הנה העדכון המבוקש לסעיף 7 במסמך:

---

## 7. פתרון טכנולוגי נבחר:

### 7.1. טופולוגית הפתרון

המערכת תהיה תוכנה שולחנית (desktop application) שתתפרש על מספר מרכיבים:

- \*\*שרת נתונים מקומי\*\*: לניהול ואחסון בסיס הנתונים.

- \*\*ממשק משתמש בצד הלקוח\*\*: יישום שולחני מבוסס WINFORM בשפת C#.

- \*\*טווח תקשורתי\*\*: שימוש בחיבור רשת מקומית (LAN) עם אפשרות לגישה דרך האינטרנט לצורך גיבוי ושיתוף מידע בין גופים שונים.

- \*\*בסיסי נתונים SQL Server\*\*: לניהול המידע.

### 7.2. טכנולוגיות בשימוש

טכנולוגיות בשימוש יהיו WINFORM בשפת C# ובסיסי נתונים SQL Server. הבחירה ב-C# ו-WINFORM נובעת מכך ש-C# היא שפה מתקדמת, יציבה ובעלת ביצועים גבוהים, המתאימה לפיתוח יישומים בסביבת .NET של מיקרוסופט. SQL Server נבחר כמערכת לניהול בסיסי נתונים בגלל ביצועיו הטובים ותמיכתו המלאה בסביבת .NET.

### 7.3. שפות הפיתוח

C# היא שפת תכנות שפותחה על ידי מיקרוסופט, והיא נמצאת בשימוש נרחב בעיקר לפיתוח יישומים בסביבת .NET. השפה מתמקדת בבטיחות, ביציבות ובביצועים גבוהים. היתרון המרכזי של C# הוא ההתמקדות בכלים ובתשתיות המסופקות על ידי מיקרוסופט, כך שמשתמשי C# יכולים להפיק מועדונים רבים מהיתרונות של סביבת הפיתוח של .NET. השפה תומכת בתכנות מונחה עצמים (OOP) ומספקת תמיכה מובנית לפיתוח יישומים רבי-לתחומים ומורכבים.

### 7.4. תיאור הארכיטקטורה הנבחרת

הארכיטקטורה של המערכת תהיה מבוססת על תוכנה שולחנית שתאפשר שיתוף מידע בין גופים שונים. המערכת תכלול מרכיבים מרכזיים כגון ניהול מלאי, ניהול הזמנות, ניהול לקוחות, ניהול ספקים, ודוחות וניתוחים.

הנה העדכון המבוקש לסעיף 8.1:

## 8.1. תיאור הארכיטקטורה הנבחרת

הארכיטקטורה הנבחרת מתבססת על תוכנה שולחנית (desktop application) שתאפשר לשתף מידע בין גופים שונים. הבחירה בתוכנה שולחנית נובעת מהיתרונות הבאים:

- \*\*ביצועים גבוהים\*\*: תוכנה שולחנית מספקת ביצועים מהירים ותגובה מיידית, מכיוון שהיא פועלת ישירות על המחשב המקומי של המשתמש.

- \*\*בטיחות מידע\*\*: מכיוון שהנתונים נשמרים מקומית, ניתן לשלוט בצורה טובה יותר על גישת המשתמשים לנתונים ולמנוע גישה לא מורשית.

- \*\*גישה וניהול נוח\*\*: קל לתחזק ולעדכן את התוכנה כאשר כל הרכיבים נמצאים במחשב המקומי.

- \*\*שיתוף מידע\*\*: באמצעות חיבור לרשת האינטרנט ושימוש בשירותי אירוח, ניתן לשתף מידע בצורה מאובטחת בין גופים שונים, לשמור על עדכניות הנתונים ולהבטיח גיבויים קבועים.

הנה העדכון המבוקש לסעיף 9.2:

## 9.2. נא פרט את שיטת האיחסון (מאגר, קבצים ובאיזה טכנולוגיה)

המערכת תשתמש במאגר בסיסי נתונים MySQL לניהול ואחסון המידע. MySQL נבחר כמערכת לניהול בסיסי נתונים בגלל ביצועיו הטובים, יכולת ההתאמה הגבוהה והגמישות הרבה שהוא מציע. MySQL מאפשר ניהול יעיל של נתונים, אחסון מידע בצורה מאובטחת, וביצועים מהירים בשימוש במאגרי נתונים גדולים.

הנה העדכון המבוקש לסעיף 9.3:

## 9.3. נא ציין מנגנוני התאוששות מנפילה\קריסה\תמיכה בטראנזקציות

המערכת תשתמש במנגנונים הבאים להתאוששות מנפילה או קריסה ולתמיכה בטראנזקציות:

- \*\*רישום (Logging)\*\*: המערכת תשתמש ברישום לשמירת פעולות המבוצעות ולאפשר תיעוד ובדיקה של המצב של המערכת לאחר חזרה מנפילה או קריסה. הרישום יכיל מידע מפורט על כל פעולה שבוצעה, כולל זמן הביצוע ומשתמש המבצע, ויאפשר זיהוי וטיפול בתקלות במהירות.

