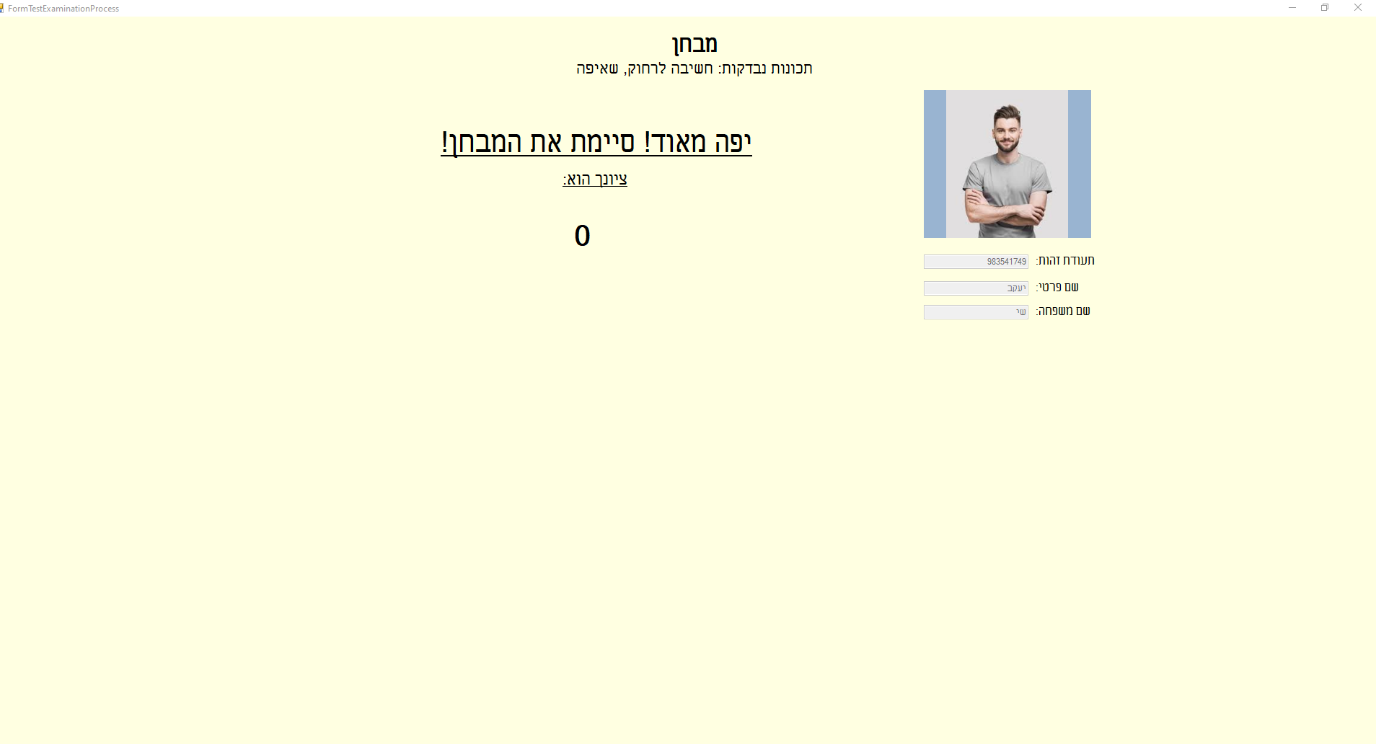
מסך ביצוע המבחן:

****

לאחר בחירת תשובה במבחן התשובה הנבחרת תסומן בעיגול שחור, כמו בתמונה:



לאחר ביצוע כל שאלות המבחן וסיום המבחן יופיע המסך הבא עם הציון:



**הקוד במערכת**

ישנם הרבה מחלקות קוד במערכת, ישנם מחלקות הדומות מאוד אחת לשנייה כמו במחלקות טפסי הטבלאות או ההוספה, וישנם גם מחלקות הייחודיות הנמצאות בטפסים כמו טופס התפריט או ההתחברות שלהם קודים ייחודים במערכת.

פירוט המחלקות יתוארו כאן לפי חילוק לטפסים שהקוד בהם חוזר על עצמם, "טפסים דומים" וטפסים הייחודים אשר יקראו- "טפסים ייחודים", להלן פירוט הקוד לפי חילוק זה:

**טפסים דומים**

**טפסי הטבלאות:**

טפסי הטבלאות שתהליכיהם תוארו בעמודים 30-33 מכילים מספר טפסים:

* FormTblAnswersInTest
* FormTblCharacters
* FormTblCharsInForm
* FormTblCities
* FormTblForms
* FormTblQuestions
* FormTblQuestionsForChar
* FormTblQuestionsInForm
* FormTblTests
* FormTblUsers

הפעולות המשותפות הקיימים לטפסים אלו:

- [FormName]\_Loadפקודה המתרחשת בפתיחת הטופס המעדכנת את הטבלה הנוכחית במידע ממסד הנתונים.

SaveButtonClick – פקודה המופעלת על ידי כפתור שמירה הנמצא בטופס, השומרת את השינויים שהמשתמש עשה לטבלה אל מסד הנתונים, במקרה שהשמירה בוצעה בהצלחה תוצג הודעה מתאימה, ובמקרה של שגיאה תופיע הודעת שגיאה עם נתונים על השגיאה.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

**טפסי ההוספה:**

טפסי ההוספה שתהליכיהם תוארו בעמודים 34-37 מכילים מספר טפסים:

* FormAddAnswersInTest
* FormAddCharacters
* FormAddCharsInForm
* FormAddCities
* FormAddForms
* FormAddQuestions
* FormAddQuestionsForChar
* FormAddQuestionsInForm
* FormAddTests
* FormAddUsers

הפעולות המשותפות הקיימים לטפסים אלו:

RefreshDataGridView – פעולה הקוראת לאחר פקודת ההוספה, המעדכנת את הטבלה של הטופס הנבחר בנתונים החדשים שנוספו לטבלה.

AddButton - פקודה המופעלת על ידי כפתור ההוספה הנמצא בטופס, המוסיפה עמודה חדשה בטבלה הנבחרת לפי הנתונים שהמשתמש מילא בטופס, במקרה שההוספה בוצעה בהצלחה תוצג הודעה מתאימה, ובמקרה של שגיאה תופיע הודעת שגיאה עם נתונים על השגיאה.

[FormName]\_Load- פקודה המתרחשת בפתיחת הטופס המעדכנת את הטבלה הנוכחית במידע ממסד הנתונים.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

פעולות יחודיות לטופס FormAddAnswersInTest :

FillComboTests – פקודה הממלאת את קומבו בוקס המבחנים במבחנים הקיימים במערכת.

GetFormName – פקודה המחזירה את שם הטופס של מספר הטופס אשר מקבלת.

FillComboAnswers – פקודה הממלאת את קומבו בוקס התשובות לפי מספר התשובות האפשריות לשאלה הנבחרת.

FillComboQuestions – פקודה הממלאת את קומבו בוקס השאלות בכל השאלות הקיימות לטופס המבחן של המבחן הנבחר.

GetQuestionOrderNum – פקודה המחזירה את כל השאלות הקיימות בטופס המבחן אשר הפקודה מקבלת את מספרו.

GetQuestionText – פקודה המחזירה את טקסט השאלה של מספר השאלה אשר הפקודה מקבלת.

פעולות יחודיות לטופס FormAddCharsInForm :

FillComboForms – פקודה הממלאת את קומבו בוקס הטפסים בכל הטפסים הקיימים.

FillComboChars – פקודה הממלאת את קומבו בוקס התכונות בכל התכונות הקימות.

פעולות יחודיות לטופס FormAddForms :

FillComboCreatorID – פקודה הממלאת את קומבו בוקס המשתמשים בכל המשתמשים הקיימים במערכת.

פעולות יחודיות לטופס FormAddQuestions :

FillComboAnswers – פקודה הממלאת את קומבו בוקס התשובות במספרים בין 2 ל10.

פעולות יחודיות לטופס FormAddQuestionsForChar :

FillComboChars – פקודה הממלאת את קומבו בוקס התכונות בכל התכונות הקימות במערכת.

FillComboQuestions – פקודה הממלאת את קומבו בוקס השאלות בכל השאלות הקימות במערכת.

FillComboFromValue – פקודה הממלאת את קומבו בוקס ערכי ההתחלה במספרים לפי כמות התשובות האפשריות לשאלה הנבחרת בקומבו בוקס השאלות.

FillComboToValue – פקודה הממלאת את קומבו בוקס ערכי הסיום במספרים בין ערך ההתחלה הנבחר בקומבו בוקס ערכי ההתחלה לבין כמות התשובות האפשריות לשאלה הנבחרת בקומבו בוקס השאלות.

FromValueClick & ToValueClick & QuestionsClick – פקודות האחריות לכך שהמשתמש ימלא את הנתונים לפי הסדר הנחוץ בשביל נתונים נכונים, בחירת שאלה – בחירת ערך התחלה – ולבסוף ערך סיום, במקרה ולא יפעל ככה המשתמש הנתונים שבחר יתאפסו והמשתמש יצטרך לבחור לפי הסדר.

פעולות יחודיות לטופס FormAddQuestionsInForm :

FillComboForms – פקודה הממלאת את קומבו בוקס טפסי המבחן בכל טפסי המבחן הקיימים במערכת.

FillComboQuestions – פקודה הממלאת את קומבו בוקס השאלות בכל השאלות הקיימות במערכת.

FormIDClosed – פקודה הקוראת לאחר בחירת טופס בחינה, וממלאה את המספר הסידורי של השאלה החדשה במספר החדש לפי מספר השאלה האחרון הנמצא בטופס בחינה הנבחר באותו הזמן.

פעולות יחודיות לטופס FormAddTests :

FillComboUsers – פקודה הממלאת את קומבו בוקס המשתמשים בכל המשתמשים הקיימים במערכת.

FillComboForms – פקודה הממלאת את קומבו בוקס טפסי המבחן בכל טפסי המבחן הקיימים במערכת.

פעולות יחודיות לטופס FormAddUsers :

FillCityCombo – פקודה אשר ממלאה את קומבו בוקס הערים בכל הערים הקיימות במערכת.

browseClick – פקודה הקוראת כאשר נלחץ כפתור העיון, ומאפשרת בחירת תמונה מהמחשב ושומרת את המיקום שלו בשדה התמונה.

IsIdValid – פקודה אשר בודקת האם תעודת הזהות אשר הקליד המשתמש חוקית.

**טפסי העדכון:**

טפסי העדכון שתהליכיהם תוארו בעמודים 38-40 מכילים מספר טפסים:

* FormUpdateAnswersInTest
* FormUpdateCharacters
* FormUpdateForms
* FormUpdateQuestions
* FormUpdateQuestionsForChar
* FormUpdateQuestionsInForm
* FormUpdateTests
* FormUpdateUsers

הפעולות המשותפות הקיימים לטפסים אלו:

firstButtonClick – פעולה המתרחשת בלחיצה על כפתור "ראשון" הפעולה בוחרת בשורה הראשונה בטבלה וממלאה את הנתונים של הטופס לפי שורה זו בעזרת הפעלת הפקודה- "FillSelectedRow"

prevButtonClick – פעולה המתרחשת בלחיצה על כפתור "הקודם" הפעולה בוחרת בשורה הקודמת בטבלה לשורה הנבחרת כרגע וממלאה את הנתונים של הטופס לפי שורה זו בעזרת הפעלת הפקודה- "FillSelectedRow"

nextButtonClick – פעולה המתרחשת בלחיצה על כפתור "הבא" הפעולה בוחרת בשורה הבאה בטבלה לשורה הנבחרת כרגע וממלאה את הנתונים של הטופס לפי שורה זו בעזרת הפעלת הפקודה- "FillSelectedRow"

lastButtonClick – פעולה המתרחשת בלחיצה על כפתור "אחרון" הפעולה בוחרת בשורה האחרונה בטבלה וממלאה את הנתונים של הטופס לפי שורה זו בעזרת הפעלת הפקודה- "FillSelectedRow"

FillSelectedRow – פקודה הממלאה את השדות בטופס לפי האינדקס שורה הנבחר של הטבלה בטופס, ולבסוף מפעיל את הפקודה "EnableButtons"

EnableButtons – פקודה המפעילה את כפתורי הניווט של הטופס לפי אינדקס השורה הנבחר כרגע, במקרה שהשורה הנבחרת היא הראשונה כפתור "הקודם" ינעל, במקרה שהשורה הנבחרת היא האחרונה כפתור "הבא" ינעל.

ButtonUpdateClick – פקודה המעדכנת את השורה הנבחרת בטופס לפי האינדקס הנבחר לפי שינוי הנתונים שהמשתמש עשה, במקרה שהעדכון בוצעה בהצלחה תוצג הודעה מתאימה, ובמקרה של שגיאה תופיע הודעת שגיאה עם נתונים על השגיאה.

RefreshDataGridView – פעולה הקוראת לאחר פקודת העדכון, המעדכנת את הטבלה של הטופס הנבחר בנתונים החדשים שנוספו לטבלה.

CellClick – פקודה המופעלת בלחיצה על תא בטבלה, הפקודה מעדכנת את האינדקס שורה הנבחר אל השורה שעליה המשתמש לחץ ומפעילה את הפקודה FillSelectedRow"" המעדכנת את שדות הטופס אל השורה הנבחרת.

[FormName]\_Load - פקודה המתרחשת בפתיחת הטופס המעדכנת את הטבלה הנוכחית במידע ממסד הנתונים.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

פעולות יחודיות לטופס FormUpdateAnswersInTest :

FillComboAnswers – פקודה הממלאת את קומבו בוקס התשובות בכל מספרי התשובות האפשריות לשאלה הנבחרת.

GetQuestionText – פקודה המחזירה את טקסט השאלה של מספר השאלה אשר מקבלת.

GetFormName – פקודה המחזירה את שם טופס המבחן של מספר הטופס אשר מקבלת.

QuestionText – פקודה המחזירה המספר הסידורי של השאלה אשר מקבלת וגם את טקסט השאלה שלה.

GetQuestionID – פקודה המחזירה את טקסט השאלה לפי מספר טופס המבחן ולפי המספר הסידורי של השאלה.

פעולות יחודיות לטופס FormUpdateForms :

FillComboCreatorID – פקודה הממלאת את קומבו בוקס המשתמשים בכל המשתמשים הקיימים במערכת.

GetCreatorName – פקודה המחזירה את פרטי המשתמש לפי תעודת הזהות שלו אשר מקבלת.

פעולות יחודיות לטופס FormUpdateQuestions :

FillComboAnswers – פקודה הממלאת את קומבו בוקס התשובות במספרים בין 2 ל10.

פעולות יחודיות לטופס FormUpdateQuestionsForChar :

FillComboFromValue – פקודה הממלאת את קומבו בוקס ערכי ההתחלה לפי התשובות האפשריות לשאלה הנבחרת.

FillComboToValue – פקודה הממלאת את קומבו בוקס ערכי הסיום במספרים בין ערך ההתחלה הנבחר לבין מספר התשובות האפשריות לשאלה הנבחרת.

FromValueClick & ToValueClick – פקודות האחריות לכך שהמשתמש ימלא את נתוני ערכי ההתחלה והסיום לפי הסדר, ובמקרה אשר מנסה לבחור את ערך הסיום לפני ערך ההתחלה הפקודות מאפסות את הקומבו בוקס ובכך נותנות לכך שהמשתמש יוכל למלא את הנתונים רק לפי הסדר.

GetQuestionText – פקודה המחזירה את מספר השאלה ואת טקסט השאלה לפי מספר השאלה אשר מקבלת.

פעולות יחודיות לטופס FormUpdateQuestionsInForm :

FillComboQuestions – פקודה הממלאת את קומבו בוקס השאלות בכל השאלות הקיימות במערכת.

SearchForQuestion – פקודה אשר מחזירה את מספר השאלה וטקסט השאלה לפי מספר השאלה אשר מקבלת.

GetFormName – פקודה המחזירה את מספר טופס המבחן ואת טקסט טופס המבחן לפי מספר טופס המבחן אשר מקבלת.

