My Parliament

## על הפרויקט

### מה זה My Parliament?

My Parliament הוא תוסף chrome ל- google meet המאפשר שיח מסודר והגון על ידי הגבלת רשות הדיבור והצגה רק למי שתורו לדבר.

הגדרות

* מנהל המפגש (host) - מי שיצר את ה- google meet או עברה לידו הסמכות של host.
* מוזמן - כל מי שבמפגש ואינו מנהל המפגש. כל מי שהמנהל הזמין לפגישה
* משתתף (participant)- כל מי שנמצא במפגש: מנהל המפגש ואורחיו
* משתתף משתמש (user participant)- אתה, המשתתף לפגישה שהתוסף רץ על המחשב שלו
* דיון - שיחה המנוהלת לפי חוקי התוסף

### מה הבעיה?

* מאוד קשה לנהל שיח הגון, במיוחד בישראל. אנו רואים במיוחד איך במקומות הכי חשובים כמו בכנסת צועקים זה על זה, מדברים זה מעל זה. זה מקשה על הצדדים לשמוע זה את זה ומונע שיחה פורה ומקדמת. **בשביל להגיע להחלטות, צריך שיח מסודר!**
* שיח חברתי בו מדברים זה מעל זה לא רק מקשה על המשתתפים לשמוע מה אומרים אלא גם מקשה על המשתתפים שפחות דומיננטים למצוא את זמנם לדבר ולתרום לשיח(מדבר מניסיון), במיוחד כאשר יש   
  פינג פונג בין שתי אנשים, מה שגורם לאחרים להרגיש מחוץ לשיחה.
* יש צורך בהנגשת שיח חברתי לאנשים שדיבור של כמה במקביל יכול להציף חושית. כדי שגם הם יוכלו להשתתף

### מה הפתרון?

* הרעיון של "כרית הדיבור", שבשיח רק למי שבבעלותו כרית הדיבור יכול לדבר. אם מישהו אחר רוצה את כרית הדיבור, הוא צריך להיות מועבר אליו. מושג זה קיים כבר במגוון סוגים של שיחות  
  **בחיים האמיתיים זה יותר קשה לשמור על סדר בשיח אבל במפגש מקוון יש לנו את הכוח הזה!**
* איך זה יעבוד? השתקת כל המשתתפים חוץ מהמנהל (יש רק למנהל זכות לדבר מעל המשתתף או להעביר את הרשות למישהו אחר) ומי שיש לו את רשות הדיבור שניתנה לו על ידי מקומו בתור או על ידי המנהל.
* התוסף ישמור תור (Queue) למפגש שהאורחים יכולים להוסיף את עצמם אליו.
  + מי שרוצה להיות הבא בתור ילחץ על הכפתור שיוסיף אותו לתור להמתין.
  + אחרי שלחץ על הכפתור, הוא ישתנה בכפתור "ביטול רשות דיבור": להוציא את מקומו מהתור או כאשר כבר תורו ורוצה להעביר את רשות הדיבור לבא בתור

### פונטציאל

אני רואה שימוש רב לתוסף זה במגוון מקרים: בשיעורים מקוונים של כיתה בו המורה לא יצטרך להאבק בלהשקיט את התלמידים המחוברים, בשיחות עם נושאים מעוררי מחלוקת, בישיבות של ועד מנהל או במפגשים חברתיים ופעילויות מקוונות.

### מה מקור ההשראה

ההשראה באה אלי כאשר לקחתי חלק ביוזמה משותפת של כמה ארגונים לארגן מפגשים מקוונים חברתיים בתקופת המלחמה.   
המיזם נקרא "המועדון החברתי הוירטואלי" והוא נועד לשפר את מצב רוחם של ישראלים שבטח חרדיים ובודדים מאוד בזמן קשה זה.  
  
רוח השיח במפגשים אלו אומץ מחוקי השיח של אחד מהארגונים מהם באה ההשראה שלי לכתוב תוסף שיישם את עקרונותיהם אוטומטית ולא ידנית.