- \*\*טרנזקציות (Transactions)\*\*: המערכת תשתמש בטרנזקציות כדי להבטיח קבלת מצב תקין לאחר החזרה מנפילה או קריסה. זה כולל נעילת משאבים וביצוע פעולות בצורה אטומית, כך שכל טרנזקציה תתבצע במלואה או לא תתבצע כלל, ובכך נמנע מצב של נתונים חלקיים או לא תקינים.

- \*\*גיבוי (Backup)\*\*: המערכת תשתמש בגיבויים כדי להבטיח שנתוני המערכת יוכלו להשתקם לאחר נפילה או קריסה. הגיבויים יאוחסנו במקום מתאים ובתדירות שתוכננה מראש, כך שבמקרה של נפילה או קריסה ניתן יהיה לשחזר את הנתונים למצב תקין.

הנה העדכונים המבוקשים לסעיפים 12 ו-13:

---

## 12. תיאור/התייחסות לנושאי אבטחת מידע

יש להתייחס לאזורים הדורשים אבטחה וכיצד ניתן מענה, כגון:

- \*\*שרת\*\*: הגנה על השרת מפני גישה בלתי מורשית באמצעות חומות אש (Firewall) ומערכות למניעת חדירות (IDS/IPS).

- \*\*בקרת גישה לאתר\*\*: שימוש בפרוטוקולי SSL/TLS להצפנת התקשורת בין המשתמשים לבין השרתים, והבטחת שלא יינתן גישה ללא זיהוי ואימות.

- \*\*חשבונות משתמשים\*\*: ניהול משתמשים והרשאות תוך שימוש בסיסמאות חזקות, אימות דו-שלבי (2FA), ובקרה על ניסיונות כניסה חשודים.

- \*\*מאגרי מידע\*\*: הצפנה של נתוני המידע במסדי הנתונים, גיבוי סדיר של הנתונים וניטור גישה למסדי הנתונים כדי למנוע גישה בלתי מורשית.

מקרים ותגובות:

- \*\*ניסיון חדירה למערכת\*\*: במקרה של זיהוי ניסיון חדירה, המערכת תחסום את הגישה מהכתובת החשודה ותשלח התראה למנהל המערכת.

- \*\*גניבת זהות משתמש\*\*: במקרה של חשד לגניבת זהות, חשבון המשתמש יינעל ותשלח הודעה למשתמש לצורך אימות זהותו מחדש.

- \*\*גישה בלתי מורשית לנתונים רגישים\*\*: שימוש במערכת ניהול הרשאות קפדנית שתאפשר גישה לנתונים רגישים רק למשתמשים מורשים ומאומתים.

נקודות האבטחה המרכזיות כוללות שמירה על נתוני הלקוחות, המלאי והתהליכים באמצעות ניהול גישה מאובטח, הצפנת נתונים במסדי הנתונים, בקרת גישה מבוקרת וניטור פעילות תוך שמירה על תקינות המערכת. מתוך כך, ניתן להשתמש במנגנונים כגון SSL/TLS להצפנת התקשורת, גישה מבוקרת למערכת, והגבלת גישה לנתונים בהתאם לרמות ההרשאה.

---

## 13. משאבים הנדרשים לפרויקט

### 13.1. מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות

- \*\*מספר שעות\*\*: הפרויקט דורש כ-800 שעות עבודה סך הכל.

- \*\*חלוקת עבודה\*\*:

300 שעות

### 13.2. ציוד נדרש

- מחשבים עם מפרט טכני מתאים לפיתוח ובדיקות.

- שרתים לבדיקות ולסביבת הפקה.

- ציוד גיבוי ואחסון.

### 13.3. תוכנות נדרשות

- Visual Studio לפיתוח בסביבת C#.

- SQL Server לניהול בסיסי נתונים.

- תוכנות לניהול גרסאות קוד (Git).

- תוכנות לבדיקות (כגון Selenium).

### 13.4. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

- ניהול בסיסי נתונים ב-MySQL.

- שימוש במנגנוני אבטחת מידע מתקדמים.

- הטמעת פרוטוקולי SSL/TLS.

### 13.5. ספרות ומקורות מידע

- תיעוד רשמי של Microsoft על פיתוח ב-C# ובסביבת .NET.

- מדריכים ותיעוד של MySQL.

- ספרים ומאמרים על ניהול פרויקטי תוכנה ואבטחת מידע.