פעולות יחודיות לטופס FormUpdateTests :

FillComboUsers – פקודה הממלאת את קומבו בוקס המשתמשים בכל המשתמשים אשר קיימים במערכת.

GetUsersName – פקודה המחזירה את ת"ז המשתמש ואת שמו לפי מספר הת"ז אשר מקבלת.

FillComboForms – פקודה הממלאת את קומבו בוקס טפסי המבחן בכל טפסי המבחן אשר קיימים במערכת.

GetFormName – פקודה המחזירה שם טופס מבחן ואת מספרו לפי מספר טופס המבחן אשר מקבלת.

פעולות יחודיות לטופס FormUpdateUsers :

FillCityCombo – פקודה הממלאת את קומבו בוקס הערים בכל הערים הקיימים במערכת.

browseClick – פקודה אשר פועלת בלחיצה על כפתור העיון ומאפשרת למשתמש בחירת תמונה ממחשבו והיא שומרת את המיקום של התמונה בשדה התמונה.

**טפסי הדוחות:**

טפסי הדוחות שתהליכיהם תוארו בעמודים 41-42 מכילים מספר טפסים:

* FormRptTest
* FormRptTestForm
* FormRptQuestionsByCharacter

הפעולות המשותפות הקיימים לטפסים אלו:

ClearReportButton – פעולה המתרחשת בלחיצה על כפתור "ניקוי דוח" ומנקה את הדוח והופכת אותו לריק.

SelectColorButton – פעולה המתרחשת בלחיצה על כפתור "בחר צבע" ונותנת למשתמש לשנות את צבע פרטי הדוח.

EditListView – פעולה הממלאת את הליסט ויוו בפרטי הדוח הרצוי.

ShowReportButton – פעולה המתרחשת בלחיצה על כפתור "הצגת דוח" המשיגה מטבלאות ממסד הנתונים את המידע הרצוי לדוח ומפעילה את פקודת ה"EditListView" הממלאה את הליסט ויוו בפרטים אלו.

[FormName]\_Load - פקודה המתרחשת בפתיחת הטופס המעדכנת את הטבלה הנוכחית במידע ממסד הנתונים.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

פעולות יחודיות לטופס FormRptTest :

TestChars – פקודה המחזירה רשימה של כל התכונות שטופס הבחינה הנבחר בודק.

GetUserInfo – פקודה המחזירה מערך המכיל פרטי משתמש- שם פרטי ושם משפחה, על פי תעודת הזהות המתקבלת לפקודה.

FormInfo – פקודה הממלאה את נתוני המחלקה הדרושים לדוח על ידי קריאה לשאר פקודות המערכת וקריאה מטבלת השאלות בטופס, לפי הטופס הנבחר, ולבסוף מחשבת את ציון הבחינה ומציגה את הציון, וקוראת לפקודת הEditListView.

GetQuesData – פקודה המחזירה את טקסט השאלה ואת מספר התשובות האפשריות לשאלה לפי מספר השאלה המתקבלת בפקודה.

CheckForNotFake – פקודה הבודקת האם התכונה המתקבלת נבדקת על ידי טופס הבחינה המתקבל בפקודה, אם כן נבדקת יוחזר true במקרה ולא יוחזר false.

GetQustForCharInfo – פקודה המחזירה את ערך ההתחלה וערך הסיום של התכונה הנבדקת ובמקרה והתשובה המתקבלת לפקודה נמצאת בטווח הפקודה מוסיפה עוד לאוביקט ה"score" המשמש לחישוב הציון.

GetAnswerInTest – פקודה המחזירה את התשובה לשאלה המתקבלת לפקודה במבחן הנבחר.

פעולות יחודיות לטופס FormRptTestForm :

GetFormChars – פקודה המחזירה את כל התכונות הנבדקות בטופס הבחינה המתקבלת בפקודה במחרוזת.

GetQuestion – פקודה הממלאה את נתוני המחלקה הדרושים לדוח על ידי קריאה לשאר פקודות המערכת וקריאה מטבלת השאלות בטופס, לפי הטופס הנבחר, וקוראת לפקודת הEditListView.

GetQuesText – פקודה המחזירה את טקסט השאלה של מספר השאלה המתקבל בפקודה.

GetQuesChar – פקודה המחזירה את התכונה הנבדקת על ידי השאלה שמספרה מתקבל על ידי הפקודה, וגם את ערך ההתחלה וערך הסיום של ההתאמה לתכונה זו.

פעולות יחודיות לטופס FormRptQuestionsByCharacter :

GetQuesTextAndAnswers – פקודה המחזירה את טקסט השאלה ואת מספר האופציות לתשובה לשאלה לפי מספר השאלה המתקבל בפקודה.

**טפסי התרשימים:**

טפסי התרשימים שתהליכיהם תוארו בעמודים 43-44 מכילים מספר טפסים:

* FormChartQuestionsByPropertiy
* FormChartQuestionsByTest
* FormChartQuestionsInTestForm

הפעולות המשותפות הקיימים לטפסים אלו:

FillArrs – פעולה הממלאה את מערכי המערכת במידע הרצוי מהטבלאות הרצויות.

CountNumberOfQuestions – פעולה המחזירה את מספר השאלות בטופס הנבחר.

EditDataGridView – פעולה הממלאה את הטבלה במידע לפי הנתונים מהמערכים שהתמלאו.

ShowChart – פעולה המציגה את הדיאגרמה לפי הנתונים מהמערכים שהתמלאו.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

**טפסי החיפוש:**

טפסי החיפוש שתהליכיהם תוארו בעמודים 45-48 מכילים מספר טפסים:

* FormSearchAnswersInTest
* FormSearchCharacters
* FormSearchCharsInForm
* FormSearchCities
* FormSearchForms
* FormSearchQuestions
* FormSearchQuestionsForChar
* FormSearchQuestionsInForm
* FormSearchTests
* FormSearchUsers

הפעולות המשותפות הקיימים לטפסים אלו:

buttonRefresh\_Click – פקודה המופעלת בעזרת כפתור ה"רענן" המציגה בטבלה את כל הנתונים של הטבלה ללא פילטר.

buttonSearch\_Click – פקודה המופעלת בעזרת כפתור ה"חפש" המציגה בטבלה רק את הנתונים המכילים את המחרוזת הכתובה בתיבת הטקסט בטופס.

[FormName]\_Load - פקודה המתרחשת בפתיחת הטופס המעדכנת את הטבלה הנוכחית במידע ממסד הנתונים.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

**טפסים ייחודים**

**טופס ההתחברות:**

טופס ההתחברות למערכת הוא הטופס שנפתח ישירות בפתיחת המערכת שמימנו ניתן להיכנס למערכת, תהליכו תואר בעמודים 26-27, הפקודות שקימות בו הם:

OpenDb – פקודה המאתחלת ופותחת את החיבור למסד הנתונים שבה נעשה שימוש כמעט בכל טפסי המערכת. הפעולה הזו מופעלת פעם ראשונה בלבד בפתיחת הטופס והיא מופעלת פעם אחת בלבד ומעבירה את החיבור אל שאר פעולות המערכת.

CheckButtonClick – פעולה המתרחשת בלחיצה על הכפתור "כניסה" או בשינוי השם שלו- "המשך", במקרה שהנתונים שהוקלדו אינם קיימים במערכת או תעודת הזהות אינה תקינה תוצג הודעת שגיאה בהתאם, במקרה שהנתונים שהוקלדו נכונים תיבות הטקסט ינעלו ותמונת המשתמש תופיע ויוצג הודעה של ברוך הבא עם שם המשתמש, ושם הכפתור ישונה ל"המשך", במקרה של לחיצה נוספת על הכפתור עם השם "המשך" תופעל הפקודה שוב והפעם תפתח את טופס התפריט הראשי.

IsIdValid – פקודה הבודקת את המחרוזת המתקבלת אליה ומחזירה true במקרה שהיא תעודת זהות תקינה, וfalse במקרה שלא.

frMenu\_Disposed – פקודה המתרחשת במקרה שהטופס הנפתח – טופס התפריט נסגר, ובמקרה זה מאפסת את הנתונים הכתובים בתיבות הטקסט וחושפת את טופס ההתחברות.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

**טופס התפריט הראשי:**

טופס התפריט הראשי הוא הטופס שנפתח ישירות לאחר ההתחברות למערכת, מימנו אפשר להגיע אל כל שאר תהליכי המערכת, תהליכו תואר בעמודים 28-29, הפקודות שקימות בו הם:

Form\_Disposed – מופעלת על ידי פקודות פתיחת הטפסים מתי שנסגר טופס שנפתח, וחושפת את טופס התפריט הראשי.

[FormName]Click – מופעלת על ידי לחיצה על התהליך הרצוי בשורת התפריטים במעלה המסך, הפקודה הזו קיימת לכל אחד מהתהליכים הנפתחים על ידי שורת התפריטים, כל פקודה פותחת את הטופס הרצוי ובמקרה של סגירת הטופס מחזירה את טופס התפריט הראשי על ידי הפקודה "Form\_Disposed".

**טופס יצירת מבחן**

טופס יצירת המבחן הוא הטופס שבו ניתן ליצור טופס מבחן חדש למערכת בצורה פשוטה וקלה ועם תהליך אוטמטי היוצר את המבחן באופן אקראי ולפי הקריטריונים הנבחרים לפי היוצר, תהליכו תואר בעמודים 49-51, הפקודות שקימות בו הם:

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

FillQuesCount – פקודה זו מופעלת מבנאי הטופס ומוסיפה לקומבו בוקס של בחירת כמות שאלות לכל תכונה מספרים מ10 עד 20 (כולל)

FillCharacters – פקודה זו מופעלת מבנאי הטופס ומוסיפה לצאק ליסט בוקס של התכונות את כל התכונות הקימות במערכת.

buttonCreateClick – פקודה זו מופעלת מלחיצה על הכפתור "צור טופס בחינה" פקודה זו בעזרת פקודות אחרות אפשר היא קוראת להם יוצרת את טופס הבחינה לפי הקריטריונים של המשתמש, פקודות אשר היא משתמשת בהם:

QuestionsForCharNumber – מחזיר את מספר השאלות הקימות לתכונה שאותה מקבלת הפקודה.

AddForm – פקודה זו יוצרת בחינה חדשה במערכת אצל ממסד הנתונים לפי הקריטריונים של המשתמש.

FormNumber – מחזיר את מספר הטופס האחרון (הכי גדול).

AddCharsInForm – פקודה זו מוסיפה אל טבלת התכונות לטפסים את התכונות שאותם הטופס החדש בודק ומתאימה אותם לטופס החדש.

AddRandomQuestions – פקודה זו מביאה באופן רנדומלי כמות שאלות לפי התכונות שאותם הטופס בודק לפי הכמות הדרושה, בהוספת 10 אחוז שאלות מזויפות לאמינות הבחינה, לאחר מכן מוסיפה הפקודה את השאלות לטבלה שאלות לטופס בחינה עם התאמה למספר הטופס שכעת נוצר.

charChecker - פקודה זו בודקת לפי מספר השאלה אשר מקבלת האם השאלה מתאימה לתכונה אותה מקבלת והאם השאלה הזו היא בודקת את התכונה הנדרשת

**טופס התחלת מבחן:**

טופס התחלת המבחן הוא הטופס שמימנו ניתן להתחיל ביצוע מבחן, בו המשתמש בוחר את עצמו ואת טופס המבחן אותו ירצה לבצע ואת המהירות שבה ירצה לבצע את המבחן, הטופס יוצר מבחן חדש במערכת ומעביר את המשתמש אל טופס תהליך המבחן בשביל ביצועו, תהליכו תואר בעמוד 52, הפקודות שקימות בו הם:

FormTestExamination\_Load - פקודה המתרחשת בפתיחת הטופס המעדכנת את הטבלה הנוכחית במידע ממסד הנתונים.

OnSizeChanged – פקודה המופעלת כשהמסך מתרחב לגודל מקסימלי, מה שקורה בבנאי של הטפסים, כשפעולה זו מופעלת היא גורמת לכל האובייקטים בטופס להיות במרכז הטופס המוגדל.

Form\_Disposed – מופעלת על ידי פקודת פתיחת הטופס מתי שנסגר טופס שנפתח, וחושפת את טופס התחלת המבחן.

CreateTestClick – פקודה זו מופעלת על ידי לחיצה על כפתור "התחל מבחן" פקודה זו יוצרת מבחן חדש במערכת בטבלת המבחנים ושומרת אותו לפי הנתונים אשר הכניס המשתמש בטופס, וגם שומרת מידע על המבחן ופותחת את טופס ביצוע המבחן ומעבירה אליו את פרטי המבחן המבוצע, כמו פרטי משתמש, פרטי טופס המבחן ופרטי המבחן שנוצר כעת.

GetUserData – מחזירה מידע על המשתמש לפי הת"ז אשר מקבלת.

CurrentCreatedTest – פקודה זו מחזירה את מספר המבחן הגדול ביותר אשר אומר שהיא מחזירה את מספר המבחן אשר כעת נוצר.

UserChosed & FormChosed – פקודות אלו מופעלות על ידי לחיצה על טבלאות המשתמשים והטפסים, וגורמת לכך שחייב לבחור גם משתמש וגם טופס על מנת שכפתור התחלת המבחן יהיה פתוח וניתן יהיה להתחיל את המבחן.

**טופס ביצוע מבחן:**

טופס ביצוע המבחן הוא הטופס שמימנו המשתמש מבצע את המבחן, ניתן להגיע אליו מטופס התחלת המבחן מידע על המשתמש מופיע בצד ימין של המסך ובמרכז שאלות לפי טופס המבחן הנבחר יופיעו לפי המהירות שנבחרה והמשתמש צריך לענות עד כמה הוא מסכים עם האמירה המופיעה לפי המספרים המוצגים במסך, לאחר ביצוע כל שאלות הבחינה המידע ישמר וציון המבחן יחושב ויוצג למשתמש, תהליכו תואר בעמודים 53-54, הפקודות שקימות בו הם:

TestChars – פקודה זו מחזירה את כל התכונות אשר נבדקות במבחן המבוצע כעת לפי טופס המבחן.

Test – הפקודה הראשית אשר מפעילה את המבחן, אשר מציגה את כמות השאלות שנותרת, את השאלה העכשווית, ומפעילה את הפקודה אשר מוסיפה את כפתורי האופציות.

StartTimer – פקודה אשר מפעילה את הטיימר של המבחן.

timer1\_Tick – פקודה הפועלת כל שניה של הטיימר, מעדכנת את הטקסט אשר מראה את כמות הזמן הנותרת לענות על השאלה, ובמקרה והזמן מגיע ל0 מחשיבה את שאלה זו לנכשלת וכותבת בתור תוצאה "-1" אשר נחשב לתשובה שאינה נכונה עקב גמר הזמן, ובמקרה ושאלה זו היא האחרונה גם יכולה לסיים את המבחן.

GetAnswerInTest – פקודה זו מחזירה את התשובה אשר המשתמש ענה על השאלה אשר היא מקבלת כקלט לפי המבחן אשר היא מקבלת כקלט.

GetScore – פקודה אשר מחשבת את הציון במבחן ומחזירה את הציון.

GetQustForCharInfo – פקודת עזר של פקודת חישוב הציון אשר בודקת את התשובה שהיא מקבלת כקלט לפי טווח ההתאמה לתכונה, במקרה והתשובה מתאימה לטווח ההתאמה הניקוד יוגדל, אחרת לא, ובמקרה של שאלה מזויפת יתווסף עוד ניקוד למד כמות השאלות המזויפות.

CheckForNotFake – בודק האם השאלה הזו מזויפת ואינה שיכת לתכונות אשר אותם בודק המבחן.

ClearButtons – פקודה אשר מסתירה ומעיפה את הצאק מכל כפתורי הרדיו.

AddButtons – פקודה החושפת את כפתורי הרדיו לפי כמות התשובות האפשרית בשאלה.

