משימת פיתוח

כללי:

משימת הפיתוח הבאה כוללת את החלקים הבאים:

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

**חלק א – צד Cliet**

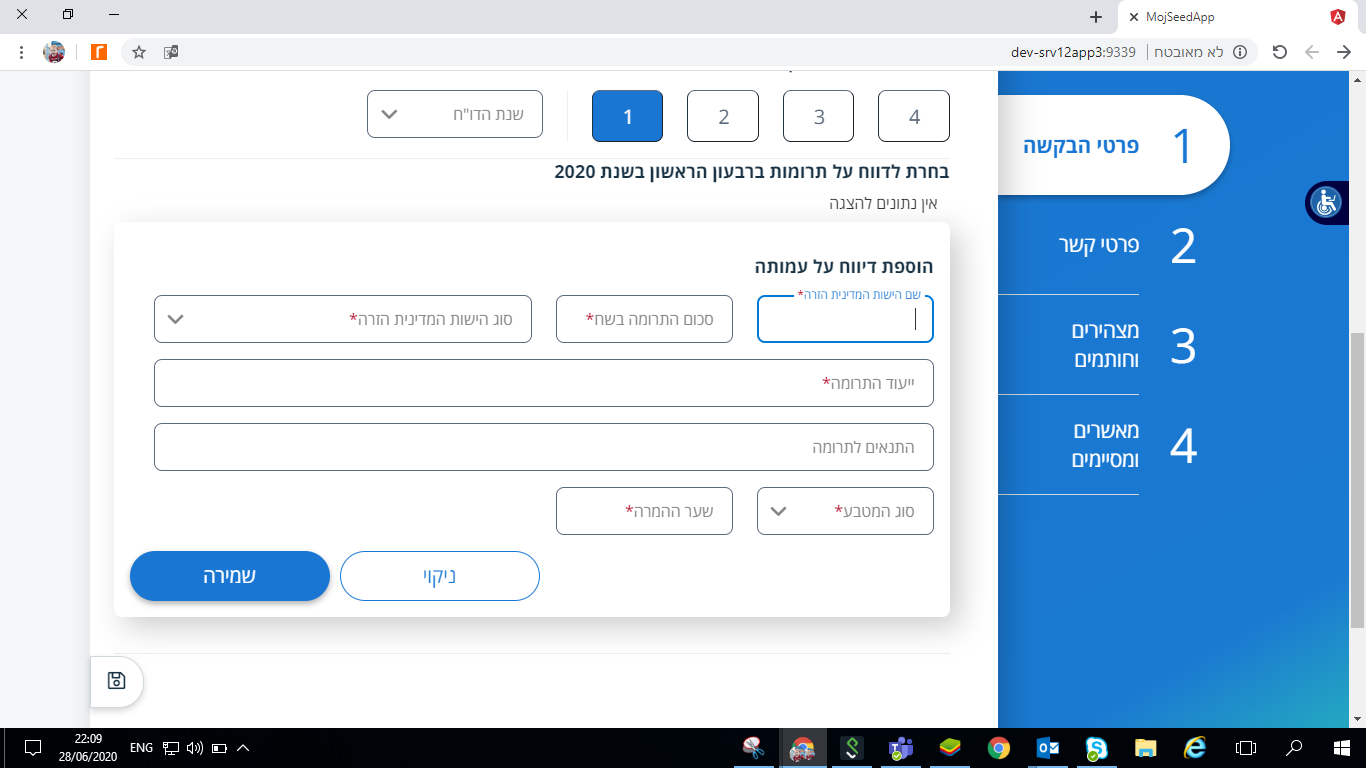
יש לפתח אפליקציית CLIENT שמממשת את היכולות הבאות.

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 תרומה**

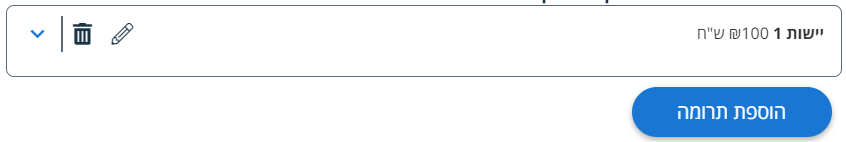
* יש לפתח כרטיסיה שתכלול את השדות הבאים, כפי שנראה בתמונה
* השדות המסומנים בכוכבית הם שדות חובה, וצריך לממש בהתאם.
* יש להוסיף ולידציה שתאפשר בשם היישות במדינית הזרה – הזנה של תווים בעברית ובאנגלית בלבד.
* בשדה סכום תרומה – יש לאפשר הזנת מספר בלבד. הולידציה תאפשר הזנה של מספר עשרוני.
* כפתור שמירה יעביר את הנתונים לרשימה שמופיעה בסעיף 2.



1. **רשימת תרומות**

הרשימה תכיל כרטיסיה שתציג רק את השדות: שם היישות המדינית הרה, וסכום התרומה.

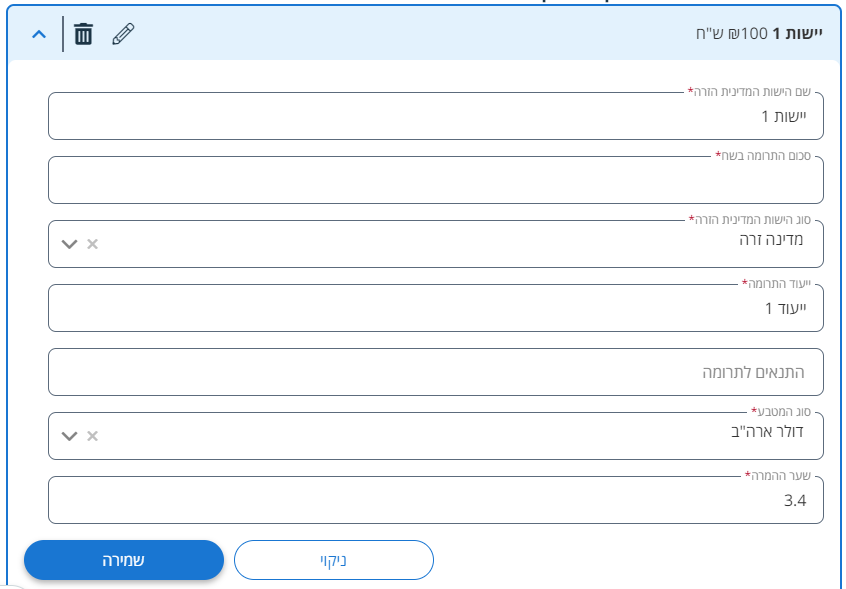
כפתור הוספת תרומה יפתח את הכרטיסיה שבסעיף 1, ויאפשר הוספת פריטים לרשימה



1. **עריכת תרומה**

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

שימו לב שהכרטיסיה משתנה ממצב תצוגה למצב עריכה.



**חלק ב – צד Server**

1. יש לפתח שירות Web API שיקלוט את פרטי התרומה למטרת שמירת ב – DB
2. יש לממש את ולידציות הבאות בצד שרת:
   1. שדות חובה – על השדות המסומנים בכוכבית
   2. שדות טקסטואלים – תווים בעברית ובאנגלית בלבד
3. אם סכום התרומה בש"ח גדול מ – 10000 יש לשלוח מייל לכתובת מייל לבחירתכם (אפשר למייל האישי)
4. אין צורך לממש בסיס נתונים ולכתוב את פונקציית השמירה ב – DB עצמו, אלא רק להכין חתימת פונקציה שתתאים לעדכון DB