טכנולוגיות בשימוש :

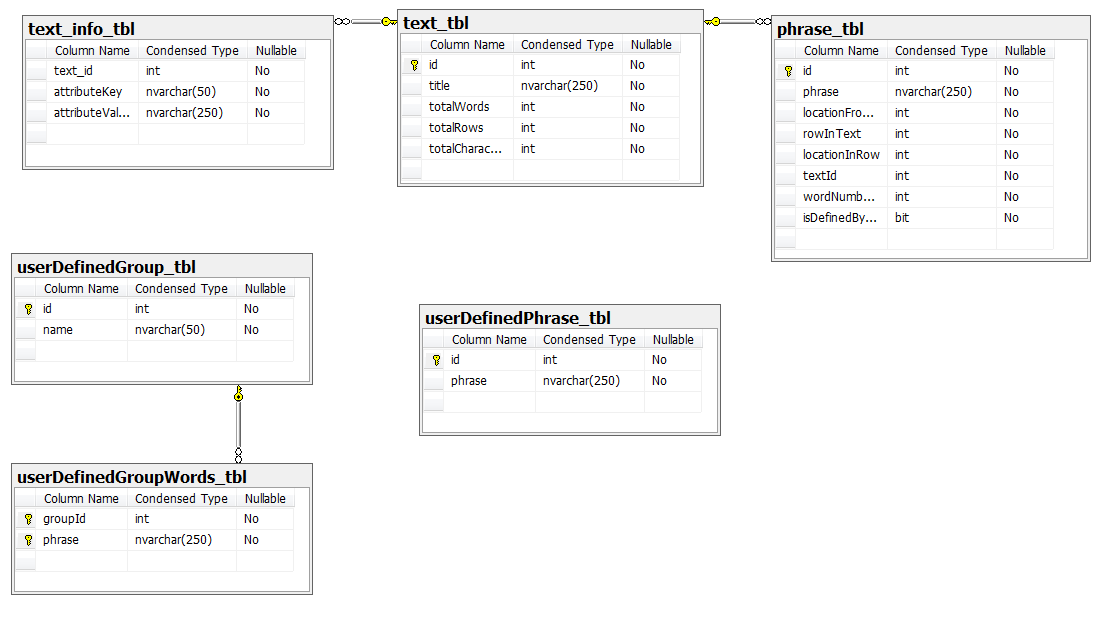
Microsoft SQL Server , Nodejs, Angular8

הסבר כללי :

המערכת נבנתה על פי דרישות הסדנה, מקבלת קבצי טקסט (מאמרים וספרים) בכל גודל ומכניסה אותם לבסיס הנתונים שורה אחרי שורה (ניתן גם להכניס טקסט בשלמותו אך לא יעיל עבור טקסטים גדולים) תוך כדי מיפוי ואיסוף סטטיסטיקות על הביטויים, המערכת מזהה ביטוי לפי הרווחים המפרידים, כשהחריג לכלל זה הם הביטויים המוגדרים ע"י המשתמש. המערכת מאפשרת חיפוש ביטוי , הצגת סטטיסטיקות , יצוא ויבוא מXML , הגדרת ביטויים וקבוצות ביטויים ע"י המשתמש , חיפוש לפי מיקום מתחילת הטקסט, חיפוש על פי שורה ומיקום בשורה.  
המערכת חושפת שורה של פרוצדורות הנקראות מהממשק ,כך גם נשמרים העקרונות של עיצוב מונחה אובייקטים וגם מודל השכבות וכן המידע מאובטח ע"י Parameter sandboxing ונקודות גישה מעטות וייעודיות ע"י המשתמשים.  
המערכת מציגה למשתמש שורה של מסכים מתוך תפריט ובהם נחשפת הפונקציונליות למשתמש

הגדרת הDB :

דיאגרמת כלל הטבלאות :



פירוט הטבלאות:

1. טבלת הביטויים

מפתח ראשי: ID

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Phrase | LocationFromStart | RowInText | LocationInRow | TextID | WordNumberInText | isDefinedByUser |
| Int identuty | nvarchar | int | int | int | Int FK | int | bit |