NumberOfQuestions – מחזירה את מספר השאלות אשר בטופס בחינה.

getQuesText – מחזיר את מספר השאלה ואת טקסט השאלה של מספר השאלה אשר קיבל.

nextQuestionClick – פקודה המופעלת במקרה שמשתמש לחץ על כפתור "לשאלה הבאה", ומפעילה את פקודת תהליך השאלה הבאה.

nextQuesProcess – פקודה אשר אחראית על לעבור לשאלה הבאה, שומרת את התשובה בעזרת פקודה נוספת, במקרה וזו השאלה האחרונה היא מסיימת את המבחן, ואם לא היא מפעילה את פקודת ה"Test" אשר מפעילה את המבחן ועוברת לשאלה הבאה.

SaveAnswer – שומרת את תשובתו של המשתמש על השאלה.

**קוד המערכת**

**קוד של טופס הוספת תשובות במבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddAnswersInTest : Form

{

private OleDbConnection dataConnection;

public FormAddAnswersInTest(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboTests();

}

private void FormAddAnswersInTest\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetAnswersInTest.tblAnswersInTest' table. You can move, or remove it, as needed.

this.tblAnswersInTestTableAdapter.Fill(this.dataSetAnswersInTest.tblAnswersInTest);

}

private void FillComboTests()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT testID, testFormID " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + GetFormName(dataReader.GetInt32(1).ToString());

comboTests.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill tests combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetFormName(string testFormID)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(dataReader.GetInt32(0).ToString().Equals(testFormID))

{

return dataReader.GetString(1);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("get form name failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private void FillComboAnswers()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesText, quesAnswers " +

"FROM tblQuestions " +

"ORDER BY quesText";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string[] arr = comboOrderNum.Text.Split('-');

while (dataReader.Read())

{

if (dataReader.GetString(0).Equals(arr[1].Substring(1)))

{

for (int i = 1; i <= dataReader.GetInt32(1); i++)

{

comboAnswers.Items.Add(i);

}

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill answers combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboQuestions()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT testID, testFormID " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string[] arr = comboTests.Text.Split(' ');

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(arr[0]))

{

string[] arr2 = GetQuestionOrderNum(dataReader.GetInt32(1).ToString());

for (int i = 0; i < arr2.Length; i++)

{

comboOrderNum.Items.Add(arr2[i]);

}

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string[] GetQuestionOrderNum(string testForm)

{

string[] error = new string[0];

try

{

string questionText;

int count = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifFormID, qifOrderNum, qifQuestionID " +

"FROM tblQuestionsInForm " +

"ORDER BY qifOrderNum";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(testForm))

{

count++;

}

}

dataReader.Close();

string[] arr = new string[count];

count = 0;

dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(testForm))

{

questionText = GetQuestionText(dataReader.GetInt32(2).ToString());

arr[count] = dataReader.GetInt32(1).ToString() + " - " + questionText;

count++;

}

}

dataReader.Close();

return arr;

}

catch (Exception err)

{

MessageBox.Show("Get order num failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return error;

}

private string GetQuestionText(string questionID)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(questionID))

{

str = dataReader.GetString(1);

return str;

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Get QuestionText failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void AddButton(object sender, EventArgs e)

{

try

{

string[] arr = comboTests.Text.Split(' ');

string[] arr2 = comboOrderNum.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT aitTestID, aitOrderNum " +

"FROM tblAnswersInTest " +

"ORDER BY aitTestID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(arr[0]) && dataReader.GetInt32(1).ToString().Equals(arr2[0]))

{

MessageBox.Show("This question already have an answer.");

return;

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblAnswersInTest failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string[] arr = comboOrderNum.Text.Split(' ');

string[] arr2 = comboTests.Text.Split(' ');

string str = string.Format

("INSERT INTO tblAnswersInTest " +

"(aitTestID, aitOrderNum, aitAnswer) " +

" VALUES ( {0}, {1}, {2})",

arr2[0], arr[0], comboAnswers.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblAnswersInTest ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblAnswersInTest failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblAnswersInTest " +

"ORDER BY aitTestID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void QuestionsClick(object sender, EventArgs e)

{

comboAnswers.Text = "";

comboOrderNum.Items.Clear();

if (comboTests.Text != "")

{

FillComboQuestions();

}

}

private void AnswersClick(object sender, EventArgs e)

{

comboAnswers.Items.Clear();

if (comboOrderNum.Text != "")

{

FillComboAnswers();

}

}

private void TestsClick(object sender, EventArgs e)

{

comboAnswers.Text = "";

comboOrderNum.Text = "";

}

}

}

**קוד של טופס הוספת תכונות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddCharacters : Form

{

private OleDbConnection dataConnection;

public FormAddCharacters(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

}

private void FormAddCharacters\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharacters.tblCharacters' table. You can move, or remove it, as needed.

this.tblCharactersTableAdapter.Fill(this.dataSetCharacters.tblCharacters);

}

private void AddButton(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblCharacters " +

"(charName, charDesc) " +

" VALUES ( \"{0}\", \"{1}\")",

charName.Text, charDesc.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblCharacters ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblCharacters failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

}

}

**קוד של טופס הוספת תכונות בטופס:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddCharsInForm : Form

{

private OleDbConnection dataConnection;

public FormAddCharsInForm(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboChars();

FillComboForms();

}

private void FormAddCharsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharsInForm.tblCharsInForm' table. You can move, or remove it, as needed.

this.tblCharsInFormTableAdapter.Fill(this.dataSetCharsInForm.tblCharsInForm);

}

private void FillComboForms()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

comboFormID.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill forms combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillComboChars()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT charName " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetString(0);

comboChars.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill chars combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void AddButton(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string[] arr = comboFormID.Text.Split(' ');

string str = string.Format

("INSERT INTO tblCharsInForm " +

"(cifFormID, cifCharName) " +

" VALUES ( {0}, \"{1}\")",

int.Parse(arr[0]), comboChars.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblCharsInForm ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblCharsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharsInForm " +

"ORDER BY cifFormID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס הוספת ערים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddCities : Form

{

private OleDbConnection dataConnection;

public FormAddCities(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormAddCities\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCities.tblCities' table. You can move, or remove it, as needed.

this.tblCitiesTableAdapter.Fill(this.dataSetCities.tblCities);

}

private void AddButton(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblCities " +

"(cityName) " +

" VALUES ( \"{0}\")",

cityName.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblCities ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblCities failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCities " +

"ORDER BY cityName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס הוספת טפסים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddForms : Form

{

private OleDbConnection dataConnection;

public FormAddForms(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboCreatorID();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillComboCreatorID()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userFirstName, userLastName " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1) + " " + dataReader.GetString(2);

comboUsers.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill users combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormAddForms\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetForms.tblForms' table. You can move, or remove it, as needed.

this.tblFormsTableAdapter.Fill(this.dataSetForms.tblForms);

}

private void AddButton(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string[] arr = comboUsers.Text.Split(' ');

string str = string.Format

("INSERT INTO tblForms " +

"(formName, formDate, formCreatorID) " +

" VALUES ( \"{0}\", \"{1}\", {2})",

formName.Text, FormDate.Text, arr[0]);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblForms ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblForms failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס הוספת שאלות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddQuestions : Form

{

private OleDbConnection dataConnection;

public FormAddQuestions(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboAnswers();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormAddQuestions\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestions.tblQuestions' table. You can move, or remove it, as needed.

this.tblQuestionsTableAdapter.Fill(this.dataSetQuestions.tblQuestions);

}

private void FillComboAnswers()

{

for (int i = 2; i <= 10; i++)

{

comboAnswers.Items.Add(i);

}

}

private void AddButton(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblQuestions " +

"(quesText, quesAnswers) " +

" VALUES ( \"{0}\", {1})",

quesText.Text, comboAnswers.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblQuestions ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblQuestions failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

} }

**קוד של טופס הוספת שאלות לתכונה:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddQuestionsForChar : Form

{

private OleDbConnection dataConnection;

public FormAddQuestionsForChar(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboQuestions();

FillComboChars();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillComboChars()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT charName " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetString(0);

comboChars.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill chars combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboQuestions()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

comboQuestions.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboFromValue()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesAnswers " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string[] arr = comboQuestions.Text.Split(' ');

while (dataReader.Read())

{

if(arr[0] == dataReader.GetInt32(0).ToString())

{

for (int i = 1; i <= dataReader.GetInt32(1); i++)

{

comboFromValue.Items.Add(i);

}

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill from value combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboToValue()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesAnswers " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string[] arr = comboQuestions.Text.Split(' ');

while (dataReader.Read())

{

if (arr[0] == dataReader.GetInt32(0).ToString())

{

for (int i = int.Parse(comboFromValue.Text); i <= dataReader.GetInt32(1); i++)

{

comboToValue.Items.Add(i);

}

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill to value combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormAddQuestionsForChar\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsForChar.tblQuestionsForChar' table. You can move, or remove it, as needed.

this.tblQuestionsForCharTableAdapter.Fill(this.dataSetQuestionsForChar.tblQuestionsForChar);

}

private void AddButton(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string[] arr = comboQuestions.Text.Split(' ');

string str = string.Format

("INSERT INTO tblQuestionsForChar " +

"(qfcCharName, qfcCharOrder, qfcFromValue, qfcToValue) " +

" VALUES ( \"{0}\", {1}, {2}, {3})",

comboChars.Text, arr[0], comboFromValue.Text, comboToValue.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblQuestionsForChar ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblQuestionsForChar failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsForChar " +

"ORDER BY qfcCharName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FromValueClick(object sender, EventArgs e)

{

comboToValue.Text = "";

comboFromValue.Items.Clear();

if (comboQuestions.Text != "")

{

FillComboFromValue();

}

}

private void ToValueClick(object sender, EventArgs e)

{

comboToValue.Items.Clear();

if (comboFromValue.Text != "")

{

FillComboToValue();

}

}

private void QuestionsClick(object sender, EventArgs e)

{

comboToValue.Text = "";

comboFromValue.Text = "";

}

}

}

**קוד של טופס הוספת שאלות לטופס****:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddQuestionsInForm : Form

{

private OleDbConnection dataConnection;

public FormAddQuestionsInForm(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboQuestions();

FillComboForms();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormAddQuestionsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsInForm.tblQuestionsInForm' table. You can move, or remove it, as needed.

this.tblQuestionsInFormTableAdapter.Fill(this.dataSetQuestionsInForm.tblQuestionsInForm);

}

private void FillComboForms()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

comboFormID.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill forms combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboQuestions()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

comboQuestions.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void AddButton(object sender, EventArgs e)

{

try

{

string[] arr = comboFormID.Text.Split(' ');

string[] arr2 = comboQuestions.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifFormID, qifQuestionID " +

"FROM tblQuestionsInForm " +

"ORDER BY qifFormID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(arr[0]) && dataReader.GetInt32(1).ToString().Equals(arr2[0]))

{

MessageBox.Show("You cant enter the same question more then 1 to a form.");

return;

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string[] arr = comboQuestions.Text.Split(' ');

string[] arr2 = comboFormID.Text.Split(' ');

string str = string.Format

("INSERT INTO tblQuestionsInForm " +

"(qifFormID, qifOrderNum, qifQuestionID) " +

" VALUES ( {0}, {1}, {2})",

arr2[0], orderNum.Text, arr[0]);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblQuestionsInForm ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblQuestionsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifFormID, qifOrderNum " +

"FROM tblQuestionsInForm " +

"ORDER BY qifFormID";

string[] arr = comboFormID.SelectedItem.ToString().Split(' ');

int biggestOrderNum = 0;

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (arr[0].Equals(dataReader.GetInt32(0).ToString()))

{

if (biggestOrderNum < dataReader.GetInt32(1))

{

biggestOrderNum = dataReader.GetInt32(1);

}

}

}

orderNum.Text = (biggestOrderNum + 1).ToString();

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsInForm " +

"ORDER BY qifOrderNum";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormIDClosed(object sender, EventArgs e)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifFormID, qifOrderNum " +

"FROM tblQuestionsInForm " +

"ORDER BY qifFormID";

string[] arr = comboFormID.SelectedItem.ToString().Split(' ');

int biggestOrderNum = 0;

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(arr[0].Equals(dataReader.GetInt32(0).ToString()))

{

if(biggestOrderNum < dataReader.GetInt32(1))

{

biggestOrderNum = dataReader.GetInt32(1);

}

}

}

orderNum.Text = (biggestOrderNum + 1).ToString();

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס הוספת מבחנים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddTests : Form

{

private OleDbConnection dataConnection;

public FormAddTests(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillComboForms();

FillComboUsers();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormAddTests\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetTests.tblTests' table. You can move, or remove it, as needed.

this.tblTestsTableAdapter.Fill(this.dataSetTests.tblTests);

}

private void FillComboUsers()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userFirstName, userLastName " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1) + " " + dataReader.GetString(2);

comboUsers.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill users combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboForms()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

ComboForms.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill forms combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void AddButtonClick(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string[] arr = comboUsers.Text.Split(' ');

string[] arr2 = ComboForms.Text.Split(' ');

string str = string.Format

("INSERT INTO tblTests " +

"(testFormID, testUserID, testDate) " +

" VALUES ( {0}, {1}, \"{2}\")",

arr2[0], arr[0], testDate.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblTests ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblTests failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס הוספת משתמשים****:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormAddUsers : Form

{

private OleDbConnection dataConnection;

public FormAddUsers(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

FillCityCombo();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillCityCombo() // Populate cities combobox

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cityName " +

"FROM tblCities " +

"ORDER BY cityName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

cityUser.Items.Add(dataReader.GetString(0));

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill cities combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void label11\_Click(object sender, EventArgs e)

{

}

private void label10\_Click(object sender, EventArgs e)

{

}

private void browseClick(object sender, EventArgs e)

{

DialogResult dlgResult = openFileDialog2.ShowDialog();

string pictureFileName = openFileDialog2.FileName;

pictureBox.ImageLocation = pictureFileName;

pictureUser.Text = pictureFileName;

}

private void FormAddUsers\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetUsers.tblUsers' table. You can move, or remove it, as needed.

this.tblUsersTableAdapter.Fill(this.dataSetUsers.tblUsers);

}

static bool IsIdValid(string str)

{

int[] arr = new int[9];

int k = 8;

int sum = 0;

int num;

if (str.Length > 9 || !int.TryParse(str, out num))

return false;

for (int i = str.Length - 1; i >= 0; i--)

{

arr[k] = str[i] - '0';

k--;

}

for (int i = 1; i < 9; i = i + 2)

{

arr[i] = arr[i] \* 2;

arr[i] = arr[i] / 10 + arr[i] % 10;

sum = sum + arr[i] + arr[i - 1];

}

int bikoret = sum / 10 \* 10;

if (bikoret < sum)

bikoret = bikoret + 10;

bikoret = bikoret - sum;

return arr[8] == bikoret;

}

private void AddButton(object sender, EventArgs e)

{

try

{

if (!IsIdValid(idUser.Text))

{

MessageBox.Show("User ID is not valid");

return;

}

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblUsers " +

"(userID, userFirstName, userLastName, userBirthDate, userAddress, userCity, userPhone, userMail, userPassword, userIsAdmin, userPicture) " +

" VALUES ( \"{0}\", \"{1}\", \"{2}\", \"{3}\", \"{4}\", \"{5}\", \"{6}\", \"{7}\", \"{8}\", {9}, \"{10}\")",

idUser.Text, firstNameUser.Text, lastNameUser.Text, birthDateUser.Text,

addressUser.Text, cityUser.Text, phoneUser.Text, mailUser.Text, passwordUser.Text, false, pictureUser.Text);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

MessageBox.Show("Insert into tblUsers ended successfully");

RefreshDataGridView();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblUsers failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView1.DataSource = tbl;

dataGridView1.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh dataGridView failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס תרשים שאלות לתכונות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormChartQuestionsByPropertiy : Form

{

private OleDbConnection dataConnection;

private List<string> CharNames = new List<string>();

private List<int> NumberOfQuestions = new List<int>();

public FormChartQuestionsByPropertiy(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

InitializeComponent();

FillArrs();

ShowChart();

EditDataGridView();

}

private void FillArrs()

{

try

{

int k = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT charName " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

CharNames.Add(dataReader.GetString(0));

