מבחן מפתח תוכנה רמה ג – משרד המשפטים

גירסה 1.1

כללי:

המבחן כולל חלקים שבודקים את יכולות הפיתוח בצד Client, וחלקים בצד Server וכן ידע הקיים אצל המפתח/ת כולל ידע בסיסי באנגלית מקצועית.

פתרון של מערכת שלמה עובדת הוא מצוין, אך בנוסף יינתן דגש וניקוד לפי המדדים הבאים (הם חשובים יותר מהשלמת פתרון עובד – שימו לב להשקיע בהם זמן מתאים)

1. כתיבת קוד בצורה נכונה, מונחית Best Practices
2. ארכיטקטורה של הקוד בצד Client ובצד Server – חלוקה למודולים, לשכבות ולרכיבי קוד מתאימים
3. שימוש בספריות Client מודרניות
4. שימוש נכון ב- CSS – שליטה ביכולות עיצוב
5. כתיבת Rest API באופן נכון
6. הפגנת ידע תיאורטי

פתרונות לצד Client – ניתן לעשות שימוש ב – HTML, JavaScript, Css

ובספריות Client מודרניות: Angular, React, Vue.js, knockout

ניקוד גבוה יותר יינתן לפתרון שכתוב באחת מספריות Client מודרניות שצוינו.

לא יתקבל פתרון שמשתמש בפקדי צד שרת (לדוגמא ASP.Net)

פתרון לצד שרת – ניתן לכתוב ב - .Net FW או ב - .Net Core

על השאלות התיאורטיות יש לענות בגוף המסמך בכתיבה במחשב (לא להדפיס ולענות בכתב יד) , ניתן להוסיף שורות לפי הצורך. שאלות אמריקאיות יש לסמן את התשובה הנכונה.

כל חומרי עזר מותרים בשימוש אולם יש לכתוב את הפתרון באופן עצמאי, ללא עזרה של אף גורם נוסף. תשאלו על הפתרון בראיון.

אם לא הספקתם לענות על כל הסעיפים – כתבו את דרך הפתרון הטכנית שהייתם מממשים.

בהצלחה!

**חלק א – צד Cliet**

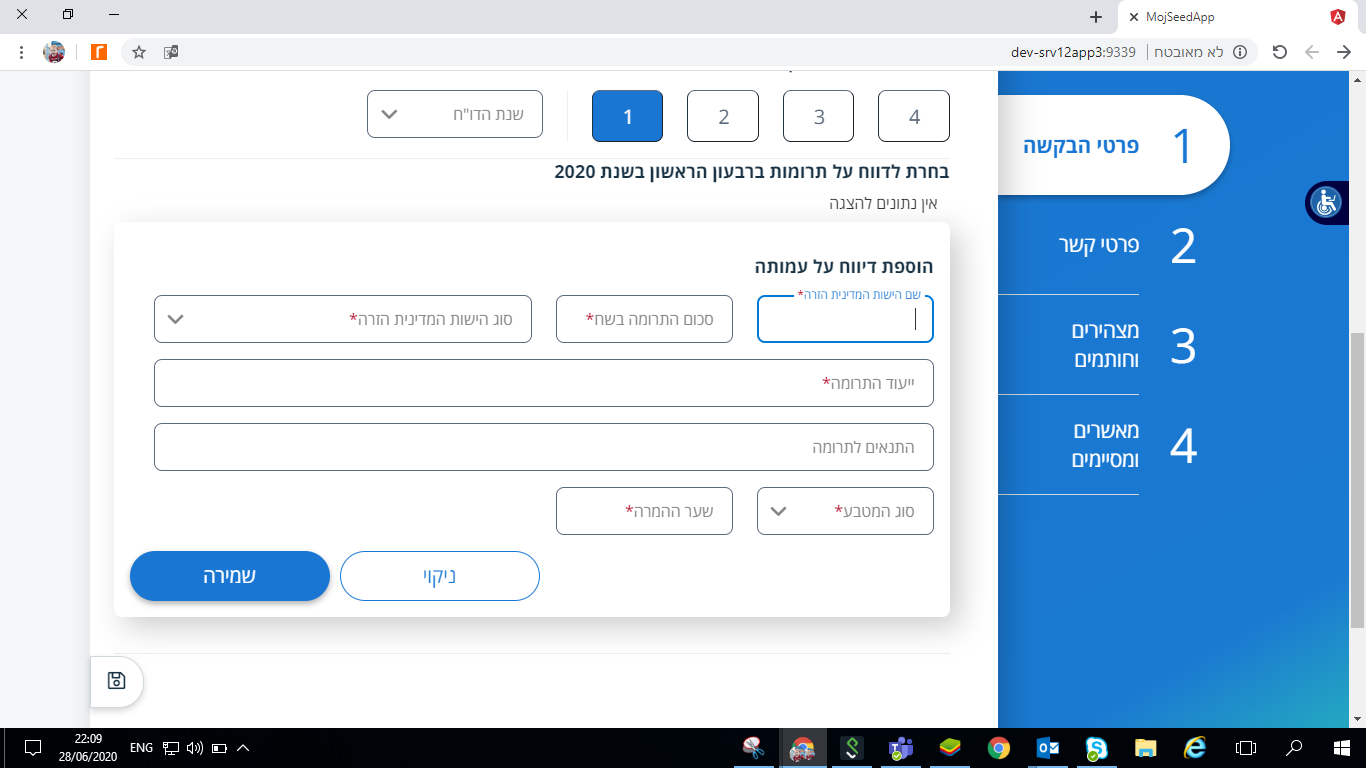
יש לפתח אפליקציית CLIENT שמממשת את היכולות הבאות.