1. טבלת הטקסטים

מפתח ראשי: ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | title | totalWords | totalRows | TotalCharacters |
| Int Identity | nvarchar | int | int | int |

1. טבלת המידע על הטקסטים

מפתח ראשי: TextID + AttributeKey

|  |  |  |
| --- | --- | --- |
| TextID | AttributeKey | AttributeValue |
| Int FK | nvarchar | nvarchar |

1. טבלת הביטויים המוגדרים אישית:

מפתח ראשי: ID

|  |  |
| --- | --- |
| ID | Phrase |
| Int identuty | nvarchar |

1. טבלת קבוצות מוגדרות אישית:

מפתח ראשי: ID

|  |  |
| --- | --- |
| ID | Name |
| Int Identity | nvarchar |

1. טבלת המילים בקבוצה מוגדרת אישית:

מפתח ראשי: GroupID + Phrase

|  |  |
| --- | --- |
| GroupID | Phrase |
| Int FK | nvarchar |

פירוט האינדקסים (למעט המפתחות הראשיים שכבר פורטו) :

אינדקס על הביטויים בטבלת הביטויים לחיפוש של ביטויים (קלסטר אינדקס)

אינדקס על, TextID ,WordNumberInText לאחזור מהיר של רצפים של טקסט

פרוצדורות חשופות החוצה:

AddNewText\_sp – מקבלת טקסט וכותרת ומוסיפה את הטקסט לטבלת הטקסטים ומאנדקסת אותו לטבלת הביטויים תוך כדי חישוב הסטטיסטיקות שלו ומחזירה מזהה לרשומה שנוספה

AddTextMetaData\_sp - מקבלת מזהה טקסט, key ו value ומכניסה לטבלה המתאימה

GetTextMetaDataAndStats\_sp - מקבלת מזהה טקסט ומחזירה את כל הפרטים שלו (כולל סטטיסטיקות)

FindTextByMetaData\_sp מקבלת key ו value ומחזירה טבלה המכילה את כל מזהי הטקסטים שיש בהם את הנתון הזה

GetIndexForPhrase\_sp – מקבלת ביטוי ואופציונלית מזהה של טקסט ומוצאת ומחזירה את כל המופעים של הביטוי בכלל או בטקסט ספציפי שצוין

GetUserDefinedGroupIndex\_sp – מקבלת מזהה קבוצה ואופציונלית מזהה של טקסט ומוצאת ומחזירה את כל המופעים של ביטויי הקבוצה בכלל או בטקסט ספציפי שצוין

GetContextForPhrase\_sp מקבל מזהה ביטוי וטווח של טקסט מביא את הטקסט שמקיף את הביטוי הספציפי עבור התצוגה לפי הטווח שצוין

GetContextForPhraseAllAppearance\_sp מביא את הטקסט שמקיף את כל מופעי הביטוי עבור התצוגה

GetTextByID\_sp מקבלתID של טקסט ומחזירה את הטקסט המתאים

AddUserDefinedPhrase\_sp מקבלת ביטוי חדש ואם הוא לא קיים עדיין אז מוצאת את כל המופעים שלו בטקסטים הקיימים ומכניסה את הרפרנסים לטבלת הביטויים ואת הביטוי עצמו לטבלת הביטויים המוגדרים אישית

DeleteUserDefinedPhrase\_sp מקבלת ID של ביטוי מוגדר ומוחקת אותו ואת המופעים שלו מהDB

GetPhraseByID\_sp מקבל ID של ביטוי ומביא את רשומת הביטוי (מופע ספציפי)

DeleteText\_sp מקבלת מזהה של טקסט ומוחקת אותו ואת כל הציטוטים שלו מהDB

CreateUserDefinedWordsGroup\_sp מקבלת שם קבוצה ומגדירה אותה, מחזירה מזהה קבוצה

AddPhraseToUserDefinedGroup\_sp מקבל ביטוי ומזהה של קבוצה ומוסיף אותו לקבוצה

AddPhraseListToUserDefinedGroup\_sp מקבל מחרוזת של ביטויים ואת התו המפריד בניהם ומזהה של קבוצה ומוסיף אותם לקבוצה

