

EduTrack – מערכת ניהול ציונים אינטרנטית

1. שם הפרויקט

EduTrack – מערכת ניהול ציונים אינטרנטית למוסדות קטנים

2. רקע

2.1 תיאור ורקע כללי

EduTrack היא מערכת מבוססת Web שמיועדת למוסדות לימוד קטנים שאין להם את התקציב לרכישת פתרונות מסחריים. המערכת מאפשרת ניהול ציונים, קורסים, מגמות וסטודנטים בצורה פשוטה ונגישה דרך הדפדפן.

2.2 מטרות המערכת

- התחברות לפי תפקיד (Admin, מרצה, סטודנט)
- ניהול מגמות, קורסים וסטודנטים
- הזנת ציונים ע"י מרצים
- צפייה בציונים ע"י סטודנטים
- הפקת דוחות ציונים לפי מגמה/שנה (SSRS)
- שמירה על אבטחת מידע (הצפנת סיסמאות, בקרת גישה)

3. סקירת מצב קיים בשוק

רוב המוסדות הקטנים משתמשים בפתרונות ידניים כמו אקסל. המערכות המסחריות יקרות או מורכבות מדי. אין פתרון פשוט ונגיש שמתאים במיוחד למוסד קטן.

4. מה הפרויקט אמור לחדש או לשפר

- מערכת פשוטה, זולה ויעילה
- תמיכה בעברית
- הפרדת תפקידים

- גישה מבוססת Web
- הפקת דוחות וניתוחים מותאמים

5. דרישות ופונקציונליות

5.1 דרישות מערכת – הסבר היסטורי וטכנולוגי

המערכת מבוססת על טכנולוגיות של חברת מיקרוסופט, שנמצאות בשימוש נרחב בעולם הפיתוח. כל רכיב במערכת נבחר בהתאם ליתרונותיו המקצועיים, היציבות והאינטגרציה שלו עם שאר המרכיבים. להלן פירוט הדרישות עם הסברים היסטוריים וטכנולוגיים:

שפת #C – שפת תכנות עילית מונחית עצמים

#C (סי-שארפ) היא שפת תכנות מודרנית שפותחה על ידי חברת Microsoft בשנת 2000, כחלק ממסגרת הפיתוח .NET Framework. היא נוצרה בהובלת המתכנת Anders Hejlsberg, שפיתח בעבר גם את Delphi. #C משלבת את העוצמה של שפות כמו ++C עם הפשטות והניקיון של Java, והיא תומכת בתכנות מונחה עצמים (OOP), תכנות אסינכרוני, LINQ, Generics ועוד. השפה מתאימה לפיתוח אתרי אינטרנט, אפליקציות שולחניות, אפליקציות לנייד, מערכות שרת, שירותים בענן ועוד.

למה #C?

- אינטגרציה מלאה עם Visual Studio
- שפה קלה ללמידה, עוצמתית ומסודרת
- תחביר מודרני ותמיכה מלאה באבטחת סוגים

ASP.NET Web Forms – פלטפורמת פיתוח אתרים מבוססת שרת

ASP.NET היא טכנולוגיית פיתוח אתרי אינטרנט שפותחה על ידי Microsoft ויצאה לראשונה בשנת 2002, כחלק מ-.NET Framework. Web Forms הוא אחד המודלים של ASP.NET, והוא מבוסס על **אירועים** (Event Driven) עם תמיכה ב-.Code Behind.

Web Forms מאפשר לפתח ממשק משתמש בצורה גרפית (כמו ב-WinForms) תוך שימוש בקוד #C בשרת.

יתרונות:

- נוח לפיתוח מהיר
- מאפשר הפרדה בין ממשק לקוד
- עיצוב פשוט עם רכיבים גרפיים מוכנים

Microsoft SQL Server – מערכת מסדי נתונים רלציונית

SQL Server הוא מסד נתונים רלציוני שפותח על ידי Microsoft ומלווה את עולם ה-IT מאז **שנות ה-90**. הגרסאות המודרניות כוללות תמיכה מלאה ב-T-SQL, Stored Procedures, טרנזקציות, הצפנה, אינדקסים מתקדמים ויכולות ניתוח נתונים. במערכת SQL Server, EduTrack, ישמש לאחסון המידע של סטודנטים, קורסים, ציונים, משתמשים ומגמות.