יש להשקיע בעיצוב הרכיבים, קרוב ככל הניתן לתמונה (אין חובה להגיע לדיוק, אלא לרמת עיצוב כללי)

יש להקפיד על כתיבה נכונה בכל חלקי האפליקציה: html, css, ts.

1. **רכיב הוספת קובץ (אין צורך לעבוד עם קבצים ממש אלא רק במידע עליהם)**

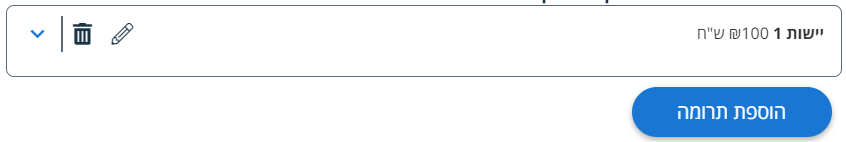
* יש לפתח כרטיסיה שתכלול את השדות הבאים, כפי שנראה בתמונה (שדות אחרים נראות דומה)
  + מספר סידורי – אוטומטי\*
  + שם קובץ \*
  + סוג הקובץ \* (PDF , Docx, xlsx, pptx, jpg) – בחירה מרשימה
  + גודל הקובץ \* – אפשרות הקשה או אפשרות הגדלה והקטנת ערך באמצעות +/- או חיצים וכדומה , אבל גם רכיב שמאפשר הגדרה לפי K,M,G Byte
  + המחבר
  + תאריך חיבור
  + האם מוצפן – כן/לא
* השדות המסומנים בכוכבית הם שדות חובה, וצריך לממש בהתאם.
* כפתור שמירה יעביר את הנתונים לרשימה שמופיעה בסעיף 2.



1. **רשימת הקבצים**

2.1 הרשימה תכיל כרטיסיה שתציג את כל השדות.

כפתור הוספת קובץ יפתח את הכרטיסייה שבסעיף 1, ויאפשר הוספת פריטים לרשימה.



2.2 יש להציג בתחתית הרשימה את סכום של נפח הקבצים ברשימה .

2.3 יש להוסיף כפתור לרשימה בשם "הכנה להפצה" , שיחשב חבילות הפצה של קבצים שלא יעלו על 1GB , ויסמן ברשימה כל חבילה בצבע אחר. (יש לכתוב אלגוריתם בשירות שיקבל רשימת הקבצים +גדלים ויחזיר את הרשימה + מספר חבילה לכל קובץ) .