DeleteUserDefinedWordFromGroup\_sp מקבל ביטוי ומזהה של קבוצה ומוחק אותו מהקבוצה

DeleteUserDefinedWordsGroup\_sp מקבל מזהה של קבוצה ומוחק אותה

GetPhraseByRowLocation\_sp מקבל מספר שורה ,מספר מילה בשורה ומזהה טקסט ומחזיר את המילה המבוקשת , יחזיר NULL אם מפנה למיקום שלא קיים בטקסט

GetAllPhrasesInAText\_sp מקבל מזהה של טקסט ומחזיר את כל הביטויים שבו (כולל המוגדרים אישית)

GetStatsForPhrase\_sp מקבל מילה ומחזיר את מספר המופעים הכולל שלה ובכמה טקסטים שונים היא מופיעה

GetStatsForRow\_sp מקבל מזהה טקסט ומספר שורה ומחזיר את מספר המילים והתווים בשורה (התווים כולל רווחים)

GetFrequencyList\_sp מחזיר את כל המילים כשעבור כל אחת מהן מצוין מספר המופעים שלה בDB

GetFrequencyListForText\_sp מחזיר את כל המילים בטקסט ספציפי כשעבור כל אחת מהן מצוין מספר המופעים שלה בDB

SearchPhrase\_sp מקבל ביטוי ומחזיר את כל המופעים שלו שנמצאו ניתן להגביל את החיפוש לטקסט ספציפי ע"י העברה של פרמטר אופציונאלי

ExportIngestedTexts\_sp מייצא לXML את הטקסטים השמורים בDB

ExportUserDefinedGroups\_sp מייצא לXML את קבוצות המילים השמורות של המשתמשים

ExportUserDefinedPhrases\_sp מייצא לXML את הביטויים מוגדרים אישית השמורים בDB

GetImportPathesForXML\_sp מחזירה את הנתיבים המוגדרים לטעינת נתונים מתוך קבצי הXML לDB

ImportIngestedTexts\_sp מייבא טקסטים לDB מתוך XML

ImportIngestedTextsFromPath\_sp מייבא טקסטים לDB מתוך קובץ XML שנשמר בנתיב (אופציה עבור קבצים גדולים במיוחד)

ImportUserDefinedGroups\_sp מייבא קבוצות מילים שמורות של משתמשים לDB מתוך XML

ImportUserDefinedGroupsFromPath\_sp מייבא קבוצות מילים שמורות של משתמשים לDB מתוך קובץ XML שנשמר בנתיב (אופציה עבור קבצים גדולים במיוחד)

ImportUserDefinedPhrases\_sp מייבא ביטויים מוגדרים אישית לDB מתוך XML

ImportUserDefinedPhrasesFromPath\_sp מייבא ביטויים מוגדרים אישית לDB מתוך קובץ XML שנשמר בנתיב (אופציה עבור קבצים גדולים במיוחד)

IngestNewTextMultyLevelProcess\_StepOne\_CreateTextEntity\_sp שלב ראשון בתהליך טעינת טקסט שורה אחר שורה, פרוצדורה זו נקראת פעם אחת כדי ליצור את מסגרת הטקסט ולהקצות לו ID במערכת

IngestNewTextMultyLevelProcess\_StepTwo\_AddNewRow\_sp שלב שני בתהליך טעינת טקסט שורה אחר שורה, פרוצדורה זו נקראת עבור כל שורה בטקסט ומכניסה אותה לDB

IngestNewTextMultyLevelProcess\_StepThree\_ApplyUDP\_sp – שלב אחרון בתהליך טעינת טקסט שורה אחר שורה, פרוצדורה זו נקראת פעם אחת כדי להחיל את הטקסט החדש את המיפוי לביטויים המוגדרים אישית של המשתמשים

GetUserDefinedGroupInfo\_sp מקבל מזהה קבוצה ומחזיר את כל הפרטים שלה ואת כל המילים שבה