מדוע SQL Server?

- אינטגרציה מושלמת עם ASP.NET
- ממשק ניהול נוח (SSMS)
- ביצועים גבוהים וניהול הרשאות מתקדם

SQL Server Reporting Services (SSRS) – הפקת דוחות מקצועיים

SSRS היא מערכת הפקת דוחות שנוספה ל-SQL Server בשנת 2004. היא מאפשרת לבנות דוחות גרפיים וטבלאיים מתוך הנתונים הקיימים במסד הנתונים, ולהציגם ישירות בדפדפן או ב-PDF.

במערכת SSRS, EduTrack תספק דוחות ציונים לפי מגמה, קורס ושנה.

יתרונות:

- שילוב ישיר עם SQL Server
- תמיכה בהפקה ל-Excel ו-PDF
- אפשרות להוספת גרפים, פילוחים וחישובים

Visual Studio – סביבת פיתוח מתקדמת

Visual Studio הושקה לראשונה בשנת 1997 ומאז נחשבת לאחת הסביבות החזקות ביותר לפיתוח מערכות מבית Microsoft. היא תומכת ב-, C#, ASP.NET, JavaScript, SQL ועוד.

יכולות עיקריות:

- עיצוב מסכים (Drop & Drag)
- ניפוי שגיאות (Debugging)
- ניהול גרסאות (Git)
- חיבור למסדי נתונים ול-SSRS

IIS – שרת אינטרנט להרצת אפליקציות

Internet Information Services (IIS) הוא שרת אינטרנט מבית Microsoft שמאפשר להריץ אתרי אינטרנט, בעיקר בטכנולוגיות ASP.NET. הוא קיים מאז שנות ה-90, ומהווה את הברירה המועדפת להרצת אתרים מבוססי .NET.

בפרויקט זה, IIS ישמש כשרת הרצה מקומי או בענן, בהתאם לצורך.

יכולות מרכזיות:

- טיפול בבקשות HTTP
- ניהול הרשאות ואבטחה
- פריסה נוחה ומהירה

5.2 דרישות פונקציונליות

- אדמין: ניהול קורסים, מגמות, משתמשים
- מרצה: הזנת ציונים, צפייה בציוני סטודנטים
- סטודנט: צפייה בציוני האישים
- מערכת Login מאובטחת
- הפקת דוחות מגמה/שנה

6. בעיות צפויות ופתרונות

6.1 בעיות צפויות

- אבטחת מידע
- ניהול הרשאות
- קישור מדויק של מרצים לקורסים
- הצגת דוחות בדפדפן

6.2 פתרונות

- Hash + Salt לסימאות
- מערכת הרשאות מובנית
- טבלאות קשר (מרצה ↔ קורס)
- שילוב SSRS עם הרשאות

7. פתרון טכנולוגי נבחר

7.1 טופולוגיית הפתרון – פרישת המערכת

מערכת EduTrack בנויה כפתרון אינטרנטי במבנה Client-Server. המשתמשים ניגשים למערכת דרך דפדפן, וכל הלוגיקה מתבצעת בצד השרת. הנתונים מאוחסנים במסד נתונים מרכזי, והמערכת כולה רצה על גבי שרת IIS מקומי במהלך הפיתוח.

מרכיבי המערכת:

- ממשק משתמש – דפי Web (Web Forms) המוצגים בדפדפן
- שרת אפליקציה – ASP.NET Web Forms עם קוד #C
- מסד נתונים – SQL Server לניהול טבלאות ציונים, קורסים, משתמשים
- שרת דוחות – SSRS להפקת דוחות מותאמים
- תווך תקשורת – HTTP דרך רשת מקומית או האינטרנט

7.2 טכנולוגיות בשימוש

- ASP.NET Web Forms – לפיתוח אתר אינטרנט אינטראקטיבי בגישת אירועים.
- #C – שפת הפיתוח המרכזית בצד השרת.
- SQL Server – מסד נתונים רלציוני מאובטח.
- SSRS – מערכת ליצירת דוחות מהמסד.
- IIS – שרת אינטרנט להרצת המערכת.
- Visual Studio 2022 – סביבת פיתוח משולבת לכתיבת הקוד וניהול כל המערכת.