\* אלגוריתם הוא בונוס , זאת בעיה מסוג NP-Complete , לכן ללא ידע מתמטי , מומלץ לכתוב אלגוריתם מקורב לדוגמא :

1. לסכם את סכום גודל בקבצים .

2. לחלק במגבלת הגודל.

3. לפתוח N מערכים (N= עיגול כלפי מעלה של החלוקה)

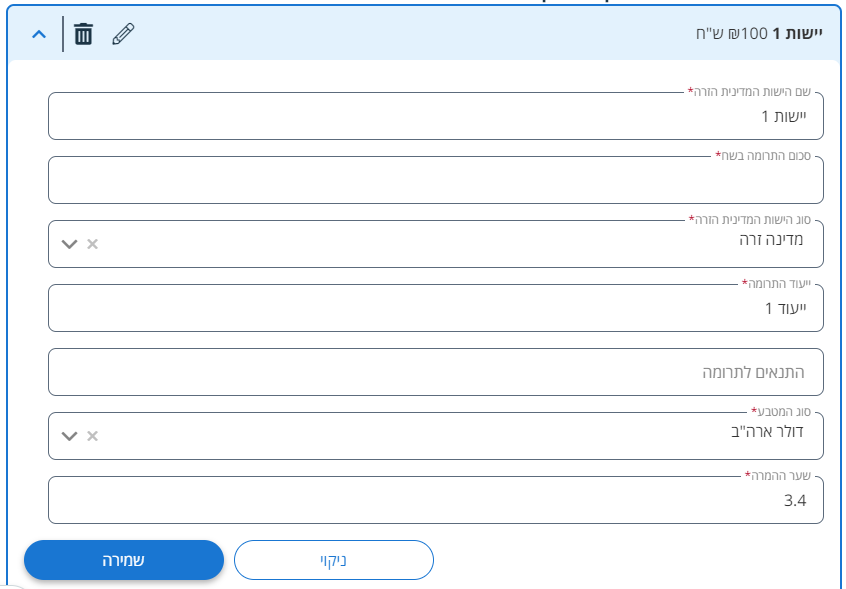
4. ולהריץ לופים למילוי המערכים עד לגודל המגבלה.

2.4 יש להוסיף כפתור נוסף בשם "פיצול קובץ" , שיתאפשר רק על קבצים מעל 1GB , ויקרא לשירות שייצר ממנו מספר קבצים באותו שם עם תוספת של מספר לשם הקיים , שכל קובץ יהיה קטן מ- 1GB.

**עריכת קובץ**

לחיצה על העיפרון (כפתור בצד שמאל של הרשימה) תחליף את הכרטיסייה למצב עריכה:

שימו לב שהכרטיסייה משתנה ממצב תצוגה למצב עריכה.



**חלק ב – צד Server**

1. יש לפתח שירות Web API שיקלוט את פרטי הקובץ למטרת שמירת ב – DB
2. אם גודל הקובץ גדול מ –100MB יש לשלוח מייל לכתובת מייל לבחירתכם (אפשר למייל האישי)
3. אין צורך לממש בסיס נתונים ולכתוב את פונקציית השמירה ב – DB עצמו (בונוס), אלא רק להכין חתימת פונקציה שתתאים לעדכון DB , ניתן גם לכתוב לקובץ Json (מומלץ)

**חלק ג – כללי**

1. יש לממש את ולדיצות הבאות:
   1. שדות חובה – על השדות המסומנים בכוכבית
   2. שדות טקסטואליים – תווים בעברית ובאנגלית בלבד
   3. בשדה קוגל הקובץ – יש לאפשר הזנת מספר בלבד. הוולידציה תאפשר הזנה של מספר עשרוני.
2. יש לתת התייחסות (אין חובה לממש) למימוש הרשאות עריכת רשומה רק עבור המשתמש שיצר את הרשומה.

שימו-לב יש שאלות בהמשך (מילוליות או בחירה)

שאלות על .Net Core

1. האם .Net Core יכול לעבוד עם .Net Framework ?  
   לא
2. מה ההבדל בין SDK ל- Runtime ב- .Net Core ?
3. מה ההבדל בין Task ל- Thread ב- .Net Core ?
4. מה היתרונות של שימוש ב- AOT ב- .Net Core ?
5. מה זה Kestrel ומתי נשתמש בו ?

kasterl זה השרת הדיפולטיבי של .net core.

