בס"ד

שאלת בונוס

שאלה מספר 2 :

**תיאור הארכיטקטורה הנבחרת:**

הארכיטקטורה הנבחרת היא חלוקה ל-3 שכבות -3Architecture Tier- .

בפרויקטים שמבוססים על Database מומלץ להשתמש בגישה של מודל השכבות ,

בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט.

היתרון העיקרי בחלוקה לשכבות הוא תחזוקה קלה, במיוחד במערכות גדולות, כך שניתן יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים בתוכנית.

בנוסף לגישה זו הוספנו את ספריית ה DTO(Entities) שאפשרה לנו בידוד מוחלט וחוסר השענות על טכנולוגיה מסוימת.

חלוקה לתוכניות ומודולים:

1.שכבת התצוגה

– (resentation LayerP) שכבת ממשק המשתמש UI- ) Interface User)

שכבה זו מורכבת מפרויקט JavaScript המציג את דפי HTML השכבה מתקשרת בין המשתמש לבין שאר השכבות של המערכת.

.2שכבת ה – WebApi שכבה זו מספקת את התקשורת בין הלקוח לשרת, היא תקבל קריאות מ, JavaScript - תיגש לפונקציות בשכבות הבאות ותחזיר נתונים משוטחים בחזרה . JavaScript

3.שכבת הDTO (Entities) –

שכבה זו מגדירה את האובייקטים איתם שכבת WebApi תתעסק.

ובכך אנו מבודדים לחלוטין את ההישענות על טכנולוגיה מסוימת.

.4שכבת הלוגיקה העסקית– (BL - Business Logic) השכבה שאמורה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה. בשכבה זו נממש את הפונקציונאליות של המערכת

.**5שכבת הנתונים– (DAL- Data Access Layer)** שכבה זו מורכבת ממקור לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

**שכבת הWebApi**

שכבה זו מספקת את התקשורת בין הלקוח לשרת,

היא תקבל קריאות מהקליינט ,(JavaScript)

תיגש לפונקציות בשכבות הבאות ותחזיר נתונים משוטחים חזרה . JavaScript

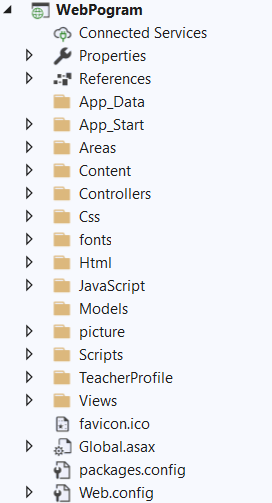
**קבצי פרויקט זה :**

תיקיית הStart\_App , מכילה קבצים מובנים,

לפיכך לא פירטנו את תכולתה.

תיקיית Controllers מכילה את קבצי ה controller- המתקשרים בין הלקוח לשרת.

הקובץ Global.asax מכיל את הפונקציות הראשיות של השרת -הרשאות וכד', וכן קריאה לפונקציות שמורצות מיד עם עליית השרת.



**השרת-**

התוכנית נבנתה בצורת חשיבה של – DNA מודל שלושת השכבות .

התוכנית חולקה לשלוש שכבות כאשר כל שכבה מטפלת בתחום שונה ומכילה כמה מחלקות.

בתוך כל מחלקה הוגדרו משתנים ואובייקטים מתאימים המיוחדים למחלקה.

כל מחלקה מטפלת בתחום מסוים.

צורת כתיבה זו נועדה להקל על המתכנת הן בכתיבת הקוד והן בתחזוקה

השוטפת של הקוד.

כמובן שבין המחלקות שבשכבות השונות קיים קשר, וקימת התייחסות מתוך

מחלקה אחת לעצמים המוגדרים במחלקות אחרות אך הכל נעשה תוך שמירה

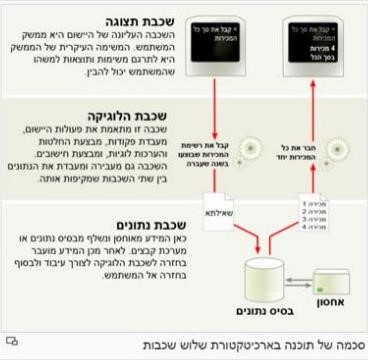
על בטחון והבטחת הנתונים והפונקציות.

**השרת מורכב מ-3 פרויקטים:** •

**שכבת**- **Dto** ממירה אובייקטים מסוג table sql ל #C וכן להפך.

**• שכבת Bll** -שכבה זו מטפלת בכל הפונקציות הקשורות לתכונה עצמה.

• **שכבת**- **Dal** שכבה המטפלת בקישור התוכנה עם בסיס הנתונים.



**שכבת Dto (Entities)**

שכבה זו מגדירה את האובייקטים איתם שכבת ה- WebApi תתעסק.

ובכך אנו מבודדים לחלוטין את ההישענות על טכנולוגיה מסוימת.

המחלקות מבטאות המרה של מסד הנתונים למחלקות שונות

בכל מחלקה קיימות 2 פונקציות בסיס שתפקידן המרת נתונים מה db לאובייקטים מסוג מחלקות ולהפך.

**שכבת הלוגיקה העסקית**

**(BLL- Business Logic Layer)**

השכבה שממונה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה.

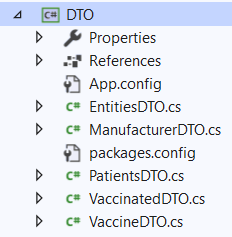
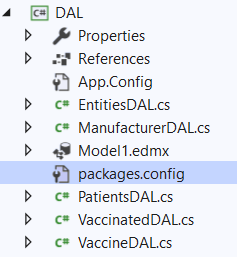
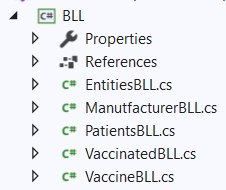
בשכבה זו נממש את הפונקציונאליות של המערכת.

המחלקות השונות מממשות פונקציות של עיבוד ושליחת נתונים

מהשרת למסד הנתונים ומהשרת למסד הנתונים.

שכבת הנתוניםDAL- Data Access Layer) )

שכבה זו מורכבת ממקור נתונים –מסד הנתונים שלנו, וממערכת תוכנה Entity Framework אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים פרויקט זה מכיל את קבצי ה Entity Frameworkהמחברים את המסד.

.

**תצוגה סכמתית של המידע בdatabase:**