7.3 שפות פיתוח

- **#C** – שפה מודרנית מונחית עצמים, נוצרה ע"י Microsoft בשנת 2000. מתאימה לפיתוח לוגיקה עסקית, חישובים, ולידציה והתחברות למסדי נתונים.
- **SQL** – שפה לעבודה עם מסדי נתונים: שליפות, עדכונים, שאילתות.
- **HTML/CSS/JavaScript** – ליצירת ועיצוב הממשק בצד המשתמש.

7.4 תיאור הארכיטקטורה הנבחרת

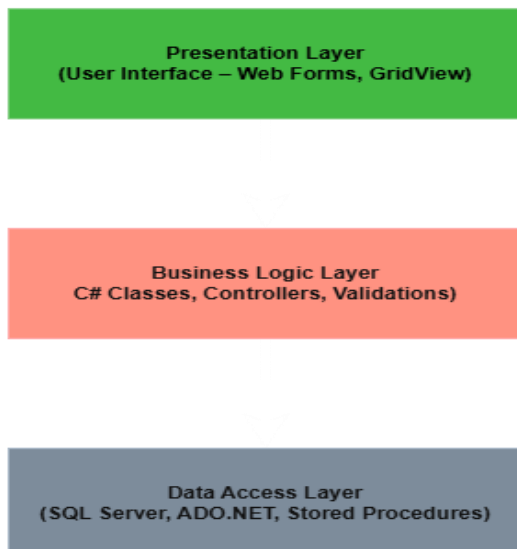
המערכת בנויה לפי מודל Three-Tier Architecture:

1. **Presentation Layer** – הממשק הגרפי בדפדפן.
2. **Business Logic Layer** – קוד #C שמבצע עיבוד נתונים ולוגיקה.
3. **Data Access Layer** – שאילתות SQL והתקשורת עם מסד הנתונים.

גישה זו מאפשרת:

- הפרדה בין תצוגה, לוגיקה ונתונים
- תחזוקה קלה
- אבטחה גבוהה וביצועים טובים

Three-Tier Architecture



7.5 חלוקה לתכניות ומודולים

- **מודול התחברות והרשאות** – ניהול גישה לפי תפקיד (אדמין, מרצה, סטודנט)
- **מודול קורסים ומגמות** – הוספה, עדכון ומחיקה
- **מודול ציונים** – עדכון וצפייה בציונים לפי קורס
- **מודול דוחות** – הפקת דוחות דרך SSRS
- **מודול ניהול משתמשים** – הוספת סטודנטים ומרצים למערכת

7.6 סביבת השרת

- פיתוח ובדיקות יתבצעו על גבי שרת IIS מקומי
- קיימת אפשרות לפרוס את המערכת בעתיד על שרת ענן (Azure) או בשרת מוסדי פנימי

7.7 ממשק המשתמש (GUI)

- הממשק מופיע בדפדפן כתצורת Web
- כולל טפסים להזנה, כפתורים, רשימות נפתחות ו-GridViews
- עיצוב ידידותי למשתמש, מותאם לשפה העברית
- משתנה לפי סוג המשתמש וההרשאות שלו

7.8 ממשקים למערכות אחרות / API

- שילוב עם SSRS להצגת דוחות ציונים בקישור ישיר
- בעתיד ניתן להרחיב את המערכת עם REST API לצורך תקשורת מול מערכות חיצוניות

7.9 שימוש בחבילות תוכנה – הסבר מפורט

במהלך הפיתוח של מערכת EduTrack, נעשה שימוש במספר חבילות תוכנה וספריות עזר (Libraries) שמייעלות את תהליך הפיתוח, משפרות את ביצועי המערכת ומקלות על תחזוקה עתידית. להלן תיאור החבילות המרכזיות בהן נעשה שימוש:

Entity Framework / ADO.NET – גישה למסד הנתונים

- **Entity Framework (EF)** היא ספריית ORM (Object Relational Mapping) של Microsoft, שמאפשרת למפתח לגשת למסד הנתונים דרך אובייקטים בשפת #C, מבלי לכתוב שאילתות SQL גולמיות.
 - **ADO.NET** היא טכנולוגיה בסיסית יותר, שמאפשרת גישה ישירה ל-SQL Server בעזרת פקודות SQL בתוך קוד #C.
- בפרויקט זה ייתכן שימוש באחת מהשיטות, לפי הצורך:
- EF – אם רוצים עבודה ברמה גבוהה, עם פחות קוד.
 - ADO.NET – לשליטה מלאה על השאילתות והביצועים.

יתרונות:

- קיצור זמן פיתוח (EF)
- ביצועים גבוהים ואופטימיזציה (ADO.NET)
- שליטה מלאה בקישור למסד הנתונים

Bootstrap – עיצוב ממשק משתמש

- Bootstrap** היא ספריית CSS ו-JavaScript הפופולרית בעולם, שפותחה ע"י Twitter. היא מספקת רכיבים עיצוביים מוכנים מראש, כמו טפסים, כפתורים, גריד (Grid System), כרטיסים (Cards) ועוד.
- שימוש ב-Bootstrap מאפשר למפתח לבנות ממשק אחיד, מודרני ורספונסיבי (מותאם למובייל), בלי צורך לעצב כל רכיב מאפס.

שימושים אפשריים בפרויקט:

- שיפור מראה טפסים (Forms)
- בניית תפריטים רספונסיביים
- התאמת התצוגה למכשירים ניידים

Git ו-GitHub – ניהול גרסאות

Git הוא כלי לניהול גרסאות קוד (Version Control) שפותח ע"י לינוס טורבאלדס. הוא מאפשר למפתח לשמור גרסאות שונות של הקוד לאורך זמן, ולחזור אחורה במקרה של בעיה.

GitHub הוא שירות ענן לאחסון ושיתוף קוד באמצעות Git.

שימושים בפרויקט:

- שמירת גרסאות לפי שלבי עבודה (Alpha, Final)
- מעקב אחר שינויים ושחזור שגיאות
- גיבוי אוטומטי בקוד בענן

8. מבני נתונים

• 8.1 מבני הנתונים במערכת

- המערכת עושה שימוש בטבלאות רלציוניות המאורגנות במבנה היררכי-קשרי (Relational Model).
- כל טבלה מייצגת ישות במערכת, עם מפתחות ראשיים (Primary Keys) ומפתחות זרים (Foreign Keys) לקשרים ביניהן.

• רשימת הטבלאות המרכזיות:

- **Users** – כל משתמשי המערכת (סטודנט, מרצה, אדמין)
- **Roles** – סוגי הרשאות לכל משתמש
- **Students** – מידע אישי על סטודנטים
- **Teachers** – מידע על מרצים
- **Majors** – מגמות לפי שנה
- **Courses** – רשימת קורסים
- **TeacherCourses** – טבלת קשר בין מרצים לקורסים
- **StudentCourses** – שיוך סטודנט לקורסים
- **Grades** – ציונים בקורסים

- כל טבלה כוללת טיפוסים מתאימים (INT, NVARCHAR, DATE) ומוגנת על ידי הגבלות שלמות (Constraints).

• 8.2 שיטת האחסון

- **סוג האחסון:** המידע נשמר במאגר (Database)
- **טכנולוגיית האחסון:** Microsoft SQL Server
- **צורת האחסון:** טבלאות מקושרות בשיטה תקנית (3 NF – Third Normal Form)
- שיטת אחסון זו מבטיחה:
 - מניעת כפילויות מידע
 - שלמות לוגית בין טבלאות
 - קלות בביצוע חיפושים, פילוחים ודוחות
- המערכת אינה משתמשת בקבצים חיצוניים (כגון XML או JSON), אלא בשאליות SQL בלבד. תיתכן שמירה זמנית של קובצי PDF במערכת הדוחות (SSRS) אך הנתונים עצמם מאוחסנים אך ורק במסד הנתונים.