1. למה משמש startup class ב- .Net Core ?  
   startup class משמש להגדרות קונפיגורציה גלובליות של הפוריקט.
2. מה היתרונות של .Net Core ?  
     
   שהיא מבוססת קוד פתוח ו cross-platform.

מנגנון Dependency Injection כחלק אינטגרלי

שיפור הביצועים שהוקפצו לרמות גבוהות בהרבה

1. האם עבדת עם .Net 5 ? יש לי נסיון עם .Net Core
2. הסבר/י מה זה Middleware ב- ASP.Net Core ?  
   זה component (מחלקה) שמבוצע בכל בקשה ביישום ASP.NET Core. ניתנת אפשרות לניהול סדר הקריאות בכל בקשה.
3. הסבר/י מה זה Filters וכמה נתמכים ב- ASP.Net Core

שאלות על מיקרו סרוויסים

1. האם יש לך ניסיון בפיתוח Microservices ? לא
2. מה הם העקרונות של עיצוב Microservices ?
3. מה היה התפקיד שלך בפרויקט האחרון שקשור ל- Microservices ?
4. מה האתגרים שהיו לך בעת מימוש Microservices ?
5. האם יש צורך ב- DB חדש ונפרד לכל Microservices ? אם כן , מדוע ?
6. האם אפליקציה מונוליטית יכולה לתקשר עם Microservices ?
7. איך לדעתך משתלב GraphQL בעיצוב של Microservices ?

**Question 1 of 9**

What is a microservice?

1. A design used primarily in functional programming and object-oriented programming
2. A small program that represents discrete logic that executes within a well-defined boundary on dedicated hardware
3. A style of design for enterprise systems based on a loosely coupled component architecture
4. A very small piece of code that never gets any bigger than 10 lines

**Question 2 of 9**

When would developers use microservices?

1. When they want to write cell phone applications that run quickly
2. When they work with ephemeral nano technology
3. When they need to create large, enterprise-level applications that are subject to changes on a frequent basis
4. When they create applications specifically for scientific test equipment

**Question 3 of 9**

Which of the following responses is an advantage of microservices?

1. Any microservice component can change independently from other components
2. They don't require a lot of expertise to program
3. They're so small that developers can typically write very powerful ones with a few lines of text
4. They are easy to manage

**Question 4 of 9**

Which of the following responses is a disadvantage of microservices?

1. Microservices are very difficult to manage at scale
2. Microservices require a lot of monitoring to operate effectively
3. Neither A nor B
4. Both A and B

**Question 5 of 9**

What are some technologies commonly used to implement microservices?

1. Docker
2. Kubernetes
3. All the above
4. None of the above

**Question 6 of 9**

How is distributed tracing used in microservices?

1. As a mechanism to ensure that failed microservices are resurrected properly
2. As a mechanism to transfer log management between a given host on demand
3. As a mechanism to observe the behaviour of distinct system calls between and within microservices
4. As a mechanism to change the behaviour of a microservice at runtime

**Question 7 of 9**

How does a backing service apply to microservices?

1. It prevents a microservice from failing
2. It acts as a dedicated service that provides essential functionality required by a microservice
3. It shuts down a microservice when it can no longer handle the computing load
4. It coordinates network activity between microservices

**Question 8 of 9**

What's the difference between a microservices-oriented architecture (MOA) and a service-oriented architecture (SOA)?

1. An SOA uses intermediation technology to facilitate communication between services
2. An MOA shares as little data as possible while an SOA shares as much data as possible
3. A developer can run a monolithic application with SOA principles
4. All the above

**Question 9 of 9**

What does the term "bounded context" mean in relation to microservices?

1. How a microservice uses memory
2. The logical domain represented by the data consumed and emitted by a microservice according to the data's purpose, structure and meaning
3. The cloud provider's region where the microservice operates
4. The common characteristics within a collection of entities in one or many business domains