

מדינת ישראל

הנהלת בתי המשפט

**המבחן נועד לספק הערכה של הידע שלך ושל יכולתך להתמודד עם אתגרים בתכנון ופיתוח פתרונות טכנולוגיים בנושאים שבהם תעבוד/י במסגרת "הנהלת בתי המשפט". המבחן תוכנן לדמות את הסביבה המעשית והיומיומית שבה תתפקד/י, ולכן איננו מצפים ממך לזכור בעל-פה מושגים או לכתוב תשתיות מאפס.**

לרשותך עומדים **כל** האמצעים שברצונך להשתמש בהם, **כולל כלים מתקדמים כמו AI**, כדי לגבש את תשובותיך.  
במסגרת ראיון שנקיים, תתבקש/י להסביר חלק מהתשובות שנתת ולפרט על התהליכים והמחשבה שהובילו לבחירות שלך.

במהלך המבחן נבקש ממך להשתמש בכלים מסוימים.  
חשוב לנו להבהיר שברור לנו שלא כל אחד/ת חייב/ת להכיר כל כלי, וזה מובן לגמרי.

התרגילים יכולים להיות ארוכים עבורך, נשמח לראות את הדברים העיקריים ואת צורת החשיבה שלך. אין חובה לסיים הכל (תרגיל מלא יוסיף ניקוד), חובה שהקוד ירוץ ללא שגיאות.

* נושא המבחן הוא הקמת אתר פניות ציבור -

בהצלחה!

## SQL

כתוב stored procedures שמחזירה דוח פניות חודשי. הדוח צריך להציג את סך הפניות לכל מחלקה, עם השוואה לחודש הקודם ולאותו חודש בשנה שעברה. הניח שיש לך טבלאות של פניות קיימות.  
הסבר את השאילתה והצע באיזה אופן ניתן לשפר את הביצועים.

מצורף בקבצים נפרדים

## .NET Core

1. פתח Web API ב-.NET עם פעולות CRUD לטופס פניות המכיל את השדות:  
שם, טלפון, אימייל, מחלקה[ ], תיאור הפנייה, כפתור שליחה.

כתוב Unit Tests בסיסי.  
הוסף endpoint המציג את התוצאה של stored procedures המחזירה דוח פניות חודשי.  
כתוב קובץ ReadMe.md, המסביר על תכולת הפרוייקט, שיטת הבנייה והרכיבים שבחרת יתרונות/חסרונות, איך מטפלים בנושא האבטחה, טיפול בשגיאות, מנגנוני קישור, קרוס דומיין, והוראות התקנה והפעלה.

לא חובה לסיים את הכל או לכתוב חיבור לDB אמיתי. אבל הפרויקט חייב לרוץ.

בסיום לשלוח קישור ל GIT. אם לא ניתן אפשר בקישור לתקייה ברשת (Drive למשל).

2. ציין עם √ ליד שם הכלי האם את מכיר אותו והאם עשית בו שימוש בעבר.

אם יש לך הערות נשמח שתציין.

|  |  |  |
| --- | --- | --- |
| **נושא** | **מכיר או לא** | **מתי עשית בו שימוש** |
| **CRUD** | כן | שימוש תדיר בטיפול בנתוני DB |
| **Exception Handling Middleware** | כן | שימוש רב בהחזרת נתונים מהשרת. |
| **JWT Authentication** | כן | יצא לי להשתמש באפליקציות מאובטחות |
| **EF Profiler** | לא | לא יצא לי להשתמש |
| **EF Core** | כן | שימוש מועט |
| **eager loading** | כן | שימוש רב, בעיקר אפליקציות בסדר גדול קטן או בינוני, טעינת נתונים מהשרת בכניסה למערכת |
| **lazy loading** | כן | באפליקציות גדולות יותר |
| **SignalR** | כן | עם טכנולוגית WebSockets |
| **aspose dll** | כן | שימוש להמרת PDF, (לא בטוחה) |
| **Exception Handling** | כן | שימוש מורחב באפליקציות, צד לקוח |
| **Micro-Services** | כן | במערכות גדולות לדוג' לצורך שליחת מייל בתוך האפליקציה |
| **K8S** | לא |  |
| **CI/CD** | כן | יצא לי להשתמש בהעלאת גרסאות אך לא לפתח את הכלי |
| Minimal API | לא |  |