• 8.3 מנגנוני התאוששות ותמיכה בטרנזקציות

- SQL Server מספק כלים מתקדמים לשמירה על יציבות הנתונים והתאוששות ממצבים של קריסה או תקלה:

• תמיכה ב־Transactions

- כל פעולת כתיבה / עדכון קריטית תתבצע בתוך TRANSACTION
- במקרה של כשל – תבוצע פקודת ROLLBACK
- מאפשר שמירה על עקביות הנתונים

• מנגנוני גיבוי

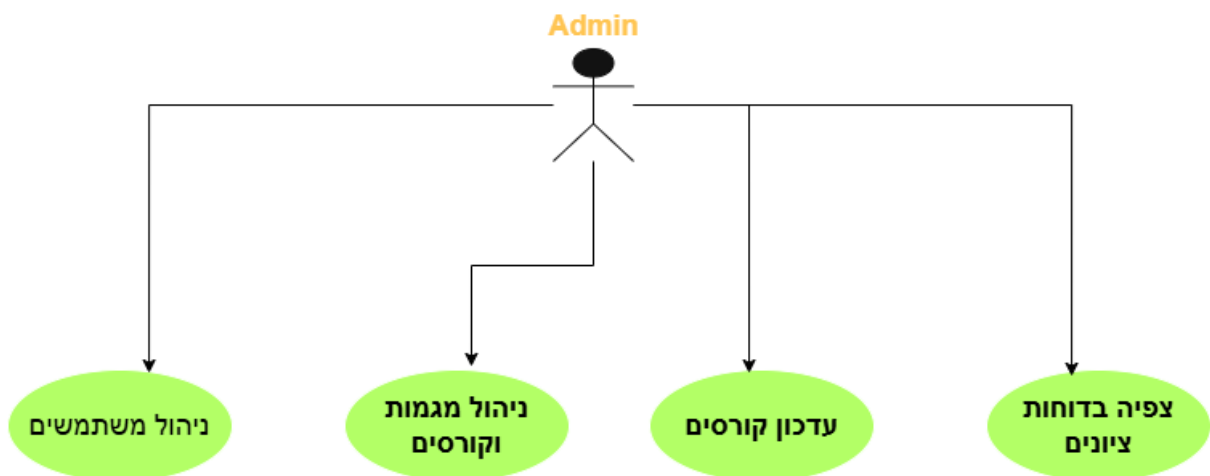
- גיבוי אוטומטי של בסיס הנתונים (Full Backup, Differential, Log Backup)
- שחזור בסיס נתונים מלא במקרה של קריסה
- ניתן לבצע גיבויים ידניים או מתוזמנים דרך SSMS

- **שחזור אוטומטי לאחר כשל (Crash Recovery)**
- SQL Server כולל מנגנון התאוששות אוטומטי עם שימוש ב-Transaction Log
- בעת הפעלה מחדש, מתבצעת קריאת הלוגים האחרונים ושחזור הפעולות שלא נשמרו
- **ניהול שגיאות בלוגיקה**
- טיפול בשגיאות באמצעות Try-Catch ב-C#, תיעוד שגיאות לקובץ לוג או למסד

