

המטרה בתרגילים האלו, היא להתחיל להרגיש פיתוח של מערכת גדולה ומלאה יותר.
 לכן, אנחנו נתחיל יחד בכיתה בפיתוח של מערכת לניהול מכללה, את השאר תשלימו בבית.
 אנחנו נבנה את זה בשלבים בכמה תרגילי בית, לא הכול היום – נתחיל עם מה שכתוב בהמשך הקובץ, וביחד נעבה את הדרישות מעת לעת, עד שיהיה לנו פרויקט נחמד ללמוד ממנו (יהיו גם שינויים עתידיים ב UI - הדרישות ישתנו).

לא צריך ללמוד Win Forms לעומק, המטרה היא להתחיל ללמד התעסקות עם UI ולהתחיל לחבר יחד UI עם לוגיקה. לכן, תשתדלו לשמור על ממשק משתמש הגיוני, בו ברור למשתמש מה הוא צריך לעשות. אין צורך לייפות מידי, אבל אני ממליץ שכל אחד כפי יכולתו ייפה את המסכים.

טיפ: אנחנו רוצים מערכת שיש לה UX טוב - שמנחה את המשתמש בפעולות שהוא צריך לעשות, אז כנראה "כשניתחנו את המערכת" חשבנו מה הפעולות שצריכות להיות זמינות.

תשקלו אם כדאי לכם להתחיל עם הפרויקט שעשינו בכיתה ולהשלים או להתחיל מחדש (כל אחד מה שנוח לו ונכון לו). לינק לפרוייקט: <https://github.com/efraimt/Hackerme>
 תעבדו רק על מה שכתוב שחובה, ואם הצלחתם ויש זמן ורצון (זה יש בטוח – שווה להשקיע!), תנסו ליישם גם מה שכתוב "לא חובה".

ה"מערכת" תאפשר להוסיף סטודנטים, לשנות להם נתונים, למחוק סטודנטים, להוסיף אותם לקורסים, להוסיף להם ציון לקורס. כל סטודנט יכול להשתתף בכמה קורסים.

נחלק את המסך לאזורים שונים באמצעות פנלים, יהיה לנו כרגע תפריט, ופנל מרכזי.

בתפריט MenuStrip יהיו כפתורים (Students, Courses, Lecturers). כשילחצו על Students יהיה תת תפריט (Actions, List).

ניצור מחלקה **קורס** (מזהה קורס, שם קורס, תאריך התחלה, מס' מפגשים).

ניצור פרוייקט (Class library - DLL) חדש נקרא לו Hackerme.DB, הוא מייצג את מאגר הנתונים שלנו. ב DLL יהיה מחלקה MyDB ובה נשמור את כל הנתונים שלנו במשתנים סטטיים.

יהיה למחלקה MyDB משתנה סטטי StudentList כל סטודנט שיוסיפו במסכים, נוסיף אותו לרשימה הזו.

יהיה למחלקה הזו גם משתנה סטטי CourseList ובתחילת התוכנית נמלא את הרשימה בקורסים (NET Basic, HTML, CSS, Core, OOP).

סטודנט מכיל את המאפיינים: (Id ייחודי, שם פרטי משפחה, תאריך לידה, טלפון, אימייל, עיר מגורים) {אין לשמור?}, הסכום שעליו לשלם, הסכום ששולם, רשימת קורסים שהשתתף בהם).

כשלוחצים על כפתור התפריט Students => Actions ייפתח מסך המציג את פרטי השדות של סטודנט. ראו דוגמה:

בדוגמה הזו שהדבקתי, שמתי רק כמה שדות לדוגמה, אתם שימו את כל השדות.

שימו לב לתפריט למטה, יש לנו כפתורים להוספה, לשמירת שינויים, לחיפוש ע"פ שם, ולמחיקה. תנסו ליישם את כל הפונקציות האלו.

רצוי שתשימו כפתורים רגילים - לא צריך תמונות!

כשתיימו, אפשר להשתמש ב *MenuStrip* ולהביא תמונות יפות מהאינטרנט במקום מילים - כמו בדוגמה שהדבקתי - אבל רק אחרי שתייטמו את הקוד של הוספה ומחיקה ועובד לכם (אז תשאלו ואכוון אתכם במידת האפשר), כרגע הלוגיקה יותר חשובה).

יש הרבה ולידציות שצריך לעשות במערכת, אבל כרגע נקפיד שכשמוסיפים סטודנט או משנים פרטים של סטודנט מולאו שדות החובה (כל השדות שבתמונה שהדבקתי הם חובה), תוודאו שהאימייל תקין!

כשלחצו על כפתור הוספה, אתם במצב "הוספה", נאפס את השדות שבטופס, ונאפשר לשמור את הנתונים עם כפתור השמירה.

לא חובה, מי שיצליח ידפיס הודעה "השמירה הצליחה" ב או בהודעה קופצת (*MessageBox*) או מאתגר יותר ב *StatusStrip* בתחתית ה *Form* (אפשר גם על *Label*).

אם מוצג סטודנט במסך (ע"י חיפוש או סטודנט שהוספנו ושמרנו אותו), אז נאפשר לשלוחות על כפתור "שמירה" ולשמור שינויים שהוקלדו בטופס.

כשמוצג סטודנט, גם כפתור "מחיקה" זמין ללחיצה (יש מאפיין *Enabled* שמאפשר לכפתור לא להיות זמין ללחיצה), וכשלוחצים על מחיקה, נמחק את הסטודנט מרשימת הסטודנטים, ונאפס את השדות שבטופס.

לא חובה, מי שיצליח, יוסיף שאלה "האם אתה בטוח שברצונך למחוק סטודנט?" או "האם אתה בטוח שברצונך למחוק את {שם הסטודנט}?"

לא חובה, מי שיצליח ידפיס בסטטוס בר למטה או בהודעה קופצת "הסטודנט נמחק".

כשלוחצים על כפתור התפריט *List => Students* נחזור ונציג באזור המרכזי רשימת סטודנטים (*dataGridView*).

מזכיר ב WinForms נעבוד באמצעות *UserControls* להחליף את האזור המרכזי.

לא חובה, אבל מי שיצליח יעשה שב *DoubleClick* על שורה נעבור לתצוגת *Actions* של הסטודנט שלחצנו עליו.

מכיוון שאנחנו עדיין לא יודעים לעבוד עם מסדי נתונים, יהיו לנו בנאים או מחלקות עזר
ש"יצרו עבורנו נתונים שנעבוד איתם כשהטופס עולה או ב Main