GetUserDefinedGroupsList\_sp מחזיר רשימה של כל הקבוצות

פונקציות פנימיות :

GetNextPhraseInText\_f מביאה את הביטוי הבא בטקסט

GetTextByIndexs\_f מחזירה בין שני אינדקסים (למשל את הטקסט בין המילה השלישית למילה הארבעים)

GetTextByIndexsNonRecurcive\_f מביאה את הביטוי הבא בטקסט ללא שימוש ברקורסיה (כך שנמנעת המגבלה על מספר תת הקריאות שקיימת בDB )

GetWordCount\_f מקבלת מילה ומחזריה את מספר התווים שבה

דוגמאות לשאילתות שיעשה בהן שימוש :

ALTER PROCEDURE [dbo].[GetPhraseByRowLocation\_sp]

@rownum INT,

@wordinrow INT,

@textid INT

AS

BEGIN

SET NOCOUNT ON;

SELECT [id] ,[phrase] ,[locationFromStart] ,[rowInText],[locationInRow] ,[textId] ,[wordNumberInText]

FROM [dbo].[phrase\_tbl]

WHERE textId = @textid

AND rowInText = @rownum

AND locationInRow = @wordinrow

END

ALTER PROCEDURE [dbo].[IngestNewTextMultyLevelProcess\_StepTwo\_AddNewRow\_sp]

@row NVARCHAR(max),

@textid INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @runningtext NVARCHAR(max) = RTRIM(LTRIM(@row));

DECLARE @currentwordinrow INT = 0;

DECLARE @currentrow INT;

DECLARE @str NVARCHAR(250);

DECLARE @currentwordintext INT;

DECLARE @currentcharacterintext INT;

DECLARE @temp\_phrase\_tbl TABLE([phrase] NVARCHAR(250), [locationFromStart] INT, [rowInText] INT, [locationInRow] INT,

[wordNumberInText] INT)

SELECT @currentwordintext = isnull([wordNumberInText],0),

@currentcharacterintext = isnull(([locationFromStart] + LEN(phrase)), 0),

@currentrow = isnull([rowInText],0)

FROM [dbo].[phrase\_tbl]

WHERE textId = @textid

AND [wordNumberInText] = (SELECT MAX([wordNumberInText])

FROM [dbo].[phrase\_tbl]

WHERE textId = @textid)

WHILE(@runningtext != '')

BEGIN

-- handle next word

SET @str = [dbo].[GetNextPhraseInText\_f](@runningtext);

SET @currentwordinrow = (@currentwordinrow + 1);

SET @currentwordintext = (@currentwordintext + 1);

--PRINT '@str :' + @str + ' num:' + CAST(@currentwordscounter AS NVARCHAR) + 'SUBSTRING(@runningtext, 1, 3):'+

--SUBSTRING(@runningtext, 1, 3);

INSERT INTO @temp\_phrase\_tbl([phrase], [locationFromStart], [rowInText], [locationInRow], [wordNumberInText])

VALUES(@str, @currentcharacterintext, @currentrow, @currentwordinrow, @currentwordintext)

SET @currentcharacterintext = (@currentcharacterintext + LEN(@str));

--PRINT ' LEN(@runningtext):' + CAST(LEN(@runningtext) AS NVARCHAR) + ' '+ ' LEN(@str):' + CAST(LEN(@str) AS NVARCHAR);

-- prepering the running text for next pass

SET @runningtext = RTRIM(LTRIM(@runningtext));

SET @runningtext = RIGHT(@runningtext,LEN(@runningtext) - LEN(@str));

SET @runningtext = RTRIM(LTRIM(@runningtext));

END

INSERT INTO [dbo].[phrase\_tbl]([phrase], [locationFromStart], [rowInText], [locationInRow], [textId], [wordNumberInText], [isDefinedByUser])

SELECT [phrase], [locationFromStart], [rowInText], [locationInRow], @textid, [wordNumberInText], 0

FROM @temp\_phrase\_tbl;

END

מסכים מממשק המשתמש :



