פרוייקט בסדנת תכנות ב #C. סמסטר ב' תשפ"א

<u>תיאור הפרוייקט</u>

הפרוייקט יממש שרת gRPC שינהל מערכת של משחקים בין הלקוחות. הלקוחות ישחקו ביניהם משחק ארבעה בשורה (באנגלית four in a row או connect four - ראו למשל http://www.mathplayground.com/connect4.html).

תוכנית הלקוח תמומש עם ממשק משתמש ב WPF.

<u>פרטי הפרוייקט:</u>

- לכל משתמש יש שם משתמש וסיסמה.
- עם הפעלת תוכנית הלקוח ייפתח מסך Login שבו יקליד את שם המשתמש והסיסמה שלו. אם הוא עדין לא משתמש רשום עליו להירשם (register). הפרטים שלו יתווספו לבסיס הנתונים.
- כל client שמתחבר ייכנס למסך ראשי ובו רשימת ההמתנה של כל הלקוחות שמחוברים אך לא משחקים. הוא יוכל לבחור יריב מביניהם.
- ברגע שמשתמש בוחר יריב למשחק, היריב יקבל הודעה והוא יוכל לאשר את ההזמנה או לדחות את ההזמנה לשחק.
- כאשר מתחיל משחק ייפתח לוח משחק אצל כל אחד משני הלקוחות. הראשון לשחק הוא זה שבחר את היריב שלו.
- ה server מנהל בסיס נתונים של המנויים, של כל המשחקים שהסתיימו (מי נגד מי – לפי שם משתמש, מתי המשחק הסתיים, מי ניצח). בנוסף לכך, ה Server יחזיק רשימה של כל המשחקים המתקיימים ברגע זה.
- התוכנית תזהה אם אחד השחקנים ביצע צעד לא חוקי (למשל, ניסיון הוספת עיגול בעמודה שכבר מלאה), מתי המשחק מסתיים, מי ניצח, וכמה נקודות נצברו, ותעדכן בהתאם את בסיס הנתונים.
- כל client שסיים יחזור למסך הראשי ויוחזר לרשימת הממתינים.
- ניהול ה DB צריך להיות ב Entity Framework core ושאילתות צריכות להיות ב LINQ.
- יש להשתמש ב Exceptions handling בכל מקום בו עשויים להיזרק exceptions. יש להיות כמה שיותר ספציפי בתפיסת ה exceptions.

<u>המרכיבים העיקריים של הפרוייקט:</u>

1. תוכנית gRPC Service בשם grpc4InRowService שתנהל את המשחקים ואת בסיס הנתונים
2. תוכנית לקוח ובה לפחות ה Forms הבאים:
   a. Login window
   b. חלון ראשי ובו רשימת השחקנים הממתינים
   c. Game window ובה ה GUI לניהול המשחק.
   d. חלון לחיפושים שונים
3. שתי התוכניות יהיו באותו solution

<u>מימוש המשחק ב GUI:</u>

- לוח המשחק הוא בגודל 7x6
- לחיצה על כל מקום בעמודה מסויימת תגרום ל"ירידה" של עיגול של אותו שחקן באותה שורה. יש לממש אנימציה המראה החלקה של עיגול מלמעלה עד למקום בו הוא נעצר (כמו בקישור למעלה)
- אין להתשמש בכפתורים כדי לקבל קלט מהמשתמש על לוח המשחק.
- התוכנית צריכה להיות פשוטה לשימוש, בלי צורך בהוראות הפעלה.

ניקוד על משחק:

- שחקן שניצח יקבל 1000 נקודות
- שחקן שלא ניצח יקבל את מספר המשבצות שמילא כפול 10. (זה כולל גם את המקרה של תיקו)
- כל שחקן יקבל בונוס של 100 נקודות אם יש לו עיגול בכל עמודה.
- אם אחד השחקנים מפסיק את המשחק לפני שהסתיים (ע"י סגירת לוח המשחק), השחקן השני הוא המנצח.

ה DB:

- בסיס הנתונים צריך להיות ממוקם בתיקייה c:\fourinrow.
- שם בסיס הנתונים צריך להיות fourinrow_(your names).mdf. למשל, fourinrow_brad_angelina.mdf
- ה DB יכלול לפחות שתי טבלאות: Users ו Games.
  - עמודות בטבלת Users: שם משתמש, סיסמה*, מספר המשחקים ששיחק, מספר הנצחונות.
  - עמודות בטבלת Games: קישור לשחקן שהתחיל במשחק, קישור לשחקן השני, קישור לשחקן שניצח (אם המשחק הסתיים בתיקו, אז הקישור הזה יהיה null), תאריך המשחק, מספר המשבצות שמולאו ע"י שני השחקנים ביחד.
- יש להקפיד על קישורים בין הטבלאות.
- סיסמאות לעולם לא נשמרות באופן גלוי. במקום זאת שומרים ערך hash של הסיסמה. עליכם לשמור את ערכי ה hash של הסיסמאות, מחושבים באמצעות אלגוריתם SHA256 (יש להשתמש בספרייה המתאימה). כאשר בודקים אם סיסמה של שחקן נכונה יש לחשב את ה hash של הסיסמה שהמשתמש הכניס ולהשוות עם הערך השמור ב DB.
- ניתן להוסיף טבלאות ל DB או עמודות לטבלאות הנ"ל לפי שיקול דעתכם.
- לצורך הבדיקה, ה DB שאתם מגישים צריך להכיל לפחות את המשתמשים הבאים:

שם משתמש	סיסמה
a1	aa11**11
b2	bb22**22
c3	cc33**33
d4	dd44**44
e5	ee55**55

כמו-כן, ה DB צריך להכיל מידע על לפחות 10 משחקים ששוחקו ע"י המשתמשים הנ"ל

אפשרויות חיפוש:

- הצגת כל המנויים עם אפשרויות מיון לפי שם משתמש, מספר משחקים, מספר נצחונות, מספר הפסדים, מספר הנקודות שצבר.
- רשימת המשחקים ששוחקו עד כה: מי שיחק, מי ניצח ומספר הנקודות שכל שחקן צבר באותו משחק, תאריך המשחק.
- רשימת המשחקים שמתנהלים עכשיו: מי נגד מי ושעת התחלת המשחק.
- הצגת רשימת כל השחקנים שמופיעים במערכת ואפשרויות החיפוש הבאות:
  - הצגת מספר המשחקים, הניצחונות, אחוז הניצחונות ומספר הנקודות שצבר שחקן שהשם שלו ברשימת השחקנים מסומן. (אם לא מסומן שחקן אחד תופיע הודעת שגיאה)
  - הצגת כל המשחקים שהתקיימו בין שני השחקנים שהשמות שלהם מסומנים ואת אחוז הנצחונות של כל אחד משני השחקנים במשחקים שביניהם (אם לא מסומנים שני שחקנים תופיע הודעת שגיאה)

בצמוד לפרוייקט יש לכלול קובץ readme שיכלול:

- שמות ות"ז של המגישים
- אופן חלוקת העבודה בין השותפים.
- תיאור המרכיבים הנוספים שהכנסתם מעבר לדרישות הבסיסיות
- אין לכלול תדפיסי קוד או דיאגרמות.

הנחיות להגשה:

1. יש להגיש תיקייה בשם cs_project מכוווצת המכילה שתי תיקיות:
   a. תיקיית ה solution בשם fourinrow
   b. תיקייה בשם db ובה שני קובצי בסיס הנתונים
   c. קובץ readme
2. העבודה בזוגות בלבד. אם אתם מעדיפים לעבוד לבד, קחו בחשבון שלא תהיה התחשבות מבחינת העומס ותאריך ההגשה.
3. חלוקת עבודה מוצעת: תכנון השרות והשיטות המופיעות בו ייעשה במשותף.
   כתיבת קוד:
   שותף 1: צד השרת
   שותף 2: צד הלקוח, כולל ה GUI.
4. **תאריך הגשה: 1 בספטמבר 2021.**
5. במקרה של צורך בהבהרות (למשל, אם התוכנית לא רצה אצלי) אזמין את הסטודנטים לפגישה בזום.
6. איחור בהגשה: ניתן לאחר בהגשה בשבועיים לכל היותר. במקרה של איחור יורדו נקודות בהתאם (5% על איחור של עד שבוע, 10% על איחור של 8-14 יום).
7. הפרוייקט צריך לעבוד על כל מחשב שמותקן עליו **Visual Studio 2019 community** ללא תקלות ובלי הכנות כמו קונפיגורציות שונות על מחשב המשתמש, העברת קבצים לסיפריות מסויימות על מחשב המשתמש וכדומה (חוץ ממיקום ה DB).

קריטריונים למתן ציון:

1. עמידה בדרישות הפרוייקט.
2. ריצה ללא תקלות – יש לכתוב קוד שיבטיח שתוכנית הלקוח לא תיפול במקרה של התנתקות השרת או השחקן השני.
3. שימוש נכון בשפה ובספריות.
4. שימוש נכון וספציפי ב exceptions – 5% מהציון
5. תיעוד בגוף התוכנית (כולל תיעוד XML) – 5% מהציון
6. עיצוב: אסתטיקה, פשטות ובהירות השימוש (20% מהציון)

אם הדרישות לא ברורות, או שיש לכם שאלות כלשהן על הפרוייקט, נא לכתוב את השאלות ב whatsapp של הקורס כך שהתשובה תהיה גלויה לכל הסטודנטים.

עבודה מהנה!