החוק המקורי לנוהל שיח של אס"י:

"מדברים לפי תור: כדי שכולם יוכלו לשמוע את כולם, משתתפים מדברים לפי תור, אחרי שקיבלו

באופן מפורש את רשות הדיבור ממנהל הפגישה.   
מנהל הפגישה יוצג בתחילת הפגישה.   
אפשר לבקש את רשות הדיבור ע"י הפעלת סימן "הרמת יד" בכלי הפגישה, או ע"י כתיבה בצ'ט הפגישה.

כמו-כן אפשר להביע דעה (בלי לחכות לתור לדבר) בצ'אט הפגישה."

### יכולות(features):

#### הצבת הגבלת זמן על דובר על ידי מנהל

#### עריכת הצבעה

מנהל המפגש יכול ליצור טופס שיופיע לכל המשתתפים. בו או יבחרו מתוך כמה אפשרויות(אולי בעתיד לאפשר multiple choice) או שימלאו טקסט.

##### בניית הטופס

טופס הוא בנוי מטקסט שאלה ומאזור הקלט. הוא בוחר אם אזור הקלט יכול להיות או תיבת טקסט, או רשימה של אפשרויות

למנהל יהיה אפשרות להוסיף הגבלת זמן

ניתן יהיה להוסיף את האפשרות של הגבלת זמן בחירה. אחרי שהזמן יעבור הטופס נסגר ויראה התוצאה.

#### האלגוריתם שקובע למי מגיע רשות הדיבור

##### להגיב לדובר הנוכחי

אם מישהו רוצה להגיב לדברים של בעל רשות הדיבור הנוכחי.   
לא הייתי רוצה לאלץ אותם לחכות לתורם אחרי שכל שאר התור ידבר והדיון ינטה לנושא אחר ודבריו כבר לא יהיו רלוונטים. זה יקשה על ניהול דיון פורה, במיוחד כאשר שני צדדים רוצים לנהל דו שיח על נושא מסוים.   
לכן ליד כפתור הצטרפות לתור יהיה כפתור "הגב לדובר הנוכחי" שיאפשר לו לעקוף את כולם למקום השני אחרי הדובר הנוכחי.

###### בעיות

אני רוצה למנוע מאנשים לנצל כפתור זה לרעה. אחרת המפגש יהיה כאוטי, כמו שאנו הישראלים מחכים בתור ומנסים כל הזמן לעקוף זה את זה.

* בין אם יהיה פינג פונג בין שתי אנשים שידברו זה בין זה שעות
* מישהו כל הזמן ילחץ על זה. ברגע שיעבור תורו לדובר הבא, הוא ילחץ על כפתור "הגב לדובר הנוכחי" ואז יוקפץ מסוף התור להגיב מיד לדובר שבא אחריו.
* יינסו להשתיק מישהו ספציפי בכך שכל פעם שעומד להגיע תורו, מישהו אחר יעקוף אותו
* צריך לבחון מה קורה כאשר כמה אנשים רוצים להגיב לאותו דובר וילחצו על הכפתור

###### פתרונות אפשריים

* כמובן אדאג לפחות שכאשר תורו של דובר מסתיים, מייד רשות הדיבור תעבור לבא בתור, כך שלא יספיק ללחוץ על הכפתור לעקוף את כולם כדי "להגיב לעצמו"
* לכל משתמש מותר ללחוץ על כפתור זה רק כל 20 דקות(לברר מה צריך להיות הגבלת הזמן)

##### לעזור למי שלא הגיב

אני אוסיף בגרסאות עתידיות אפשרות לכפתור שהמנהל יכול ללחוץ כדי לעזור לאורחים שעוד לא דיברו לתרום לשיחה. הפעלת אפשרות זו תראה למנהל מי שהגיבו הכי פחות ותראה הודעה לו ולאורחים הכי פחות פעילים להציע לצרף אותם לתור.  
  
אני אוסיף למבנה נתונים תור(Queue) נוסף של כל שמות האורחים וכאשר למישהו הגיע תורו, שמו יוצא מהתור ומתווסף חזרה לסוף התור.   
ככה אחרי זמן מה, מי שיהיה בראש התור יהיו אלו שהכי פחות דיברו.   
חצי שעה אחרי שהדיון מתחיל, הודעה תצוץ לאורח הכי פחות פעיל המציעה לו להירשם לתור. בנוסף לכך הודעה תצוץ למנהל המפגש המציעה לו להעביר את רשות הדיבור לאלו שלא יצא להם עוד לדבר. הוא יוכל לראות אותם מהתור המודד פעילות אורחים.