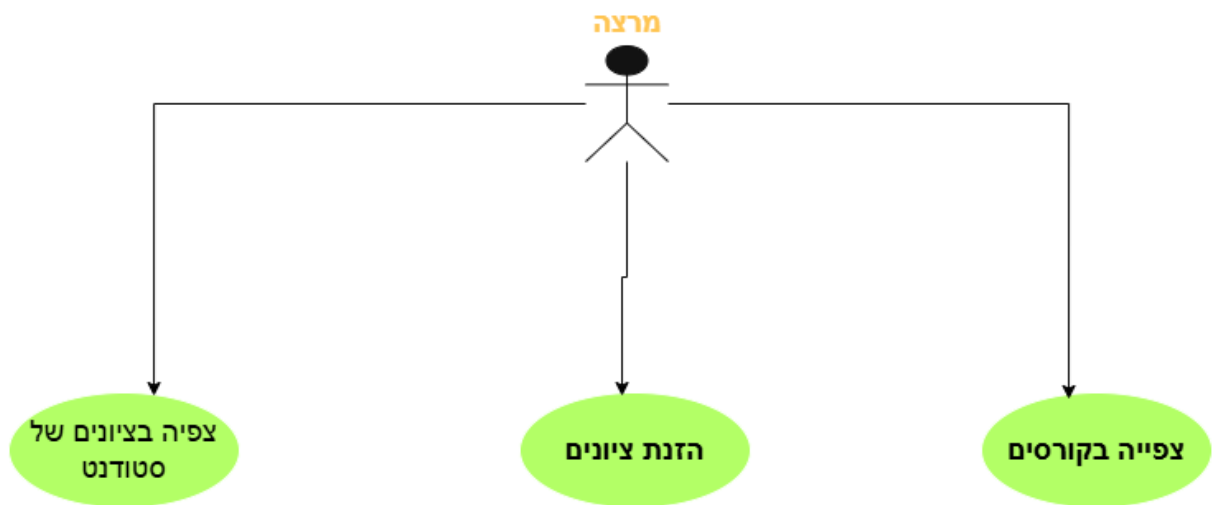
9. תרחישים מרכזיים

Use Case Diagrams 9.1

Admin: ניהול קורסים, משתמשים, צפייה בדוחות

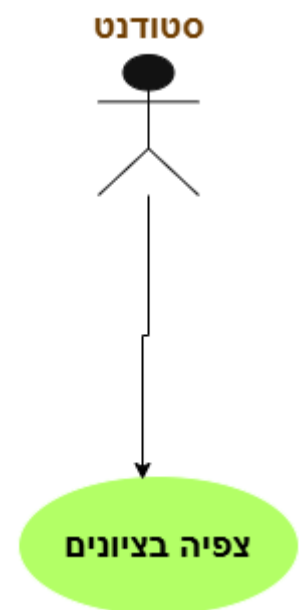


מרצה: הזנת ציונים, צפייה בקורסים



סטודנט: צפייה בציוניו

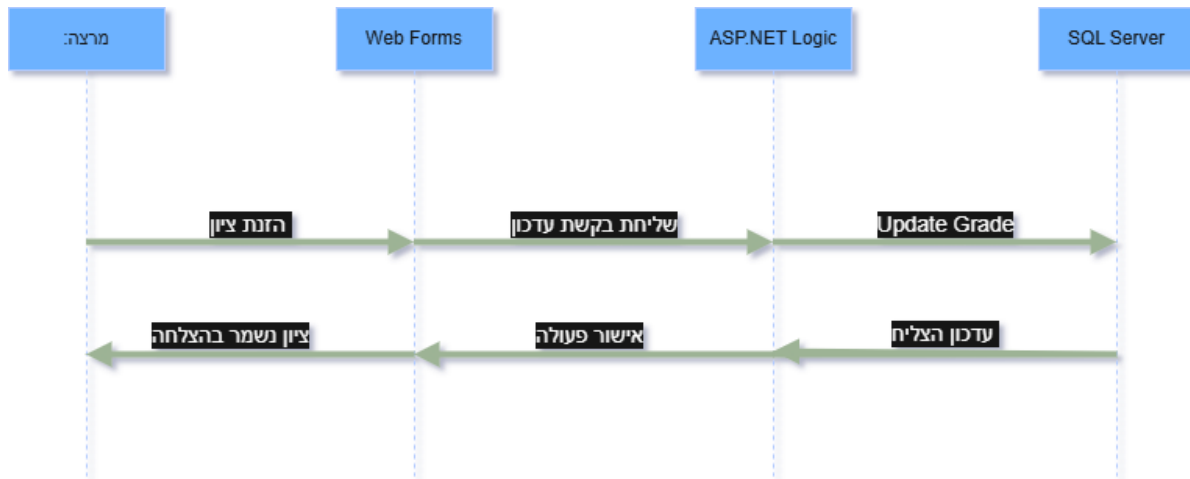
: ניהול קורסים, משתמשים, צפייה בדוחות



Sequence Diagrams 9.2

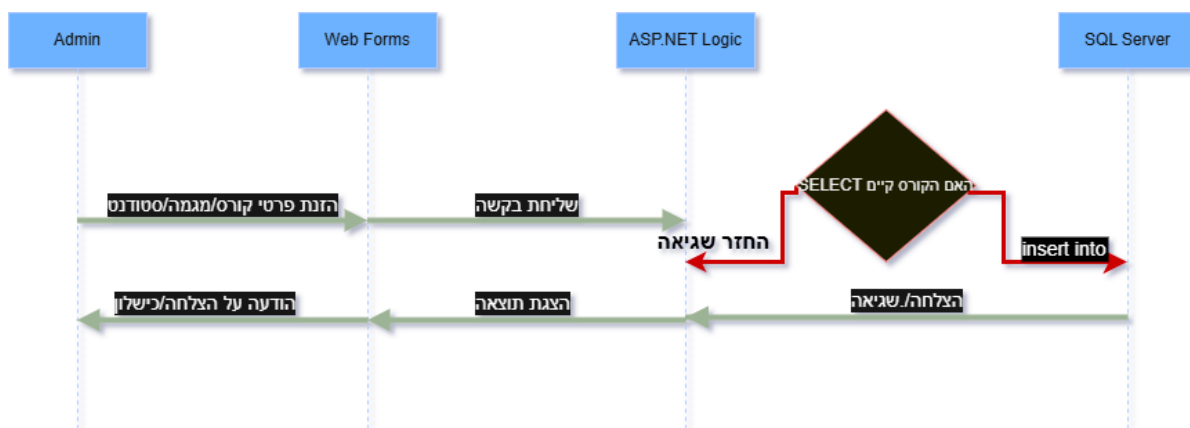
תרשים 1: מרצה מעדכן ציון

שליחה → ולידציה → עדכון במסד → הצגת אישור



תרשים 2: Admin מוסיף קורס חדש

בדיקת קיום INSERT → (SELECT) אם לא קיים



תרשים 3: Admin יוצר משתמש חדש

בדיקת קיום → יצירת משתמש חדש במסד

תרשים 4: Admin משייך מרצה לקורס

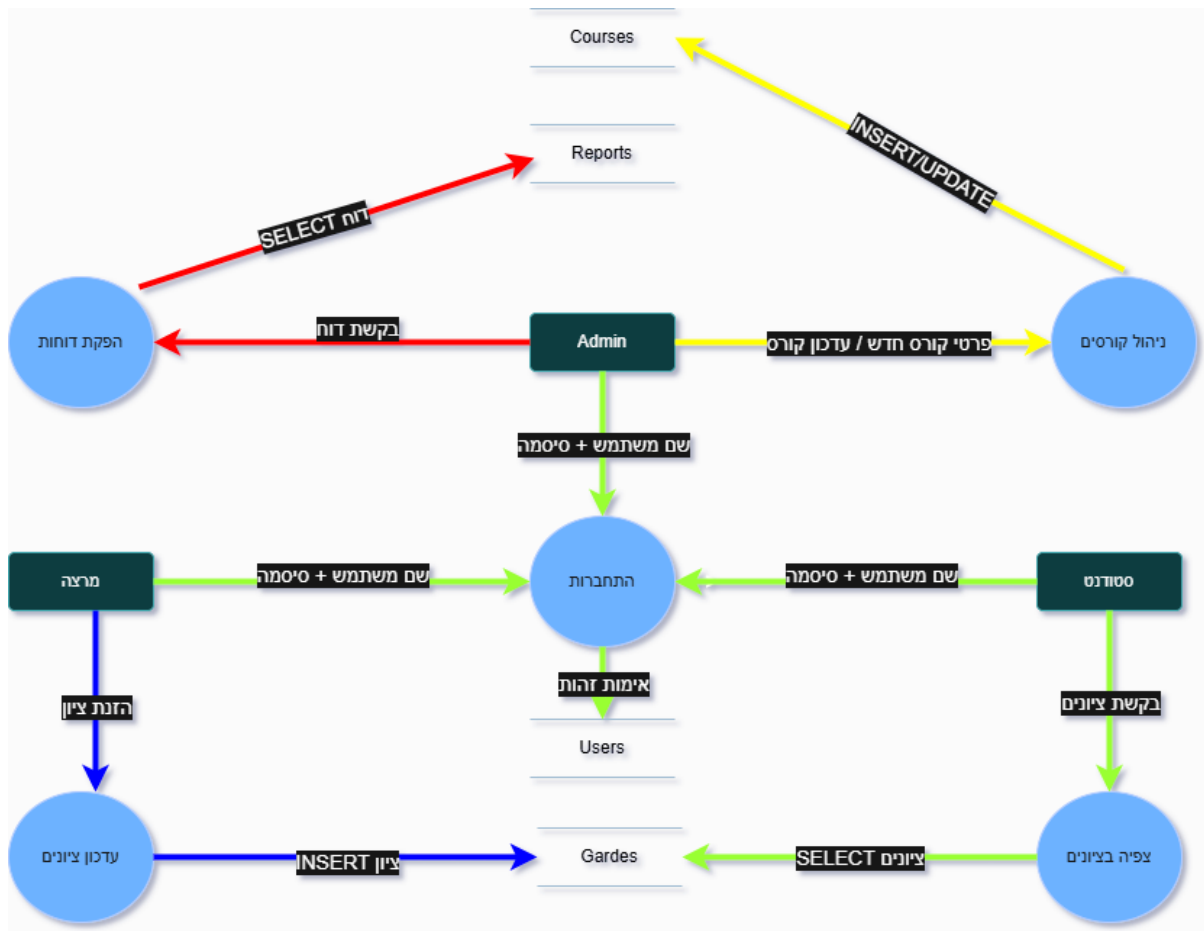
הזנת פרטים → שיוך בטבלת קשר

תרשים 5: סטודנט צופה בציוניו

