שמירה ישירה ל-DB ללא שליפת נתונים:

1. נתקלתי במצב שבו פונקציות קיבלו ערכים מבלי לשלוף אותם קודם מה DB שהוביל לבעיה בניסיון לשמור נתונים

 **פתרון:** דאגתי לשלוף את הנתונים מה DB בהתחלה כדי להבטיח שהשינויים יישמרו בצורה תקינה.

תהליכים מרובי THREAD

2. כאשר מספר תהליכים (threads) רצו במקביל ולא לפי הסדר הנדרש, נגרמו בעיות תיאום.

פתרון: הפרדתי את התהליכים בצורה כזו שכל פונקציה תתמקד בתהליך אחד בלבד, מה ששיפר את הסדר והיציבות.

שגיאת סירקולציה (Circular Dependency):

3. בעת הזרקת שירותים (services) לקונטרולרים באמצעות Dependency Injection, נוצרה רקורסיה אינסופית שהובילה לקריסות.

פתרון: הוספנו שירות IServiceProvider שמנהל את כל ה-Services ומונע יצירת סירקולציות.

שירותים חסרים בקובץ Program:

4. שכחתי לרשום את השירותים ב-Program, מה שגרם לבעיות בהזרקת התלויות.

פתרון: הוספתי את השירותים באמצעות AddScope כדי להבטיח שהמערכת תזהה אותם כראוי.

בעיית קישור בין MVC ל-API:

5. נתקלתי בבעיית קישור בין MVC ל-API שהתבררה כנובעת מהפעלת התוכנית על HTTPS במקום HTTP.

פתרון: עדכנתי את הגדרות השרת להריץ על HTTP כדי לפתור את בעיית הקישור.

בחירת גישה להבאת מידע ל-MVC:

6. היה אתגר בהחלטה האם להביא את כל המידע ל-MVC ולסנן שם, או להביא רק את המידע המבוקש.

פתרון: החלטתי להביא את כל המידע ולהשתמש בפונקציות קיימות לסינון, במקום ליצור פונקציות חדשות.

תזמון וביצוע פונקציות:

7. שמתי לב לצורך לוודא שכל הפונקציות מבוצעות במקומות ובזמנים הנכונים בקוד.

פתרון: עברתי על הקוד היטב כדי להבטיח שכל דבר ירוץ בצורה תקינה ובתזמון הנדרש.

יצירת פונקציות טהורות (Pure Functions):

סיכום

במהלך הפרויקט נתקלתי באתגרים שונים שנבעו משגיאות נפוצות, אך באמצעות פתרונות יצירתיים ושיפור תהליכים, הצלחתי להתגבר עליהם ולהביא את הקוד לרמה גבוהה של יציבות ואיכות.

תודה על ההשקעה במהלך החודשיים האחרונים😊