NumberOfQuestions.Add(CountNumberOfQuestions(dataReader.GetString(0)));

k++;

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblSubjects failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int CountNumberOfQuestions(string charName)

{

try

{

int x = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcCharOrder " +

"FROM tblQuestionsForChar " +

"WHERE qfcCharName = \"" + charName + "\" " +

"ORDER BY qfcCharOrder";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while(dataReader.Read())

{

x++;

}

dataReader.Close();

return x;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

private void EditDataGridView()

{

try

{

for (int i = 0; i < CharNames.Count; i++)

{

DataGridViewRow row = (DataGridViewRow)dataGridView1.Rows[0].Clone();

row.Cells[0].Value = CharNames[i];

row.Cells[1].Value = NumberOfQuestions[i].ToString();

dataGridView1.Rows.Add(row);

}

}

catch (Exception ex)

{

MessageBox.Show("Edit listview item failed " + ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void ShowChart()

{

try

{

chart1.ChartAreas[0].AxisX.LabelStyle.Angle = -45;

chart1.ChartAreas["ChartArea1"].AxisX.Interval = 1;

chart1.ChartAreas["ChartArea1"].AxisX.ScaleView.Zoom(1, 10);

chart1.ChartAreas["ChartArea1"].CursorX.IsUserSelectionEnabled = true;

chart1.ChartAreas["ChartArea1"].AxisX.ScaleView.Zoomable = true;

chart1.ChartAreas["ChartArea1"].AxisX.ScrollBar.IsPositionedInside = true;

chart1.ChartAreas["ChartArea1"].AxisY.ScrollBar.IsPositionedInside = true;

for (int i = 0; i < CharNames.Count; i++)

{

chart1.Series["מספר שאלות"].Points.AddXY(CharNames[i], NumberOfQuestions[i]);

}

}

catch (Exception ex)

{

MessageBox.Show("Show chart failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס תרשים שאלות למבחנים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormChartQuestionsByTest : Form

{

private OleDbConnection dataConnection;

private List<string> TestNumbers = new List<string>();

private List<int> NumberOfQuestions = new List<int>();

public FormChartQuestionsByTest(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

InitializeComponent();

FillArrs();

ShowChart();

EditDataGridView();

}

private void FillArrs()

{

try

{

int k = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT testID, testFormID " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

TestNumbers.Add(dataReader.GetInt32(0).ToString());

NumberOfQuestions.Add(CountNumberOfQuestions(dataReader.GetInt32(1).ToString()));

k++;

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblSubjects failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int CountNumberOfQuestions(string formID)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT MAX(qifOrderNum) FROM tblQuestionsInForm WHERE qifFormID = " + formID + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

return dataReader.GetInt32(0);

}

}

catch (Exception err)

{

MessageBox.Show("tblQuestionsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

private void EditDataGridView()

{

try

{

for (int i = 0; i < TestNumbers.Count; i++)

{

DataGridViewRow row = (DataGridViewRow)dataGridView1.Rows[0].Clone();

row.Cells[0].Value = TestNumbers[i];

row.Cells[1].Value = NumberOfQuestions[i].ToString();

dataGridView1.Rows.Add(row);

}

}

catch (Exception ex)

{

MessageBox.Show("Edit listview item failed " + ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void ShowChart()

{

try

{

chart1.ChartAreas[0].AxisX.LabelStyle.Angle = -45;

chart1.ChartAreas["ChartArea1"].AxisX.Interval = 1;

chart1.ChartAreas["ChartArea1"].AxisX.ScaleView.Zoom(1, 10);

chart1.ChartAreas["ChartArea1"].CursorX.IsUserSelectionEnabled = true;

chart1.ChartAreas["ChartArea1"].AxisX.ScaleView.Zoomable = true;

chart1.ChartAreas["ChartArea1"].AxisX.ScrollBar.IsPositionedInside = true;

chart1.ChartAreas["ChartArea1"].AxisY.ScrollBar.IsPositionedInside = true;

for (int i = 0; i < TestNumbers.Count; i++)

{

chart1.Series["מספר שאלות"].Points.AddXY("מבחן מספר " + TestNumbers[i], NumberOfQuestions[i]);

}

}

catch (Exception ex)

{

MessageBox.Show("Show chart failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס תרשים שאלות לטופסי בחינה****:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormChartQuestionsInTestForm : Form

{

private OleDbConnection dataConnection;

private List<string> FormIDs = new List<string>();

private List<string> FormNames = new List<string>();

private List<int> NumberOfQuestions = new List<int>();

public FormChartQuestionsInTestForm(OleDbConnection dataConnection)

{

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

InitializeComponent();

FillArrs();

ShowChart();

EditDataGridView();

}

private void FillArrs()

{

try

{

int k = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

FormIDs.Add(dataReader.GetInt32(0).ToString());

FormNames.Add(dataReader.GetString(1));

NumberOfQuestions.Add(CountNumberOfQuestions(dataReader.GetInt32(0).ToString()));

k++;

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblSubjects failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int CountNumberOfQuestions(string formID)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT MAX(qifOrderNum) FROM tblQuestionsInForm WHERE qifFormID = " + formID + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

return dataReader.GetInt32(0);

}

}

catch (Exception err)

{

MessageBox.Show("tblQuestionsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

private void EditDataGridView()

{

try

{

for (int i = 0; i < FormIDs.Count; i++)

{

DataGridViewRow row = (DataGridViewRow)dataGridView1.Rows[0].Clone();

row.Cells[0].Value = FormIDs[i];

row.Cells[1].Value = FormNames[i];

row.Cells[2].Value = NumberOfQuestions[i].ToString();

dataGridView1.Rows.Add(row);

}

}

catch (Exception ex)

{

MessageBox.Show("Edit listview item failed " + ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void ShowChart()

{

try

{

chart1.ChartAreas[0].AxisX.LabelStyle.Angle = -45;

chart1.ChartAreas["ChartArea1"].AxisX.Interval = 1;

chart1.ChartAreas["ChartArea1"].AxisX.ScaleView.Zoom(1, 10);

chart1.ChartAreas["ChartArea1"].CursorX.IsUserSelectionEnabled = true;

chart1.ChartAreas["ChartArea1"].AxisX.ScaleView.Zoomable = true;

chart1.ChartAreas["ChartArea1"].AxisX.ScrollBar.IsPositionedInside = true;

chart1.ChartAreas["ChartArea1"].AxisY.ScrollBar.IsPositionedInside = true;

for (int i = 0; i < FormIDs.Count; i++)

{

chart1.Series["מספר שאלות"].Points.AddXY(FormIDs[i] + " - " + FormNames[i], NumberOfQuestions[i]);

}

}

catch (Exception ex)

{

MessageBox.Show("Show chart failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס התחברות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormLogin : Form

{

private OleDbConnection dataConnection;

private bool enter = false;

public FormLogin()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

OpenDb();

}

private void OpenDb()

{

dataConnection = new OleDbConnection();

try

{

dataConnection.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\\Projects\_2023\\Project\_OriBurg\\Access\\dbReliabilityTest.accdb";

dataConnection.Open();

}

catch (Exception e)

{

MessageBox.Show("Error accessing the database: " +

e.Message,

"Errors",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

}

private void CheckButtonClick(object sender, EventArgs e)

{

String line, password;

bool isManager;

int id;

try

{

if (!IsIdValid(idNumber.Text))

{

MessageBox.Show("User ID is not valid");

return;

}

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userPassword, userIsAdmin, userFirstName, userLastName, userPicture " +

"FROM tblUsers " + "WHERE userID = " + this.idNumber.Text;

OleDbDataReader dataReader = datacommand.ExecuteReader();

dataReader.Read();

id = dataReader.GetInt32(0);

password = dataReader.GetString(1);

isManager = dataReader.GetBoolean(2);

if (password == this.passwordText.Text)

{

if(enter == false)

{

name.Text = "ברוך הבא: " + dataReader.GetString(3) + " " + dataReader.GetString(4);

pictureBox.ImageLocation = dataReader.GetString(5);

checkButton.Text = "המשך";

idNumber.Enabled = false;

passwordText.Enabled = false;

enter = true;

}

else

{

idNumber.Enabled = true;

passwordText.Enabled = true;

enter = false;

name.Text = "";

checkButton.Text = "כניסה";

pictureBox.ImageLocation = null;

this.Hide();

FormMenu frMenu = new FormMenu(dataConnection, isManager, idNumber.Text);

frMenu.Show();

frMenu.Disposed += new EventHandler(frMenu\_Disposed);

}

}

else

{

line = "Invalid password: " + this.passwordText.Text;

MessageBox.Show(line, "Error");

}

}

catch (Exception err)

{

line = "Invalid id: " + this.idNumber.Text + err.Message;

MessageBox.Show(line, "Error");

}

}

static bool IsIdValid(string str)

{

int[] arr = new int[9];

int k = 8;

int sum = 0;

int num;

if (str.Length > 9 || !int.TryParse(str, out num))

return false;

for (int i = str.Length - 1; i >= 0; i--)

{

arr[k] = str[i] - '0';

k--;

}

for (int i = 1; i < 9; i = i + 2)

{

arr[i] = arr[i] \* 2;

arr[i] = arr[i] / 10 + arr[i] % 10;

sum = sum + arr[i] + arr[i - 1];

}

int bikoret = sum / 10 \* 10;

if (bikoret < sum)

bikoret = bikoret + 10;

bikoret = bikoret - sum;

return arr[8] == bikoret;

}

void frMenu\_Disposed(object sender, EventArgs e)

{

this.idNumber.Text = "";

this.passwordText.Text = "";

this.Show();

this.Activate();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label1.Location = new Point((scrWidth - label1.Size.Width) / 2 +60, label1.Location.Y);

label2.Location = new Point((scrWidth - label2.Size.Width) / 2 +60, label2.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

passwordText.Location = new Point((scrWidth - passwordText.Size.Width) / 2-30, passwordText.Location.Y);

idNumber.Location = new Point((scrWidth - idNumber.Size.Width) / 2 -30, idNumber.Location.Y);

checkButton.Location = new Point((scrWidth - checkButton.Size.Width) / 2, checkButton.Location.Y);

pictureBox.Location = new Point((scrWidth - pictureBox.Size.Width) / 2, pictureBox.Location.Y);

name.Location = new Point((scrWidth - name.Size.Width) / 2 - 80, name.Location.Y);

}

}

}

**קוד של טופס תפריט:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormMenu : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private string ID;

private int scrWidth;

private int scrHeight;

public FormMenu(OleDbConnection dataConnection, bool isManager, string ID)

{

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

this.ID = ID;

WindowState = FormWindowState.Maximized;

if(!isManager)

{

tablesToolStripMenuItem.Enabled = false;

addToolStripMenuItem.Enabled = false;

updateToolStripMenuItem.Enabled = false;

}

}

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{

}

void Form\_Disposed(object sender, EventArgs e)

{

this.Show();

this.Activate();

}

private void questionsForCharToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void UsersClick(object sender, EventArgs e)

{

this.Hide();

FormTblUsers frUsers = new FormTblUsers();

frUsers.Show();

frUsers.Disposed += new EventHandler(Form\_Disposed);

}

private void CitiesClick(object sender, EventArgs e)

{

this.Hide();

FormTblCities frCities = new FormTblCities();

frCities.Show();

frCities.Disposed += new EventHandler(Form\_Disposed);

}

private void QuestionsClick(object sender, EventArgs e)

{

this.Hide();

FormTblQuestions frQuestions = new FormTblQuestions();

frQuestions.Show();

frQuestions.Disposed += new EventHandler(Form\_Disposed);

}

private void CharactersClick(object sender, EventArgs e)

{

this.Hide();

FormTblCharacters frCharacters = new FormTblCharacters();

frCharacters.Show();

frCharacters.Disposed += new EventHandler(Form\_Disposed);

}

private void QuestionsForCharClick(object sender, EventArgs e)

{

this.Hide();

FormTblQuestionsForChar frQuestionsForChar = new FormTblQuestionsForChar();

frQuestionsForChar.Show();

frQuestionsForChar.Disposed += new EventHandler(Form\_Disposed);

}

private void FormsClick(object sender, EventArgs e)

{

this.Hide();

FormTblForms frForms = new FormTblForms();

frForms.Show();

frForms.Disposed += new EventHandler(Form\_Disposed);

}

private void CharsInFormClick(object sender, EventArgs e)

{

this.Hide();

FormTblCharsInForm frCharsInForm = new FormTblCharsInForm();

frCharsInForm.Show();

frCharsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void QuestionsInFormClick(object sender, EventArgs e)

{

this.Hide();

FormTblQuestionsInForm frQuestionsInForm = new FormTblQuestionsInForm();

frQuestionsInForm.Show();

frQuestionsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void TestsClick(object sender, EventArgs e)

{

this.Hide();

FormTblTests frTests = new FormTblTests();

frTests.Show();

frTests.Disposed += new EventHandler(Form\_Disposed);

}

private void AnswersInTestClick(object sender, EventArgs e)

{

this.Hide();

FormTblAnswersInTest frAnswersInTest = new FormTblAnswersInTest();

frAnswersInTest.Show();

frAnswersInTest.Disposed += new EventHandler(Form\_Disposed);

}

private void addUsersToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddUsers frAddUsers = new FormAddUsers(dataConnection);

frAddUsers.Show();

frAddUsers.Disposed += new EventHandler(Form\_Disposed);

}

private void addCharactersToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddCharacters frAddCharacters = new FormAddCharacters(dataConnection);

frAddCharacters.Show();

frAddCharacters.Disposed += new EventHandler(Form\_Disposed);

}

private void addCitiesToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddCities frAddCities = new FormAddCities(dataConnection);

frAddCities.Show();

frAddCities.Disposed += new EventHandler(Form\_Disposed);

}

private void addQuestionsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddQuestions frAddQuestions = new FormAddQuestions(dataConnection);

frAddQuestions.Show();

frAddQuestions.Disposed += new EventHandler(Form\_Disposed);

}

private void addQuestionsForCharToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddQuestionsForChar frAddQuestionsForChar = new FormAddQuestionsForChar(dataConnection);

frAddQuestionsForChar.Show();

frAddQuestionsForChar.Disposed += new EventHandler(Form\_Disposed);

}

private void addFormsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddForms frAddForms = new FormAddForms(dataConnection);

frAddForms.Show();

frAddForms.Disposed += new EventHandler(Form\_Disposed);

}

private void addCharsInFormToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddCharsInForm frAddCharsInForm = new FormAddCharsInForm(dataConnection);

frAddCharsInForm.Show();

frAddCharsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void addQuestionsInFormToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddQuestionsInForm frAddQuestionsInForm = new FormAddQuestionsInForm(dataConnection);

frAddQuestionsInForm.Show();

frAddQuestionsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void addTestsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddTests frAddTests = new FormAddTests(dataConnection);

frAddTests.Show();

frAddTests.Disposed += new EventHandler(Form\_Disposed);

}

private void addAnswersInTestToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormAddAnswersInTest frAddAnswersInTest = new FormAddAnswersInTest(dataConnection);

frAddAnswersInTest.Show();

frAddAnswersInTest.Disposed += new EventHandler(Form\_Disposed);

}

private void updateUsersToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateUsers frUpdateUsers = new FormUpdateUsers(dataConnection, isManager);

frUpdateUsers.Show();

frUpdateUsers.Disposed += new EventHandler(Form\_Disposed);

}

private void updateQuestionsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateQuestions frUpdateQuestions = new FormUpdateQuestions(dataConnection, isManager);

frUpdateQuestions.Show();

frUpdateQuestions.Disposed += new EventHandler(Form\_Disposed);

}

private void updateCharactersToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateCharacters frUpdateCharacters = new FormUpdateCharacters(dataConnection, isManager);

frUpdateCharacters.Show();

frUpdateCharacters.Disposed += new EventHandler(Form\_Disposed);

}

private void updateQuestionForCharToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateQuestionsForChar frUpdateQuestionsForChar = new FormUpdateQuestionsForChar(dataConnection, isManager);

frUpdateQuestionsForChar.Show();

frUpdateQuestionsForChar.Disposed += new EventHandler(Form\_Disposed);