## Angular

1. צור אפליקציית Angular שתתקשר עם ה-API שיצרת, לטופס פניות המכיל את השדות: שם, טלפון, אימייל, מחלקה[ ], תיאור הפנייה, כפתור שליחה.
2. הוסף דף שמציג את התוצאה של stored procedures שמחזירה דוח פניות חודשי.
3. אתה רשאי להשתמש בכל גרסת אנגולר שתרצה.
4. הממשק צריך להיות מותאם היטב למכשירים ניידים ולעבוד חלק במסכים בגדלים שונים.
5. שימוש באחת מהאופציות הנ"ל יהוה בונוס. Material, Kendo, PrimeNg .
6. הוסף קובץ README.md ותפרט על הפרוייקט, במה התמודדת, איך לפרוס, שיקולים בבחירת הגרסה, וכדו'.
7. בסיום לשלוח קישור ל GIT. אם לא ניתן אפשר בקישור לתקייה ברשת (Drive למשל)
8. ציין עם √ ליד שם הכלי האם אתה מכיר אותו והאם עשית בו שימוש.

אם יש לך הערות נשמח שתציין.

|  |  |  |
| --- | --- | --- |
| **נושא** | **מכיר או לא** | **עבור מה זה משמש?** |
| **PIPE** | כן | שימוש בצינורות (pipes) להעברת ערכים בתבנית. לדוגמה: {{ value | date }} , לא נעשה שימוש בתרגיל |
| **Virtual Scrolling** | כן | טכניקה להצגת רשימה ארוכה עם גלילה בדרך כלל משתמשים ברכיב כמו <cdk-virtual-scroll-viewport>.  לא נעשה שימוש בתרגיל |
| **ChangeDetectionStrategy.OnPush** | כן | בדוק שינויים בקומפוננטה הזו רק כשיש שינוי במידע שמגיע אליה (input properties) או כשיש אירוע מפורש בקומפוננטה (כגון אירוע UI או Observable שמשדר ערך חדש ,  לא נעשה שימוש בתרגיל |
| **שימוש ב $ למשל: $user** | לא מספיק | סימון של Observables בשמות המשתנים. לא נעשה שימוש בתרגיל |
| **Observable** | כן | הוא אובייקט או ישות שמייצגת **זרם של נתונים או אירועים** שניתן להאזין להם ולקבל מהם ערכים לאורך זמן. מאפשר טיפול בזרמים אסינכרוניים של מידע בצורה נוחה, גמישה ויעילה.  נעשה שימוש ע"י קריאה ל- .subscribe() ב-contactService.submitContact() וב-contactService.getMonthlyReport() |
| **Angular DevTools** | כן | כלי ניתוח חיצוני (לא קוד),  אין שימוש ישיר בקוד. |
| **Material, Kendo, PrimeNg** | כן- **Material** | שימוש ברכיבי UI מספריות אלו,   * שימוש לדוג': mat-form-field * mat-select * mat-option * mat-table |
| **async** | כן | שימוש בצינור async בטמפלייט:  לא עשה שימוש בתרגיל |
| **lazy loading** | כן | הגדרת מודולים שמיטענים באופן עצל (Lazy) ב-Routing,  ז"א רק בעת הגלילה לשם,  לא נעשה שימוש בתרגיל |
| **`trackBy with `ngFor** | כן | משתמשים בפונקציית trackBy ב- ngFor לשיפור ביצועים:,  לא נעשה שימוש בתרגיל |
| **canActivate** | לא מספיק | שימוש ב-Guard ב-Routing:  לא נעשה שימוש בתרגיל |
| **@Input()** | כן | דקורטור לקבלת פרמטרים מרכיב הורה:  לא נעשה שימוש בתרגיל |
| **אם יש לך מה להוסיף זה יהיה נחמד** |  |  |