כניסה → בקשה ל-Grades → הצגת תוצאה

Data Flow Diagram (DFD) 9.3

שחקנים: Admin, מרצה, סטודנט
תהליכים: התחברות, ניהול קורסים, צפייה/עדכון ציונים, הפקת דוחות
מאגרי מידע: Users, Courses, Grades, Reports
(יש לצרף תרשים DFD)



10. מרכיב אלגוריתמי

- חישוב ממוצע ציונים לכל סטודנט
- סינון לפי מגמות/שנים
- סטטיסטיקות ציונים: כישלונות, הצטיינות

11. אבטחת מידע

- הצפנת סיסמאות
- הרשאות לפי תפקיד
- סשן בטוח
- סינון קלטים
- הגנה מ-SQL Injection

12. משאבים נדרשים

- סטודנט אחד
- Visual Studio, SQL Server
- ידע ב-SSRS, ASP.NET
- מדריכים: Microsoft Docs, Stack Overflow

13. תכנית עבודה

- שבוע 1 – אפיון
- שבוע 2 – מסד נתונים
- שבוע 3 – 5 - פיתוח
- שבוע 6 – דוחות
- שבוע 7 – בדיקות
- שבוע 8 – מסירה + מצגת

14. תכנון בדיקות

מס'	נושא הבדיקה	מטרת הבדיקה	תרחישים לבדיקה
1	Login לפי תפקיד	לוודא שכל משתמש מקבל גישה לפי תפקידו וההרשאות	התחברות עם פרטים תקינים / שגויים, משתמש חסום, תפקיד לא נכון
2	הזנת ציונים	לוודא שמרצה יכול להזין ציונים תקפים לקורסים שלו	ציונים תקינים, טווח לא תקני, קלט טקסט, ניסיון לקורס לא מוקצה
3	הפקת דוח לפי מגמה	לוודא שדוחות SSRS מופקים בהתאם למגמה ולשנה	הפקת דוח עם נתונים, ללא נתונים, מגמה שאינה קיימת
4	בדיקות שדה/טופס	לוודא תקינות נתונים קלטים בטפסים	לוודא תקינות נתונים קלטים בטפסים
5	שאליות למסד הנתונים	לוודא שהמערכת מבצעת שאליות תקינות ומדויקות	שליפות לפי סטודנט, קורס, ממוצע וציונים, קשרים Join תקינים

15. בקרת גרסאות

- Git + GitHub
- Commit עם תיעוד
- שימוש ב-Branches
- יצירת גרסאות Alpha, Final