}

private void updateFormsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateForms frUpdateForms = new FormUpdateForms(dataConnection, isManager);

frUpdateForms.Show();

frUpdateForms.Disposed += new EventHandler(Form\_Disposed);

}

private void updateQuestionsInFormToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateQuestionsInForm frUpdateQuestionsInForm = new FormUpdateQuestionsInForm(dataConnection, isManager);

frUpdateQuestionsInForm.Show();

frUpdateQuestionsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void updateTestsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateTests frUpdateTests = new FormUpdateTests(dataConnection, isManager);

frUpdateTests.Show();

frUpdateTests.Disposed += new EventHandler(Form\_Disposed);

}

private void updateAnswersInTestToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormUpdateAnswersInTest frUpdateAnswersInTest = new FormUpdateAnswersInTest(dataConnection, isManager);

frUpdateAnswersInTest.Show();

frUpdateAnswersInTest.Disposed += new EventHandler(Form\_Disposed);

}

private void createTestToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormTestCreate frTestCreate = new FormTestCreate(dataConnection,ID);

frTestCreate.Show();

frTestCreate.Disposed += new EventHandler(Form\_Disposed);

}

private void testFormReportToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormRptTestForm frRptTestForm = new FormRptTestForm(dataConnection, isManager);

frRptTestForm.Show();

frRptTestForm.Disposed += new EventHandler(Form\_Disposed);

}

private void testQustionsByCharacterToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormRptQuestionsByCharacter frRptQuestionsByCharacter = new FormRptQuestionsByCharacter(dataConnection, isManager);

frRptQuestionsByCharacter.Show();

frRptQuestionsByCharacter.Disposed += new EventHandler(Form\_Disposed);

}

private void testToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormRptTest frRptTest = new FormRptTest(dataConnection, isManager);

frRptTest.Show();

frRptTest.Disposed += new EventHandler(Form\_Disposed);

}

private void doATestToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormTestExamination frTestExamination = new FormTestExamination(dataConnection, isManager);

frTestExamination.Show();

frTestExamination.Disposed += new EventHandler(Form\_Disposed);

}

private void searchToolStripMenuItem1\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchUsers frSearchUsers = new FormSearchUsers(dataConnection);

frSearchUsers.Show();

frSearchUsers.Disposed += new EventHandler(Form\_Disposed);

}

private void searchQuestionsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchQuestions frSearchQuestions = new FormSearchQuestions(dataConnection);

frSearchQuestions.Show();

frSearchQuestions.Disposed += new EventHandler(Form\_Disposed);

}

private void searchCharactersToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchCharacters frSearchCharacters = new FormSearchCharacters(dataConnection);

frSearchCharacters.Show();

frSearchCharacters.Disposed += new EventHandler(Form\_Disposed);

}

private void searchFormsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchForms frSearchForms = new FormSearchForms(dataConnection);

frSearchForms.Show();

frSearchForms.Disposed += new EventHandler(Form\_Disposed);

}

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label1.Location = new Point((scrWidth - label1.Size.Width) / 2, label1.Location.Y);

label2.Location = new Point((scrWidth - label2.Size.Width) / 2, label2.Location.Y);

}

private void חיפושתשובותבמבחןToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchAnswersInTest frSearchAnswersInTest = new FormSearchAnswersInTest(dataConnection);

frSearchAnswersInTest.Show();

frSearchAnswersInTest.Disposed += new EventHandler(Form\_Disposed);

}

private void חיפושתכונותבטופסToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchCharsInForm frSearchCharsInForm = new FormSearchCharsInForm(dataConnection);

frSearchCharsInForm.Show();

frSearchCharsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void חיפושעריםToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchCities frSearchCities = new FormSearchCities(dataConnection);

frSearchCities.Show();

frSearchCities.Disposed += new EventHandler(Form\_Disposed);

}

private void חיפוששאלותלתכונהToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchQuestionsForChar frSearchQuestionsForChar = new FormSearchQuestionsForChar(dataConnection);

frSearchQuestionsForChar.Show();

frSearchQuestionsForChar.Disposed += new EventHandler(Form\_Disposed);

}

private void חיפוששאלותבטופסToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchQuestionsInForm frSearchQuestionsInForm = new FormSearchQuestionsInForm(dataConnection);

frSearchQuestionsInForm.Show();

frSearchQuestionsInForm.Disposed += new EventHandler(Form\_Disposed);

}

private void חיפושמבחניםToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormSearchTests frSearchTests = new FormSearchTests(dataConnection);

frSearchTests.Show();

frSearchTests.Disposed += new EventHandler(Form\_Disposed);

}

private void תרשיםשאלותלטופסיבחינהToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormChartQuestionsInTestForm frChartQuestionsInTestForm = new FormChartQuestionsInTestForm(dataConnection);

frChartQuestionsInTestForm.Show();

frChartQuestionsInTestForm.Disposed += new EventHandler(Form\_Disposed);

}

private void תרשיםשאלותלתכונותToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormChartQuestionsByPropertiy frChartQuestionsByPropertiy = new FormChartQuestionsByPropertiy(dataConnection);

frChartQuestionsByPropertiy.Show();

frChartQuestionsByPropertiy.Disposed += new EventHandler(Form\_Disposed);

}

private void תרשיםשאלותלמבחניםToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Hide();

FormChartQuestionsByTest frChartQuestionsByTest = new FormChartQuestionsByTest(dataConnection);

frChartQuestionsByTest.Show();

frChartQuestionsByTest.Disposed += new EventHandler(Form\_Disposed);

}

}

}

**קוד של טופס דוח שאלות לתכונה:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormRptQuestionsByCharacter : Form

{

private OleDbConnection dataConnection;

private string saveColor = "";

private string charName;

private string QuesNumber;

private string QuesText;

private string QuesAnswers;

private string QuesFromValue;

private string QuesToValue;

public FormRptQuestionsByCharacter(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormRptQuestionsByCharacter\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharacters.tblCharacters' table. You can move, or remove it, as needed.

this.tblCharactersTableAdapter.Fill(this.dataSetCharacters.tblCharacters);

}

private void ShowReportButton(object sender, EventArgs e)

{

listView1.Items.Clear();

try

{

int counter = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcCharOrder, qfcFromValue, qfcToValue " +

"FROM tblQuestionsForChar " +

"WHERE qfcCharName = \"" + dataGridView1.SelectedRows[0].Cells[0].Value.ToString() + "\" " +

"ORDER BY qfcCharOrder";

OleDbDataReader dataReader = datacommand.ExecuteReader();

charName = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

while (dataReader.Read())

{

counter++;

string[] s = GetQuesTextAndAnswers(dataReader.GetInt32(0).ToString());

QuesText = s[0];

QuesNumber = dataReader.GetInt32(0).ToString();

QuesAnswers = s[1];

QuesFromValue = dataReader.GetInt32(1).ToString();

QuesToValue = dataReader.GetInt32(2).ToString();

EditListView(counter);

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string[] GetQuesTextAndAnswers(string quesID)

{

string[] arr = new string[2];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesText, quesAnswers " +

"FROM tblQuestions " +

"WHERE quesID = " + quesID + " " +

"ORDER BY quesText";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

arr[0] = dataReader.GetString(0);

arr[1] = dataReader.GetInt32(1).ToString();

return arr;

}

dataReader.Close();

return arr;

}

private void EditListView(int counter)

{

try

{

string[] arr = new string[6];

if (counter == 1)

{ arr[0] = charName; }

arr[1] = QuesNumber;

arr[2] = QuesText;

arr[3] = QuesAnswers;

arr[4] = QuesFromValue;

arr[5] = QuesToValue;

ListViewItem item = new ListViewItem(arr);

if (saveColor != "")

item.ForeColor = Color.FromArgb(int.Parse(saveColor));

listView1.Items.Add(item);

}

catch (Exception ex)

{

MessageBox.Show("Edit listview item failed " + ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void SelectColorButton(object sender, EventArgs e)

{

ColorDialog cd = new ColorDialog();

cd.ShowDialog();

buttonColor.BackColor = cd.Color;

saveColor = buttonColor.BackColor.ToArgb().ToString();

}

private void ClearReportButton(object sender, EventArgs e)

{

listView1.Items.Clear();

}

}

}

**קוד של טופס דוח מבחנים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormRptTest : Form

{

private OleDbConnection dataConnection;

private string saveColor = "";

private string testID;

private string firstName;

private string lastName;

private string QuesOrderNum;

private string QuesText;

private string QuesFromValue;

private string QuesToValue;

private string QuesAnswers;

private string Answer;

private float Score = 0;

private float fakeCount = 0;

private void FormRptTest\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetTests.tblTests' table. You can move, or remove it, as needed.

this.tblTestsTableAdapter.Fill(this.dataSetTests.tblTests);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

public FormRptTest(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

}

private void ShowReportButton(object sender, EventArgs e)

{

listView1.Items.Clear();

Score = 0;

fakeCount = 0;

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT testFormID, testUserID " +

"FROM tblTests " +

"WHERE testID = " + dataGridView1.SelectedRows[0].Cells[0].Value.ToString() + " " +

"ORDER BY testFormID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

testID = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

while (dataReader.Read())

{

string[] userData = GetUserInfo(dataReader.GetInt32(1).ToString());

firstName = userData[0];

lastName = userData[1];

List<string> charData = TestChars(dataReader.GetInt32(0).ToString());

label2.Text = "תכונות נבדקות: ";

foreach(string s in charData)

{

label2.Text = label2.Text + s + ", ";

}

label2.Text = label2.Text.Substring(0,label2.Text.Length - 2);

FormInfo(dataReader.GetInt32(0).ToString());

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblForms failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private List<string> TestChars(string formID)

{

try

{

List<string> arr = new List<string>();

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cifCharName " +

"FROM tblCharsInForm " +

"WHERE cifFormID = " + formID + " " +

"ORDER BY cifCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

arr.Add(dataReader.GetString(0));

}

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private string[] GetUserInfo(string userID)

{

try

{

string[] arr = new string[2];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userFirstName, userLastName " +

"FROM tblUsers " +

"WHERE userID = " + userID + " " +

"ORDER BY userFirstName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

arr[0] = dataReader.GetString(0);

arr[1] = dataReader.GetString(1);

}

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblUsers failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private void FormInfo(string formID)

{

try

{

int counter = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifOrderNum, qifQuestionID " +

"FROM tblQuestionsInForm " +

"WHERE qifFormID = " + formID + " " +

"ORDER BY qifOrderNum";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

QuesOrderNum = dataReader.GetInt32(0).ToString();

string[] quesData = GetQuesData(dataReader.GetInt32(1).ToString());

QuesText = quesData[0];

QuesAnswers = quesData[1];

Answer = GetAnswerInTest(testID, QuesOrderNum);

string[] qustForCharData = GetQustForCharInfo(dataReader.GetInt32(1).ToString(), Answer, formID);

QuesFromValue = qustForCharData[0];

QuesToValue = qustForCharData[1];

counter++;

EditListView(counter);

}

dataReader.Close();

int finalScore = Convert.ToInt32(Score/(counter-fakeCount)\*100);

scoreShow.Text = "ציונך הוא " + finalScore.ToString();

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsInForm failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string[] GetQuesData(string quesID)

{

string[] arr = new string[2];

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesText, quesAnswers " +

"FROM tblQuestions " +

"WHERE quesID = " + quesID + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

dataReader.Read();

arr[0] = dataReader.GetString(0);

arr[1] = dataReader.GetInt32(1).ToString();

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestions failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private bool CheckForNotFake(string character, string formID)

{

try

{

string[] arr = new string[3];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cifCharName " +

"FROM tblCharsInForm " +

"WHERE cifFormID = " + formID + " " +

"ORDER BY cifCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (character.Equals(dataReader.GetString(0)))

{ return true;}

}

dataReader.Close();

return false;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

private string[] GetQustForCharInfo(string CharOrder, string Answer, string formID)

{

try

{

string[] arr = new string[3];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcFromValue, qfcToValue, qfcCharName " +

"FROM tblQuestionsForChar " +

"WHERE qfcCharOrder = " + CharOrder + " " +

"ORDER BY qfcFromValue";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

arr[0] = dataReader.GetInt32(0).ToString();

arr[1] = dataReader.GetInt32(1).ToString();

if(dataReader.GetInt32(0) <= Convert.ToInt32(Answer) && Convert.ToInt32(Answer) <= dataReader.GetInt32(1) && CheckForNotFake(dataReader.GetString(2), formID))

{ Score++; }

if (!CheckForNotFake(dataReader.GetString(2), formID)) { fakeCount++; }

}

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private string GetAnswerInTest(string testID, string quesOrderNum)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT aitOrderNum, aitAnswer " +

"FROM tblAnswersInTest " +

"WHERE aitTestID = " + testID + " " +

"ORDER BY aitOrderNum";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(quesOrderNum))

{

return dataReader.GetInt32(1).ToString();

}

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblAnswersInTest failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private void EditListView(int counter)

{

try

{

string[] arr = new string[10];

if (counter == 1)

{

arr[0] = testID;

arr[1] = firstName;

arr[2] = lastName;

}

arr[3] = QuesOrderNum;

arr[4] = QuesText;

arr[5] = QuesFromValue;

arr[6] = QuesToValue;

arr[7] = Answer;

arr[8] = QuesAnswers;

ListViewItem item = new ListViewItem(arr);

if (saveColor != "")

item.ForeColor = Color.FromArgb(int.Parse(saveColor));

listView1.Items.Add(item);

}

catch (Exception ex)

{

MessageBox.Show("Edit listview item failed " + ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void SelectColorButton(object sender, EventArgs e)

{

ColorDialog cd = new ColorDialog();

cd.ShowDialog();

buttonColor.BackColor = cd.Color;

saveColor = buttonColor.BackColor.ToArgb().ToString();

}

private void ClearReportButton(object sender, EventArgs e)

{

listView1.Items.Clear();

scoreShow.Text = "";

label2.Text = "";

}

}

}

**קוד של טופס דוח טפסי מבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormRptTestForm : Form

{

private OleDbConnection dataConnection;

private string saveColor = "";

private string formID;

private string formName;

private string QuesText;

private string QuesChar;

private string QuesFromValue;

private string QuesToValue;

public FormRptTestForm(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormRptTestForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetForms.tblForms' table. You can move, or remove it, as needed.

this.tblFormsTableAdapter.Fill(this.dataSetForms.tblForms);

}

private void ShowReportButton(object sender, EventArgs e)

{

listView1.Items.Clear();

if (GetFormChars(dataGridView1.SelectedRows[0].Cells[0].Value.ToString()) == "")

{ charLabel.Text = "לטופס זה אין שום תכונות מוגדרות"; }

else { charLabel.Text = "תכונות הטופס: " + GetFormChars(dataGridView1.SelectedRows[0].Cells[0].Value.ToString()); }

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"WHERE formID = " + dataGridView1.SelectedRows[0].Cells[0].Value.ToString() + " " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

formID = dataReader.GetInt32(0).ToString();

formName = dataReader.GetString(1);

GetQuestion();

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblForms failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetFormChars(string formID)

{

try

{

string str = "";

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cifCharName " +

"FROM tblCharsInForm " +

"WHERE cifFormID = " + dataGridView1.SelectedRows[0].Cells[0].Value.ToString() + " " +

"ORDER BY cifCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = str + dataReader.GetString(0) + ", ";

}

dataReader.Close();

if(str.Length == 0) { return ""; }

return str.Substring(0, str.Length - 2);

}

catch (Exception ex)

{

MessageBox.Show("Select tblCharInForms failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "ERROR";

}

private void GetQuestion()

{

try

{

int counter = 0;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifQuestionID " +

"FROM tblQuestionsInForm " +

"WHERE qifFormID = " + formID + " " +

"ORDER BY qifQuestionID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

QuesText = dataReader.GetInt32(0) + " - " + GetQuesText(dataReader.GetInt32(0));

string[] charInfo = GetQuesChar(dataReader.GetInt32(0)).Split('|');

QuesChar = charInfo[0];