## CRM

מלא את הטבלה (מה שהינך מכיר בלבד):

לא יצא לי להתעסק עם CRM

|  |  |
| --- | --- |
| **נושא** | **למה משמש?** |
| **תהליכים עסקים** |  |
| **ישויות** |  |
| Workflows |  |
| SLA |  |
| Plugins |  |
| SLA KPI |  |
| כיצד ניתן לעשות Debugging ל-Workflows ול- Plugin |  |
| **איזה CRM אתה מכיר?** |  |

Claude

1. האם יצא לך לעבוד עם Beadrock?   
2. מה מידת ההיכרות שלך עם Linux?

3. חברה רוצה להעביר את המערכת הקיימת On Preme שלה ל-ענן:

תאר את תהליך שתעבור לתכנון וביצוע המעבר.

התייחס לאתגרים אפשריים, אסטרטגיות להעברת נתונים, ושיקולים לאינטגרציה עם מערכות קיימות.

תשובות:

1. לא

2. עבדתי בעבר עם מערכת של לינוקס מכירה את הפקודות וההתעסקות איתה

3. **תהליך לפני המעבר:**

**1.  הערכת המערכת הקיימת**

* מיפוי מלא של כל רכיבי המערכת: שרתים, אפליקציות, מסדי נתונים, חיבורים למערכות חיצוניות.
* זיהוי תלות בין רכיבים.
* הבנת דרישות ביצועים, אבטחה, זמינות וגיבוי.
* הערכת עומסי עבודה ונתונים.

**2. הגדרת מטרות ויעדים**

* החלטה על סוג הענן (פרטי, ציבורי, היברידי).
* בחירת ספק ענן (AWS, Azure, GCP וכו').
* הגדרת קריטריונים להצלחה: זמינות, מהירות גישה, עלות וכו'.

**3.  תכנון ארכיטקטורת ענן**

* החלטה על מודל פריסה: IaaS, PaaS, SaaS.
* תכנון סקיילביליות, ניהול עומסים, גיבויים.
* תכנון אבטחה: ניהול זהויות, הצפנה, firewall, IAM.

**4. אסטרטגיות העברת נתונים**

* **גיבוי ראשוני והורדה (Backup and Restore)** – העתקה של גיבוי מסדי הנתונים לשרתי הענן ושחזור.
* **סנכרון נתונים (Data Sync)** – שימוש בכלים לסנכרון בין המקור לענן בזמן אמת או בזמנים קבועים.
* **העברה פיזית (Physical Transfer)** – למשל עם שירותים כמו AWS Snowball להעברת כמויות גדולות של נתונים.
* **העברה הדרגתית (Incremental Migration)** – העברת נתונים בהדרגה, עם בדיקות רציפות.

**5. ביצוע המעבר (Migration Execution)**

* יצירת סביבת הענן (סביבת בדיקות ראשונית).
* העתקת הנתונים והרכיבים.
* בדיקות אינטגרציה, ביצועים ואבטחה.
* מעבר הדרגתי או מיידי בהתאם למורכבות.
* **Cutover** – הפיכת הסביבה בענן לסביבה הראשית.

**6. בדיקות ואופטימיזציה לאחר המעבר**

* בדיקות עומס, זמני תגובה ויציבות.
* ניטור רציף.
* כוונון משאבים לצורך חיסכון בעלויות ויעילות.

**7. הדרכה ותמיכה**

* הכשרת צוותי IT ותפעול.
* שדרוג תהליכים ותמיכה טכנית.

**אתגרים אפשריים במעבר לענן**

* **מורכבות אינטגרציה** – התאמת ממשקים עם מערכות קיימות שלא עובדות בענן.
* **הפרעות בשירות בזמן המעבר** – הפסקות שירות שעלולות לפגוע בעסקים.
* **אבטחת מידע ורגולציה** – שמירה על פרטיות ועמידה בתקנים.
* **ניהול עלויות** – עלויות בלתי צפויות בעקבות שימוש במשאבים ענניים.
* **התאמת תשתית ותוכנה** – צורך לשדרג או להתאים אפליקציות שלא מותאמות לענן.
* **חששות מביצועים** – הבטחת זמן תגובה ואמינות כמו בסביבה On-Premise.