QuesFromValue = charInfo[1];

QuesToValue = charInfo[2];

counter++;

EditListView(counter);

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("GetQuestion failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetQuesText(int quesID)

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesText " +

"FROM tblQuestions " +

"WHERE quesID = " + quesID + " " +

"ORDER BY quesText";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

return dataReader.GetString(0);

}

dataReader.Close();

return "Error";

}

private string GetQuesChar(int quesID)

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcCharName, qfcFromValue, qfcToValue " +

"FROM tblQuestionsForChar " +

"WHERE qfcCharOrder = " + quesID + " " +

"ORDER BY qfcCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

return dataReader.GetString(0) + "|" + dataReader.GetInt32(1) + "|" + dataReader.GetInt32(2);

}

dataReader.Close();

return "Error";

}

private void EditListView(int counter)

{

try

{

string[] arr = new string[6];

if (counter == 1)

{

arr[0] = formID;

arr[1] = formName;

}

arr[2] = QuesText;

arr[3] = QuesChar;

arr[4] = QuesFromValue;

arr[5] = QuesToValue;

ListViewItem item = new ListViewItem(arr);

if (saveColor != "")

item.ForeColor = Color.FromArgb(int.Parse(saveColor));

listView1.Items.Add(item);

}

catch (Exception ex)

{

MessageBox.Show("Edit listview item failed " + ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void SelectColorButton(object sender, EventArgs e)

{

ColorDialog cd = new ColorDialog();

cd.ShowDialog();

buttonColor.BackColor = cd.Color;

saveColor = buttonColor.BackColor.ToArgb().ToString();

}

private void ClearReportButton(object sender, EventArgs e)

{

listView1.Items.Clear();

charLabel.Text = "";

}

private void listView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

**קוד של טופס חיפוש תשובות במבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchAnswersInTest : Form

{

private void FormSearchAnswersInTest\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetAnswersInTest.tblAnswersInTest' table. You can move, or remove it, as needed.

this.tblAnswersInTestTableAdapter.Fill(this.dataSetAnswersInTest.tblAnswersInTest);

}

private OleDbConnection dataConnection;

public FormSearchAnswersInTest(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblAnswersInTest WHERE " +

"aitTestID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"aitOrderNum LIKE \"%" + searchStr.Text + "%\" OR \n" +

"aitAnswer LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY aitTestID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblAnswersInTest " +

"ORDER BY aitTestID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס חיפוש תכונות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchCharacters : Form

{

private OleDbConnection dataConnection;

public FormSearchCharacters(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private void FormSearchCharacters\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharacters.tblCharacters' table. You can move, or remove it, as needed.

this.tblCharactersTableAdapter.Fill(this.dataSetCharacters.tblCharacters);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharacters WHERE " +

"charName LIKE \"%" + searchStr.Text + "%\" OR \n" +

"charDesc LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY charName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס חיפוש תכונות בטופס:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchCharsInForm : Form

{

private OleDbConnection dataConnection;

public FormSearchCharsInForm(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharsInForm WHERE " +

"cifFormID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"cifCharName LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY cifFormID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharsInForm " +

"ORDER BY cifFormID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormSearchCharsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharsInForm.tblCharsInForm' table. You can move, or remove it, as needed.

this.tblCharsInFormTableAdapter.Fill(this.dataSetCharsInForm.tblCharsInForm);

}

}

}

**קוד של טופס חיפוש ערים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchCities : Form

{

private OleDbConnection dataConnection;

public FormSearchCities(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCities WHERE " +

"cityName LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY cityName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCities " +

"ORDER BY cityName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormSearchCities\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCities.tblCities' table. You can move, or remove it, as needed.

this.tblCitiesTableAdapter.Fill(this.dataSetCities.tblCities);

}

}

}

**קוד של טופס חיפוש טפסים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchForms : Form

{

private OleDbConnection dataConnection;

public FormSearchForms(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblForms WHERE " +

"formID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"formDate LIKE \"%" + searchStr.Text + "%\" OR \n" +

"formCreatorID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"formName LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY formID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridForms.DataSource = tbl;

dataGridForms.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridForms.DataSource = tbl;

dataGridForms.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormSearchForms\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetForms.tblForms' table. You can move, or remove it, as needed.

this.tblFormsTableAdapter.Fill(this.dataSetForms.tblForms);

}

}

}

**קוד של טופס חיפוש שאלות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchQuestions : Form

{

private OleDbConnection dataConnection;

public FormSearchQuestions(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private void FormSearchQuestions\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestions.tblQuestions' table. You can move, or remove it, as needed.

this.tblQuestionsTableAdapter.Fill(this.dataSetQuestions.tblQuestions);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestions WHERE " +

"quesID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"quesText LIKE \"%" + searchStr.Text + "%\" OR \n" +

"quesAnswers LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY quesID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridQuestions.DataSource = tbl;

dataGridQuestions.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridQuestions.DataSource = tbl;

dataGridQuestions.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס חיפוש שאלות לתכונה:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchQuestionsForChar : Form

{

private OleDbConnection dataConnection;

public FormSearchQuestionsForChar(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsForChar WHERE " +

"qfcCharName LIKE \"%" + searchStr.Text + "%\" OR \n" +

"qfcCharOrder LIKE \"%" + searchStr.Text + "%\" OR \n" +

"qfcFromValue LIKE \"%" + searchStr.Text + "%\" OR \n" +

"qfcToValue LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY qfcCharName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsForChar " +

"ORDER BY qfcCharName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormSearchQuestionsForChar\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsForChar.tblQuestionsForChar' table. You can move, or remove it, as needed.

this.tblQuestionsForCharTableAdapter.Fill(this.dataSetQuestionsForChar.tblQuestionsForChar);

}

}

}

**קוד של טופס חיפוש שאלות בטופס:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchQuestionsInForm : Form

{

private OleDbConnection dataConnection;

public FormSearchQuestionsInForm(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsInForm WHERE " +

"qifFormID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"qifOrderNum LIKE \"%" + searchStr.Text + "%\" OR \n" +

"qifQuestionID LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY qifFormID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsInForm " +

"ORDER BY qifFormID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormSearchQuestionsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsInForm.tblQuestionsInForm' table. You can move, or remove it, as needed.

this.tblQuestionsInFormTableAdapter.Fill(this.dataSetQuestionsInForm.tblQuestionsInForm);

}

}

}

**קוד של טופס חיפוש מבחנים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchTests : Form

{

private OleDbConnection dataConnection;

public FormSearchTests(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblTests WHERE " +

"testID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"testFormID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"testUserID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"testDate LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY testID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Search failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridCharacters.DataSource = tbl;

dataGridCharacters.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FormSearchTests\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetTests.tblTests' table. You can move, or remove it, as needed.

this.tblTestsTableAdapter.Fill(this.dataSetTests.tblTests);

}

}

}

**קוד של טופס חיפוש משתמשים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormSearchUsers : Form

{

private OleDbConnection dataConnection;

public FormSearchUsers(OleDbConnection dataConnection)

{

InitializeComponent();

WindowState = FormWindowState.Maximized;

this.dataConnection = dataConnection;

}

private void FormSearchUsers\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetUsers.tblUsers' table. You can move, or remove it, as needed.

this.tblUsersTableAdapter.Fill(this.dataSetUsers.tblUsers);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblUsers WHERE " +

"userID LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userFirstName LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userLastName LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userAddress LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userCity LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userPhone LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userMail LIKE \"%" + searchStr.Text + "%\" OR \n" +

"userPicture LIKE \"%" + searchStr.Text + "%\" \n" +

"ORDER BY userID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridUsers.DataSource = tbl;

dataGridUsers.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("SearchTblUsers failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonRefresh\_Click(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridUsers.DataSource = tbl;

dataGridUsers.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**קוד של טופס טבלת תשובות במבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblAnswersInTest : Form

{

public FormTblAnswersInTest()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void FormTblAnswersInTest\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetAnswersInTest.tblAnswersInTest' table. You can move, or remove it, as needed.

this.tblAnswersInTestTableAdapter.Fill(this.dataSetAnswersInTest.tblAnswersInTest);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetAnswersInTest changes = (DataSetAnswersInTest)dataSetAnswersInTest.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblAnswersInTest.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblAnswersInTestTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetAnswersInTest.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetAnswersInTest.RejectChanges();

}

}

}

}

**קוד של טופס טבלת תכונות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblCharacters : Form

{

public FormTblCharacters()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblCharacters\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharacters.tblCharacters' table. You can move, or remove it, as needed.

this.tblCharactersTableAdapter.Fill(this.dataSetCharacters.tblCharacters);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetCharacters changes = (DataSetCharacters)dataSetCharacters.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblCharacters.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblCharactersTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetCharacters.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetCharacters.RejectChanges();

}

}

}

}

**קוד של טופס טבלת תכונות בטופס:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblCharsInForm : Form

{

public FormTblCharsInForm()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblCharsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharsInForm.tblCharsInForm' table. You can move, or remove it, as needed.

this.tblCharsInFormTableAdapter.Fill(this.dataSetCharsInForm.tblCharsInForm);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetCharsInForm changes = (DataSetCharsInForm)dataSetCharsInForm.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblCharsInForm.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblCharsInFormTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetCharsInForm.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetCharsInForm.RejectChanges();

}

}

}

}

**קוד של טופס טבלת ערים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblCities : Form

{

public FormTblCities()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblCities\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCities.tblCities' table. You can move, or remove it, as needed.

this.tblCitiesTableAdapter.Fill(this.dataSetCities.tblCities);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetCities changes = (DataSetCities)dataSetCities.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblCities.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblCitiesTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetCities.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetCities.RejectChanges();

}

}

}

}

**קוד של טופס טבלת טפסים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblForms : Form

{

public FormTblForms()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblForms\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetForms.tblForms' table. You can move, or remove it, as needed.

this.tblFormsTableAdapter.Fill(this.dataSetForms.tblForms);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetForms changes = (DataSetForms)dataSetForms.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblForms.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblFormsTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetForms.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetForms.RejectChanges();

}

}

}

}

**קוד של טופס טבלת שאלות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblQuestions : Form

{

public FormTblQuestions()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblQuestions\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestions.tblQuestions' table. You can move, or remove it, as needed.

this.tblQuestionsTableAdapter.Fill(this.dataSetQuestions.tblQuestions);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView2.Location = new Point((scrWidth - dataGridView2.Size.Width) / 2, dataGridView2.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetQuestions changes = (DataSetQuestions)dataSetQuestions.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblQuestions.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblQuestionsTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetQuestions.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetQuestions.RejectChanges();

}

}

}

}

**קוד של טופס טבלת שאלות לתכונה:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblQuestionsForChar : Form

{

public FormTblQuestionsForChar()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblQuestionsForChar\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsForChar.tblQuestionsForChar' table. You can move, or remove it, as needed.

this.tblQuestionsForCharTableAdapter.Fill(this.dataSetQuestionsForChar.tblQuestionsForChar);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetQuestionsForChar changes = (DataSetQuestionsForChar)dataSetQuestionsForChar.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblQuestionsForChar.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblQuestionsForCharTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetQuestionsForChar.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetQuestionsForChar.RejectChanges();

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

}

}

**קוד של טופס טבלת שאלות בטופס:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblQuestionsInForm : Form

{

public FormTblQuestionsInForm()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblQuestionsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsInForm.tblQuestionsInForm' table. You can move, or remove it, as needed.

this.tblQuestionsInFormTableAdapter.Fill(this.dataSetQuestionsInForm.tblQuestionsInForm);

// TODO: This line of code loads data into the 'dataSetQuestionsInForm.tblQuestionsInForm' table. You can move, or remove it, as needed.

this.tblQuestionsInFormTableAdapter.Fill(this.dataSetQuestionsInForm.tblQuestionsInForm);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetQuestionsInForm changes = (DataSetQuestionsInForm)dataSetQuestionsInForm.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblQuestionsInForm.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblQuestionsInFormTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetQuestionsInForm.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetQuestionsInForm.RejectChanges();

}

}

}

}

**קוד של טופס טבלת מבחנים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblTests : Form

{

public FormTblTests()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblTests\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetTests.tblTests' table. You can move, or remove it, as needed.

this.tblTestsTableAdapter.Fill(this.dataSetTests.tblTests);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetTests changes = (DataSetTests)dataSetTests.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblTests.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblTestsTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetTests.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetTests.RejectChanges();

}

}

}

}

**קוד של טופס טבלת משתמשים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ReliabilityTest

{

public partial class FormTblUsers : Form

{

public FormTblUsers()

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

}

private void FormTblUsers\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetUsers.tblUsers' table. You can move, or remove it, as needed.

this.tblUsersTableAdapter.Fill(this.dataSetUsers.tblUsers);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

saveButton.Location = new Point((scrWidth - saveButton.Size.Width) / 2, saveButton.Location.Y);

}

private void SaveButtonClick(object sender, EventArgs e)

{

try

{

DataSetUsers changes = (DataSetUsers)dataSetUsers.GetChanges();

if (changes == null)

return;

// check for errors

DataTable dt = changes.tblUsers.GetChanges();

DataRow[] badRows = dt.GetErrors();

//find the errors and tell the user

if (badRows.Length > 0)

{

string errorMsg = "";

foreach (DataRow row in badRows)

{

foreach (DataColumn col in row.GetColumnsInError())

{

errorMsg = errorMsg + row.GetColumnsInError() + "\n";

}

}

MessageBox.Show("Errors in data: " + errorMsg,

"Please fix", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//no error found, update the database

int numRows = tblUsersTableAdapter.Update(changes);

MessageBox.Show("Updated " + numRows + " rows", "Success");

dataSetUsers.AcceptChanges();

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Erros",

MessageBoxButtons.OK, MessageBoxIcon.Error);

dataSetUsers.RejectChanges();

}

}

}

}

**קוד של טופס יצירת מבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormTestCreate : Form

{

private OleDbConnection dataConnection;

private string ID;

public FormTestCreate(OleDbConnection dataConnection, string ID)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.ID = ID;

creatorID.Text = ID;

FillCharacters();

FillQuesCount();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillQuesCount()

{

for(int i = 10; i < 21; i++)

{

quesCount.Items.Add(i.ToString());

}

}

private void FillCharacters() // Populate cities combobox

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT charName " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

charactersList.Items.Add(dataReader.GetString(0));

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill cities combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonCreateClick(object sender, EventArgs e)

{

int count = 0;

foreach(var character in charactersList.CheckedItems)

{

count++;

}

if(count < 1 || count > 3)

{

MessageBox.Show("Sorry, you need to select between 1 to 3 characters, you selected- " + count);

return;

}

foreach(var character in charactersList.CheckedItems)

{

if(QuestionsForCharNumber(character.ToString()) < int.Parse(quesCount.Text))

{

MessageBox.Show("Sorry, not enough questions for the character: " + character.ToString() + ", only "+ QuestionsForCharNumber(character.ToString()) + " questions.");

return;

}

}

createButton.Text = "...יוצר";

AddForm();

AddCharsInForm(FormNumber());

AddRandomQuestions(FormNumber());

createButton.Text = "צור טופס בחינה";

MessageBox.Show("!יצירת הטופס הושלמה בהצלחה \n" , "Success",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private int QuestionsForCharNumber(string charName)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcCharName " +

"FROM tblQuestionsForChar " +

"ORDER BY qfcCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

int count = 0;

while (dataReader.Read())

{

if (dataReader.GetString(0).Equals(charName))

{

count++;

}

}

dataReader.Close();

return count;

}

catch (Exception err)

{

MessageBox.Show("QuestionsForCharNumber failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

private void AddForm()

{

try

{

DateTime now = DateTime.Now;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblForms " +

"(formName, formDate, formCreatorID) " +

" VALUES ( \"{0}\", \"{1}\", {2})",

formName.Text, now.Date, ID);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblForms failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int FormNumber()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

int maxNum = 0;

while (dataReader.Read())

{

if (dataReader.GetInt32(0) > maxNum)

{

maxNum = dataReader.GetInt32(0);

}

}

dataReader.Close();

return maxNum;

}

catch (Exception err)

{

MessageBox.Show("Get form number failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

private void AddCharsInForm(int formNum)

{

foreach(var character in charactersList.CheckedItems)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblCharsInForm " +

"(cifFormID, cifCharName) " +

" VALUES ( {0}, \"{1}\")",

formNum, character.ToString());

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblCharsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void AddRandomQuestions(int FormID)

{

List<int> QuesList = new List<int>();

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

QuesList.Add(dataReader.GetInt32(0));

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Get questions for list failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

//

int charCount = 0;

foreach (var character in charactersList.CheckedItems)

{

charCount++;

}

//

Dictionary<string, int> charList = new Dictionary<string, int>();

foreach (var character in charactersList.CheckedItems)

{

charList.Add(character.ToString(), 0);

}

charList.Add("fake", 0);

int fakeQuestionNumber = (int.Parse(quesCount.Text) \* charactersList.CheckedItems.Count) / 10;

List<int> newTestQues = new List<int>();

while (newTestQues.Count != int.Parse(quesCount.Text) \* charactersList.CheckedItems.Count + fakeQuestionNumber) // כמות השאלות ועוד עשר אחוז שאלות דמה

{

Random rnd = new Random(); int randNum = rnd.Next(1,QuesList.Count);

int tempNum = QuesList[randNum];

if(tempNum != -1)

{

bool isDone = false;

foreach(var item in charList)

{

if(item.Value < int.Parse(quesCount.Text) && charCheck(tempNum, item.Key))

{

isDone = true;

charList[item.Key]++;

QuesList[randNum] = -1;

newTestQues.Add(tempNum);

break;

}

}

if (!isDone && charList["fake"] < fakeQuestionNumber)

{

charList["fake"]++;

QuesList[randNum] = -1;

newTestQues.Add(tempNum);

continue;

}

}

}

int count = 1;

foreach(int Q in newTestQues)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblQuestionsInForm " +

"(qifFormID, qifOrderNum, qifQuestionID) " +

" VALUES ( {0}, {1}, {2})",

FormID, count, Q);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

count++;

}

catch (Exception err)

{

MessageBox.Show("Insert into tblQuestionsInForms failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private bool charCheck(int tempNum,string charName)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcCharOrder, qfcCharName " +

"FROM tblQuestionsForChar " +

"ORDER BY qfcCharOrder";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0) == tempNum)

{

if(charName.Equals(dataReader.GetString(1)))

{

dataReader.Close();

return true;

}

dataReader.Close();

return false;

}

}

dataReader.Close();

return false;

}

catch (Exception err)

{

MessageBox.Show("charCheck failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

}

}

**קוד של טופס התחלת ביצוע מבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormTestExamination : Form

{

private OleDbConnection dataConnection;

private bool userChosed;

private bool formChosed;

private int testSpeed;

public FormTestExamination(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

}

private void FormTestExamination\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetForms.tblForms' table. You can move, or remove it, as needed.

this.tblFormsTableAdapter.Fill(this.dataSetForms.tblForms);

// TODO: This line of code loads data into the 'dataSetUsers.tblUsers' table. You can move, or remove it, as needed.

this.tblUsersTableAdapter.Fill(this.dataSetUsers.tblUsers);

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

label1.Location = new Point((scrWidth - label1.Size.Width) / 2, label1.Location.Y);

label2.Location = new Point((scrWidth - label2.Size.Width) / 2, label2.Location.Y);

label4.Location = new Point((scrWidth - label4.Size.Width) / 2, label4.Location.Y);

label3.Location = new Point((scrWidth - label3.Size.Width) / 2, label3.Location.Y);

dataGridView2.Location = new Point((scrWidth - dataGridView2.Size.Width) / 2, dataGridView2.Location.Y);

dataGridView1.Location = new Point((scrWidth - dataGridView1.Size.Width) / 2, dataGridView1.Location.Y);

startButton.Location = new Point((scrWidth - startButton.Size.Width) / 2, startButton.Location.Y);

groupBox1.Location = new Point((scrWidth - groupBox1.Size.Width) / 2, groupBox1.Location.Y);

}

private void CreateTestClick(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

DateTime now = DateTime.Now;

string str = string.Format

("INSERT INTO tblTests " +

"(testFormID, testUserID, testDate) " +

" VALUES ( {0}, {1}, \"{2}\")",

dataGridView2.SelectedRows[0].Cells[0].Value.ToString(), dataGridView1.SelectedRows[0].Cells[0].Value.ToString(), now.Date);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

}

catch (Exception err)

{

MessageBox.Show("Start test failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

int testNumber = CurrentCreatedTest();

string userID = dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

string formID = dataGridView2.SelectedRows[0].Cells[0].Value.ToString();

string[] data = GetUserData(userID);

this.Hide();

if(radioButton1.Checked) { testSpeed = 5; } else if (radioButton2.Checked){ testSpeed = 10; } else { testSpeed = 15; }

FormTestExaminationProcess frTestExaminationProcess = new FormTestExaminationProcess(dataConnection, userID, data[0], data[1], data[2], formID, testNumber.ToString(), testSpeed);

frTestExaminationProcess.Show();

frTestExaminationProcess.Disposed += new EventHandler(Form\_Disposed);

}

void Form\_Disposed(object sender, EventArgs e)

{

this.Show();

this.Activate();

}

private string[] GetUserData(string userID)

{

try

{

string[] arr = new string[3];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userFirstName, userLastName, userPicture " +

"FROM tblUsers " +

"WHERE userID = " + userID + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

dataReader.Read();

arr[0] = dataReader.GetString(0);

arr[1] = dataReader.GetString(1);

arr[2] = dataReader.GetString(2);

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblUsers failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private int CurrentCreatedTest()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT MAX(testID) FROM tblTests ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

return dataReader.GetInt32(0);

}

}

catch (Exception err)

{

MessageBox.Show("QuestionsForCharNumber failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

private void UserChosed(object sender, DataGridViewCellEventArgs e)

{

userChosed = true;

if (userChosed && formChosed)

{

startButton.Enabled = true;

}

}

private void FormChosed(object sender, DataGridViewCellEventArgs e)

{

formChosed = true;

if (userChosed && formChosed)

{

startButton.Enabled = true;

}

}

}

}

**קוד של טופס תהליך ביצוע מבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormTestExaminationProcess : Form

{

private OleDbConnection dataConnection;

private string formID;

private int numberOfQuestions;

private int index = 1;

private string testNumber;

private int testSpeed;

private int tempTimerTime;

private float score = 0;

private float fakeCount = 0;

Timer timer1 = new Timer();

public FormTestExaminationProcess(OleDbConnection dataConnection, string userID, string userFirstName, string userLastName, string pictureUser, string formID, string testNumber, int testSpeed)

{

InitializeComponent();

this.dataConnection = dataConnection;

idUser.Text = userID;

firstNameUser.Text = userFirstName;

lastNameUser.Text = userLastName;

pictureBox.ImageLocation = pictureUser;

this.formID = formID;

numberOfQuestions = NumberOfQuestions();

this.testNumber = testNumber;

this.testSpeed = testSpeed;

List<string> charData = TestChars();

charsShow.Text = "תכונות נבדקות: ";

foreach (string s in charData)

{

charsShow.Text = charsShow.Text + s + ", ";

}

charsShow.Text = charsShow.Text.Substring(0, charsShow.Text.Length - 2);

timer1.Interval = 1000; // set the interval to 1 second

timer1.Enabled = false; // start the timer disabled

timer1.Tick += new EventHandler(timer1\_Tick); // set the event handler for the timer's Tick event

WindowState = FormWindowState.Maximized;

Test();

}

private List<string> TestChars()

{

try

{

List<string> arr = new List<string>();

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cifCharName " +

"FROM tblCharsInForm " +

"WHERE cifFormID = " + formID + " " +

"ORDER BY cifCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

arr.Add(dataReader.GetString(0));

}

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

public void Test()

{

try

{

questionIndex.Text = index + "/" + numberOfQuestions;

tempTimerTime = testSpeed;

StartTimer();

ClearButtons();

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifQuestionID " +

"FROM tblQuestionsInForm " +

"WHERE qifFormID = " + formID + " " +

" AND qifOrderNum = " + index;

OleDbDataReader dataReader = datacommand.ExecuteReader();

dataReader.Read();

string[] data = getQuesText(dataReader.GetInt32(0).ToString());

ques.Text = index + " - " + data[0];

AddButtons(Convert.ToInt32(data[1]));

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsInForm failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void StartTimer()

{

timerShow.Text = "זמן שנותר לענות: " + tempTimerTime.ToString();

timer1.Enabled = true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

tempTimerTime--;

timerShow.Text = "זמן שנותר לענות: " + tempTimerTime.ToString();

if (tempTimerTime == 0)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblAnswersInTest " +

"(aitTestID, aitOrderNum, aitAnswer) " +

" VALUES ( {0}, {1}, {2})",

testNumber, index, -1);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblAnswersInTest failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

//////

if (index >= numberOfQuestions)

{

nextQuesButton.Visible = false;

ClearButtons();

ques.Visible = false;

Answer.Visible = false;

timerShow.Visible = false;

timer1.Enabled = false;

label1.Visible = false;

finish1.Visible = true;

finish2.Visible = true;

finishScore.Visible = true;

finishScore.Text = GetScore();

return;

}

index++;

Test();

}

}

private string GetAnswerInTest(string testID, string quesOrderNum)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT aitOrderNum, aitAnswer " +

"FROM tblAnswersInTest " +

"WHERE aitTestID = " + testID + " " +

"ORDER BY aitOrderNum";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(quesOrderNum))

{

return dataReader.GetInt32(1).ToString();

}

}

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblAnswersInTest failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private string GetScore()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifOrderNum, qifQuestionID " +

"FROM tblQuestionsInForm " +

"WHERE qifFormID = " + formID + " " +

"ORDER BY qifOrderNum";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

string AnswerInQues = GetAnswerInTest(testNumber, dataReader.GetInt32(0).ToString());

GetQustForCharInfo(dataReader.GetInt32(1).ToString(), AnswerInQues);

}

dataReader.Close();

return (Convert.ToInt32(score / (numberOfQuestions-fakeCount) \* 100)).ToString();

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsInForm failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private void GetQustForCharInfo(string CharOrder, string Answer)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qfcFromValue, qfcToValue, qfcCharName " +

"FROM tblQuestionsForChar " +

"WHERE qfcCharOrder = " + CharOrder + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

dataReader.Read();

if (dataReader.GetInt32(0) <= Convert.ToInt32(Answer) && Convert.ToInt32(Answer) <= dataReader.GetInt32(1) && CheckForNotFake(dataReader.GetString(2), formID))

{ score++; }

if(!CheckForNotFake(dataReader.GetString(2), formID)) { fakeCount++; }

dataReader.Close();

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private bool CheckForNotFake(string character, string formID)

{

try

{

string[] arr = new string[3];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cifCharName " +

"FROM tblCharsInForm " +

"WHERE cifFormID = " + formID + " " +

"ORDER BY cifCharName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (character.Equals(dataReader.GetString(0)))

{ return true; }

}

dataReader.Close();

return false;

}

catch (Exception ex)

{

MessageBox.Show("Select tblQuestionsForChar failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

private void ClearButtons()

{

foreach (RadioButton button in Answer.Controls)

{

button.Visible = false;

button.Checked = false;

}

radioButton11.Checked = true;

}

private void AddButtons(int num)

{

if (num >= 1)

{ radioButton1.Visible = true; }

if (num >= 2)

{ radioButton2.Visible = true; }

if (num >= 3)

{ radioButton3.Visible = true; }

if (num >= 4)

{ radioButton4.Visible = true; }

if (num >= 5)

{ radioButton5.Visible = true; }

if (num >= 6)

{ radioButton6.Visible = true; }

if (num >= 7)

{ radioButton7.Visible = true; }

if (num >= 8)

{ radioButton8.Visible = true; }

if (num >= 9)

{ radioButton9.Visible = true; }

if (num >= 10)

{ radioButton10.Visible = true; }

}

private int NumberOfQuestions()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT MAX(qifOrderNum) FROM tblQuestionsInForm WHERE qifFormID = " + formID + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

return dataReader.GetInt32(0);

}

}

catch (Exception err)

{

MessageBox.Show("tblQuestionsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return -1;

}

public string[] getQuesText(string quesID)

{

try

{

string[] arr = new string[2];

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesText, quesAnswers " +

"FROM tblQuestions " +

"WHERE quesID = " + quesID + " ";

OleDbDataReader dataReader = datacommand.ExecuteReader();

dataReader.Read();

arr[0] = dataReader.GetString(0);

arr[1] = dataReader.GetInt32(1).ToString();

dataReader.Close();

return arr;

}

catch (Exception ex)

{

MessageBox.Show("Select tblForms failed " +

ex.Message, "Errors",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return null;

}

private void nextQuestionClick(object sender, EventArgs e)

{

nextQuesProcess();

}

private void nextQuesProcess()

{

SaveAnswer();

if (index >= numberOfQuestions)

{

nextQuesButton.Visible = false;

ClearButtons();

ques.Visible = false;

Answer.Visible = false;

timerShow.Visible = false;

timer1.Enabled = false;

label1.Visible = false;

finish1.Visible = true;

finish2.Visible = true;

finishScore.Visible = true;

finishScore.Text = GetScore();

return;

}

index++;

Test();

}

private void SaveAnswer()

{

string ans = "-1";

foreach (RadioButton button in Answer.Controls)

{

if (button.Checked)

{

ans = button.Text;

break;

}

}

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string str = string.Format

("INSERT INTO tblAnswersInTest " +

"(aitTestID, aitOrderNum, aitAnswer) " +

" VALUES ( {0}, {1}, {2})",

testNumber, index, ans);

datacommand.CommandText = str;

datacommand.ExecuteNonQuery();

}

catch (Exception err)

{

MessageBox.Show("Insert into tblAnswersInTest failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

}

}

**קוד של טופס עדכון תשובות במבחן:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateAnswersInTest : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

private string answerText;

public FormUpdateAnswersInTest(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

RefreshDataGridView();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormUpdateAnswersInTest\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetAnswersInTest.tblAnswersInTest' table. You can move, or remove it, as needed.

this.tblAnswersInTestTableAdapter.Fill(this.dataSetAnswersInTest.tblAnswersInTest);

}

private void FillComboAnswers()

{

try

{

string[] arr = orderNum.Text.Split('-');

comboAnswers.Items.Clear();

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesText, quesAnswers " +

"FROM tblQuestions " +

"ORDER BY quesText";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetString(0).Equals(arr[1].Substring(1)))

{

for (int i = 1; i <= dataReader.GetInt32(1); i++)

{

comboAnswers.Items.Add(i);

}

break;

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill answers combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetQuestionText(string questionID)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(questionID))

{

str = dataReader.GetString(1);

answerText = str;

return str;

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Get QuestionText failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private string GetTestForm(string testFormID)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT testID, testFormID " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(testFormID))

{

return dataReader.GetInt32(1).ToString();

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("get form test failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private string GetFormName(string FormID)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(FormID))

{

return dataReader.GetString(1);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("get form name failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private string QuestionText(string questionOrderNum)

{

try

{

string str;

string[] arr = testID.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT testID, testFormID " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(arr[0]))

{

str = questionOrderNum + " - " + GetQuestionID(GetTestForm(dataReader.GetInt32(0).ToString()), questionOrderNum);

answerText = GetQuestionID(dataReader.GetInt32(0).ToString(), questionOrderNum);

return str;

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Get QuestionText failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private string GetQuestionID(string testForm, string questionOrderNum)

{

string error = "error";

try

{

string questionText;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT qifFormID, qifOrderNum, qifQuestionID " +

"FROM tblQuestionsInForm " +

"ORDER BY qifOrderNum";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string str = "No Question Exist";

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(testForm) && dataReader.GetInt32(1).ToString().Equals(questionOrderNum))

{

questionText = GetQuestionText(dataReader.GetInt32(2).ToString());

str = questionText;

}

}

dataReader.Close();

answerText = str;

return str;

}

catch (Exception err)

{

MessageBox.Show("Get order num failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return error;