**שיקולים לאינטגרציה עם מערכות קיימות**

* **API ושכבות שירות** – שימוש בממשקי API סטנדרטיים כדי לשמור על תקשורת בין מערכות On-Premise לענן.
* **פתרונות Middleware** – התקנת תוכנות תיווך לניהול זרימת מידע ואבטחה.
* **סינכרון נתונים** – קביעת דרכי סנכרון ועדכון בין מערכות בזמן אמת או בזמנים קבועים.
* **ניהול זהויות והרשאות** – אינטגרציה עם מערכות ניהול זהויות (כגון Active Directory) לשמירה על הרשאות אחידות.
* **בדיקות תאימות** – בדיקת תאימות גרסאות תוכנה, פורמטים ופרוטוקולים.

## הכרת המועמד/ת:

(לא חובה לענות על כל שאלה)

1. תאר/י פרויקט שעבדת עליו לאחרונה. מה היו האתגרים העיקריים ואיך התמודדת איתם?

2. איך את/ה מתעדכן בטכנולוגיות חדשות ומגמות בתעשייה?

3. תאר מצב שבו היית/ה צריך לעבוד עם טכנולוגיה או כלי שלא היכרת, איך התמודדת עם המצב?

4. איך את/ה מתמודד עם דדליינים צפופים ולחץ בעבודה?

5. איך את/ה מתמודד עם חילוקי דעות בצוות לגבי גישות טכניות או ארכיטקטוניות?

6. מה המוטיבציה שלך לעבוד בתחום הפיתוח במשרד ממשלתי?

7. היכן את/ה רואה את עצמך מקצועית בשנים הקרובות?

8. איך היה המבחן?

תשובות:

1. כשעבדתי במשטרת ישראל פיתחתי טפסי אנגוולר דינמיים, הטפסים היו אמורים לרדת גם בפורמט PDF אך בצורה מאובטחת ולעמוד בכללים משפטיים מדויקים, הייתי צריכה ללמוד צורות להמרת PDF לאחר מכן להחליט באיזה צד תבוצע ההמרה- קליינט או סרבר ומשום שבסרבר זה היה יותר מאובטח היה צורך לחקור ולהתמודד עם בניית תבניות HTML בצד ה-C# ושתילת הנתונים בטמפלייטים המיועדים לכך

2. טכונלוגיות חדשות אני לומדת או ע"י AI או ע"י קורסים אינטרנטיים, לדוג' קורס בחשכל שנתי

3. יצא לי לאחרונה לעבוד עם שפת ריאקט עם סיפריה שלא הכרתי, פתחתי מדריך המסבי את הרעיון והשימוש בו ולאחר מכן ניסיתי ללמוד לפי הקוד של הפרויקט

4. אני עובדת לפי משימות מהדחופות ביותר לפחות, ולפי רשימות עצמאיות שאני עורכת לעצמי

5. מנסה לגשר ולהפעיל לדוג' מישהו מדרג גבוה יותר, כמו יועץ חיצוני של הארגון, שהוא יכריע

6. אני אוהבת לעבוד בארגונים גדולים אך אם זאת מחפשת בכל פרויקט כן להרגיש שאני מתקדמת וכותבת קוד ברמה גבוהה עם שימוש בטכנולוגיות רלוונטיות וחדשניות

7. אני שואפת להכיר ולהשתמש בטכנולוגיות חדשות שלא יצא לי לעבוד איתן לעומק כמו .core להרחיב את הידע בעולמות הענן ושימוש מסיבי עם AI לקידום פרויקטים גדולים

8. המבחן היה קצת ארוך, מקיף נושאים מסוימים

תודה לך!

שרה לוין