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

testID.Text = dataGridView.SelectedRows[0].Cells[0].Value.ToString() + " - " + GetFormName(GetTestForm(dataGridView.SelectedRows[0].Cells[0].Value.ToString()));

orderNum.Text = QuestionText(dataGridView.SelectedRows[0].Cells[1].Value.ToString());

comboAnswers.Text = dataGridView.SelectedRows[0].Cells[2].Value.ToString();

FillComboAnswers();

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

string[] arr = testID.Text.Split(' ');

string[] arr2 = orderNum.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblAnswersInTest \n" +

"SET aitAnswer = \"" + comboAnswers.Text + "\" \n" +

"WHERE aitOrderNum = " + arr2[0] +

"AND aitTestID = " + arr[0];

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblAnswersInTest ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblAnswersInTest failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblAnswersInTest " +

"ORDER BY aitTestID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblAnswersInTest table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון תכונות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateCharacters : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateCharacters(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

RefreshDataGridView();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormUpdateCharacters\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetCharacters.tblCharacters' table. You can move, or remove it, as needed.

this.tblCharactersTableAdapter.Fill(this.dataSetCharacters.tblCharacters);

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

charName.Text = dataGridView.SelectedRows[0].Cells[0].Value.ToString();

charDesc.Text = dataGridView.SelectedRows[0].Cells[1].Value.ToString();

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblCharacters \n" +

"SET charDesc = \"" + charDesc.Text + "\" \n" +

"WHERE charName = \"" + charName.Text + "\" \n";

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblCharacters ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblCharacters failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblCharacters " +

"ORDER BY charName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblCharacters table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון טפסים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateForms : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateForms(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

FillComboCreatorID();

RefreshDataGridView();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillComboCreatorID()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userFirstName, userLastName " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1) + " " + dataReader.GetString(2);

comboUsers.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill users combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetCreatorName(int num)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userFirstName, userLastName " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(num.Equals(dataReader.GetInt32(0)))

{

return dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1) + " " + dataReader.GetString(2);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill users combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "error";

}

private void FormUpdateForms\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetForms.tblForms' table. You can move, or remove it, as needed.

this.tblFormsTableAdapter.Fill(this.dataSetForms.tblForms);

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

formID.Text = dataGridView.SelectedRows[0].Cells[0].Value.ToString();

formName.Text = dataGridView.SelectedRows[0].Cells[1].Value.ToString();

FormDate.Text = dataGridView.SelectedRows[0].Cells[2].Value.ToString();

comboUsers.Text = GetCreatorName(int.Parse(dataGridView.SelectedRows[0].Cells[3].Value.ToString()));

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

string[] arr = comboUsers.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblForms \n" +

"SET formName = \"" + formName.Text + "\" , \n" +

"formDate = \"" + FormDate.Text + "\" , \n" +

"formCreatorID = \"" + arr[0] + "\" \n" +

"WHERE formID = " + formID.Text;

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblForms ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblForms failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblForms table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון שאלות:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateQuestions : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateQuestions(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

RefreshDataGridView();

FillComboAnswers();

}

private void FillComboAnswers()

{

for (int i = 2; i <= 10; i++)

{

comboAnswers.Items.Add(i);

}

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormUpdateQuestions\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestions.tblQuestions' table. You can move, or remove it, as needed.

this.tblQuestionsTableAdapter.Fill(this.dataSetQuestions.tblQuestions);

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

quesID.Text = dataGridView.SelectedRows[0].Cells[0].Value.ToString();

quesText.Text = dataGridView.SelectedRows[0].Cells[1].Value.ToString();

comboAnswers.Text = dataGridView.SelectedRows[0].Cells[2].Value.ToString();

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblQuestions \n" +

"SET quesText = \"" + quesText.Text + "\" , \n" +

"quesAnswers = \"" + comboAnswers.Text + "\" \n" +

"WHERE quesID = " + quesID.Text;

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblQuestions ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblQuestions failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblUsers table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון שאלות לתכונה:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateQuestionsForChar : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateQuestionsForChar(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

RefreshDataGridView();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FormUpdateQuestionsForChar\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsForChar.tblQuestionsForChar' table. You can move, or remove it, as needed.

this.tblQuestionsForCharTableAdapter.Fill(this.dataSetQuestionsForChar.tblQuestionsForChar);

}

private void FillComboFromValue()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesAnswers " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string[] arr = charOrder.Text.Split(' ');

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(arr[0]))

{

for (int i = 1; i <= dataReader.GetInt32(1); i++)

{

comboFromValue.Items.Add(i);

}

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill from value combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FillComboToValue()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesAnswers " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

string[] arr = charOrder.Text.Split(' ');

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(arr[0]))

{

for (int i = int.Parse(comboFromValue.Text); i <= dataReader.GetInt32(1); i++)

{

comboToValue.Items.Add(i);

}

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill to value combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void FromValueClick(object sender, EventArgs e)

{

comboToValue.Text = "";

comboFromValue.Items.Clear();

if (charOrder.Text != "")

{

FillComboFromValue();

}

}

private void ToValueClick(object sender, EventArgs e)

{

comboToValue.Items.Clear();

if (charOrder.Text != "")

{

FillComboToValue();

}

}

private string GetQuestionText(int num)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(num.Equals(dataReader.GetInt32(0)))

{

return dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "error";

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

charName.Text = dataGridView.SelectedRows[0].Cells[0].Value.ToString();

charOrder.Text = GetQuestionText(int.Parse(dataGridView.SelectedRows[0].Cells[1].Value.ToString()));

comboFromValue.Text = dataGridView.SelectedRows[0].Cells[2].Value.ToString();

comboToValue.Text = dataGridView.SelectedRows[0].Cells[2].Value.ToString();

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

string[] arr = charOrder.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblQuestionsForChar \n" +

"SET qfcFromValue = " + comboFromValue.Text + " , \n" +

"qfcToValue = " + comboToValue.Text + " \n" +

"WHERE qfcCharName = \"" + charName.Text + " \" "+

" AND qfcCharOrder = " + arr[0];

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblQuestionsForChar ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblQuestionsForChar failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsForChar " +

"ORDER BY qfcCharName";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblQuestionsForChar table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון שאלות בטופס:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateQuestionsInForm : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateQuestionsInForm(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

RefreshDataGridView();

FillComboQuestions();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillComboQuestions()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

comboQuestions.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string SearchForQuestion(int num)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT quesID, quesText " +

"FROM tblQuestions " +

"ORDER BY quesID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(num.Equals(dataReader.GetInt32(0)))

{

return (dataReader.GetInt32(0) + " - " + dataReader.GetString(1));

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill questions combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "error";

}

private string GetFormName(string testFormID)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if (dataReader.GetInt32(0).ToString().Equals(testFormID))

{

return dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("get form name failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "Error";

}

private void FormUpdateQuestionsInForm\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetQuestionsInForm.tblQuestionsInForm' table. You can move, or remove it, as needed.

this.tblQuestionsInFormTableAdapter.Fill(this.dataSetQuestionsInForm.tblQuestionsInForm);

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

comboFormID.Text = GetFormName(dataGridView.SelectedRows[0].Cells[0].Value.ToString());

orderNum.Text = dataGridView.SelectedRows[0].Cells[1].Value.ToString();

comboQuestions.Text = SearchForQuestion(int.Parse(dataGridView.SelectedRows[0].Cells[2].Value.ToString()));

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

string[] arr = comboQuestions.Text.Split(' ');

string[] arr2 = comboFormID.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblQuestionsInForm \n" +

"SET qifQuestionID = \"" + arr[0] + "\" \n" +

"WHERE qifFormID = " + arr2[0] +

"AND qifOrderNum = " + orderNum.Text;

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblQuestionsInForm ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblQuestionsInForm failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblQuestionsInForm " +

"ORDER BY qifFormID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblUsers table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון מבחנים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateTests : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateTests(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

RefreshDataGridView();

FillComboForms();

FillComboUsers();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillComboUsers()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userFirstName, userLastName " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1) + " " + dataReader.GetString(2);

comboUsers.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill users combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetUsersName(int num)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT userID, userFirstName, userLastName " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(num.Equals(dataReader.GetInt32(0)))

{

return dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1) + " " + dataReader.GetString(2);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill users combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "error";

}

private void FillComboForms()

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

str = dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

ComboForms.Items.Add(str);

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill forms combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private string GetFormName(int num)

{

try

{

string str;

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT formID, formName " +

"FROM tblForms " +

"ORDER BY formID";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

if(num.Equals(dataReader.GetInt32(0)))

{

return dataReader.GetInt32(0).ToString() + " - " + dataReader.GetString(1);

}

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill forms combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return "error";

}

private void FormUpdateTests\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetTests.tblTests' table. You can move, or remove it, as needed.

this.tblTestsTableAdapter.Fill(this.dataSetTests.tblTests);

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow--;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow++;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridView.Rows[lastRow].Selected = false;

lastRow = dataGridView.Rows.Count - 1;

dataGridView.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

testNumber.Text = dataGridView.SelectedRows[0].Cells[0].Value.ToString();

ComboForms.Text = GetFormName(int.Parse(dataGridView.SelectedRows[0].Cells[1].Value.ToString()));

comboUsers.Text = GetUsersName(int.Parse(dataGridView.SelectedRows[0].Cells[2].Value.ToString()));

testDate.Text = dataGridView.SelectedRows[0].Cells[3].Value.ToString();

dataGridView.CurrentCell = dataGridView[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridView.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

string[] arr = comboUsers.Text.Split(' ');

string[] arr2 = ComboForms.Text.Split(' ');

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblTests \n" +

"SET testFormID = \"" + arr2[0] + "\" , \n" +

"testUserID = \"" + arr[0] + "\" , \n" +

"testDate = \"" + testDate.Text + "\" \n" +

"WHERE testID = " + testNumber.Text;

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridView.CurrentCell = dataGridView[0, lastRow];

MessageBox.Show("Update tblTests ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblTests failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblTests " +

"ORDER BY testID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridView.DataSource = tbl;

dataGridView.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblTests table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridView.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}

**קוד של טופס עדכון משתמשים:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

namespace ReliabilityTest

{

public partial class FormUpdateUsers : Form

{

private OleDbConnection dataConnection;

private bool isManager;

private int lastRow = 0;

public FormUpdateUsers(OleDbConnection dataConnection, bool isManager)

{

WindowState = FormWindowState.Maximized;

InitializeComponent();

this.dataConnection = dataConnection;

this.isManager = isManager;

if (!isManager)

buttonUpdate.Enabled = false;

FillCityCombo();

RefreshDataGridView();

}

private int scrWidth;

private int scrHeight;

private void OnSizeChanged(object sender, EventArgs e)

{

scrWidth = Width;

scrHeight = Height;

panel1.Location = new Point((scrWidth - panel1.Size.Width) / 2, panel1.Location.Y);

}

private void FillCityCombo() // Populate cities combobox

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "SELECT cityName " +

"FROM tblCities " +

"ORDER BY cityName";

OleDbDataReader dataReader = datacommand.ExecuteReader();

while (dataReader.Read())

{

cityUser.Items.Add(dataReader.GetString(0));

}

dataReader.Close();

}

catch (Exception err)

{

MessageBox.Show("Fill cities combobox failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void browseClick(object sender, EventArgs e)

{

DialogResult dlgResult = openFileDialog1.ShowDialog();

string pictureFileName = openFileDialog1.FileName;

pictureBox.ImageLocation = pictureFileName;

pictureUser.Text = pictureFileName;

}

private void FormUpdateUsers\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dataSetUsers.tblUsers' table. You can move, or remove it, as needed.

this.tblUsersTableAdapter.Fill(this.dataSetUsers.tblUsers);

}

private void firstButtonClick(object sender, EventArgs e)

{

dataGridViewUsers.Rows[lastRow].Selected = false;

lastRow = 0;

dataGridViewUsers.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void prevButtonClick(object sender, EventArgs e)

{

dataGridViewUsers.Rows[lastRow].Selected = false;

lastRow--;

dataGridViewUsers.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void nextButtonClick(object sender, EventArgs e)

{

dataGridViewUsers.Rows[lastRow].Selected = false;

lastRow++;

dataGridViewUsers.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void lastButtonClick(object sender, EventArgs e)

{

dataGridViewUsers.Rows[lastRow].Selected = false;

lastRow = dataGridViewUsers.Rows.Count - 1;

dataGridViewUsers.Rows[lastRow].Selected = true;

FillSelectedRow();

}

private void FillSelectedRow()

{

try

{

idUser.Text = dataGridViewUsers.SelectedRows[0].Cells[0].Value.ToString();

firstNameUser.Text = dataGridViewUsers.SelectedRows[0].Cells[1].Value.ToString();

lastNameUser.Text = dataGridViewUsers.SelectedRows[0].Cells[2].Value.ToString();

birthDateUser.Text = dataGridViewUsers.SelectedRows[0].Cells[3].Value.ToString();

addressUser.Text = dataGridViewUsers.SelectedRows[0].Cells[4].Value.ToString();

cityUser.Text = dataGridViewUsers.SelectedRows[0].Cells[5].Value.ToString();

phoneUser.Text = dataGridViewUsers.SelectedRows[0].Cells[6].Value.ToString();

mailUser.Text = dataGridViewUsers.SelectedRows[0].Cells[7].Value.ToString();

passwordUser.Text = dataGridViewUsers.SelectedRows[0].Cells[8].Value.ToString();

isUserManager.Checked = dataGridViewUsers.SelectedRows[0].Cells[9].Value.ToString() == "True";

pictureUser.Text = dataGridViewUsers.SelectedRows[0].Cells[10].Value.ToString();

pictureBox.ImageLocation = pictureUser.Text;

dataGridViewUsers.CurrentCell = dataGridViewUsers[0, lastRow];

EnableButtons();

}

catch (Exception err)

{

MessageBox.Show("Fill Selected Row \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnableButtons()

{

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

if (lastRow == 0)

buttonPrev.Enabled = false;

if (lastRow == dataGridViewUsers.Rows.Count - 1)

buttonNext.Enabled = false;

}

private void ButtonUpdateClick(object sender, EventArgs e)

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

datacommand.CommandText = "UPDATE tblUsers \n" +

"SET userFirstName = \"" + firstNameUser.Text + "\" , \n" +

"userLastName = \"" + lastNameUser.Text + "\" , \n" +

"userBirthDate = \"" + birthDateUser.Text + "\" , \n" +

"userAddress = \"" + addressUser.Text + "\" , \n" +

"userCity = \"" + cityUser.Text + "\" , \n" +

"userPhone = \"" + phoneUser.Text + "\" , \n" +

"userMail = \"" + mailUser.Text + "\" , \n" +

"userPassword = \"" + passwordUser.Text + "\" , \n" +

"userIsAdmin = " + isUserManager.Checked + " , \n" +

"userPicture = \"" + pictureUser.Text + "\" \n" +

"WHERE userID = " + idUser.Text;

datacommand.ExecuteNonQuery();

RefreshDataGridView();

dataGridViewUsers.CurrentCell = dataGridViewUsers[0, lastRow];

MessageBox.Show("Update tblUsers ended successfluly");

}

catch (Exception err)

{

MessageBox.Show("Update tblUsers failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RefreshDataGridView()

{

try

{

OleDbCommand datacommand = new OleDbCommand();

datacommand.Connection = dataConnection;

string sqlCommand = "SELECT \* " +

"FROM tblUsers " +

"ORDER BY userID";

OleDbDataAdapter dataAdapter = new OleDbDataAdapter(sqlCommand, dataConnection);

DataTable tbl = new DataTable();

dataAdapter.Fill(tbl);

dataGridViewUsers.DataSource = tbl;

dataGridViewUsers.AllowUserToAddRows = false;

}

catch (Exception err)

{

MessageBox.Show("Refresh tblUsers table failed \n" + err.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void CellClick(object sender, DataGridViewCellEventArgs e)

{

lastRow = dataGridViewUsers.CurrentRow.Index;

buttonPrev.Enabled = true;

buttonNext.Enabled = true;

FillSelectedRow();

}

}

}