## איך זה יעבוד

ההרחבה תרוץ על כל משתתף כולל המנהל(צריך לברר בניסוי האם זה נכון? האם ההרחבה תרוץ על מחשבים של משתתפים או שעליהם להוריד אותה? האם היא עובדת על טלפונים ניידים?)

### מבנה הנתונים

התורים יישמרו ב-firebase (אני משתמש ב- cloud firestore) .

אני שומר את המידע במבנה נתונים מרוכז בענן ולא לוקלית על מחשבי המשתתפים או רק על מחשב המנהל, משום שהמידע צריך להיות נגיש לקריאה ושינוי מכל מחשבי משתתפי המפגש!   
אם משתתף הצטרף לתור, אצטרך לזהות איזה מחשב מאחסן את מבנה הנתונים, או יותר גרוע, לעדכן בכל מחשב על כל שינוי!  
יותר מכך, מידע השמור במחשב הלקוחות: chrome storage ו cookies נגישים אליהם כי המידע יושב על המחשב שלהם ומידע יכול להישחת בכוונה או לא בכוונה.   
לדוגמא אורח שהתור שמור במחשב שלו יכול לנצל את זה שתמיד יהיה לו את רשות הדיבור.   
זו בעיה ביטחונית רצינית!

הוא צריך להיות במקום אחד מאובטח שנגיש מכל מחשב.  
  
Firebase הוא חינמי למידע קטן ומאוד בקלות מתחבר ל- Front כי היא BAAS, אז לא צריך לכתוב backend מיוחד להתחבר לשרתי DB כמו MySQL.

כל מפגש יהיה document ב- collection הנקרא "Meetings". כאשר הID של כל Document יהיה ה- ID של המפגש.

כאשר אחד מה- hosts יילחץ על "התחל שיחה", יווצר document חדש של מפגש זה בו ישמרו:

* ID של המפגש יהיה הID של המסמך(משום שכל ID של מפגש הוא ייחודי לפי גוגל, אני לא רואה למה לא שמסמך של מפגש יצויין לפי ID המפגש)
* UUID של מנחה השיחה
* [Time To Live](https://firebase.google.com/docs/firestore/ttl?hl=en) שהמסמך יימחק אחרי 24 שעות
* התור- יהיה מערך של אימיילים/שמות משתמש(לברר עם יש IDs למשתתפים.
* זמן תחילת הדיון
* Map של שמות אורחים והזמן האחרון שלחצו: אם מישהו לוחץ בפעם הראשונה, הוא נוסף לMap עם השעה שבה לחץ.   
  אם הוא לוחץ שוב וההפרש מהפעם הקודמת שלחץ לעכשיו קטן מ 20 דקות, הוא לא יכול לעקוף. אבל אם גדול מ 20 דקות, אז יהיה לו את האפשרות ללחוץ, והזמן שלחץ שנית יחליף את הזמן שלחץ קודם

במחשב של כל משתמש יישמר בCASHE האם יש לו את הרשות לדיבור(בוליאני, כן או לא) ואת התור.

ואני אגדיר מאזין לקובץ של המפגש (Onsnapshot), אשר יעדכן את הCASHE כל פעם שהמסמך משתנה.

יתרונות:

1. יעלה את מהירות וביצוע התוסף: אם משתמש ילחץ ספאם על כפתור הMUTE וכל לחיצה התוסף יישלח בקשה לFIRESTORE לבדוק אם יש לו את הרשות דיבור, זה לא רק יאט את ביצוע המחשבים אלא יכול להיות עשרות קריאות לשווא ובשירות כמו FIREBASE בו קריאות עולות כסף, כדאי לחסוף
2. יהיה ניתן בPOPUP לשלוף את התור באופן מיידי לUI שלו מבלי כל פעם שייפתח את הPOPUP ייבדק

חסרונות:

1. משתמש יוכל לערוך את הCAHE שלו ולהגדיר את האם יש לו רשות דיבור לTRUE

אבל! אם נגדיר שבONSNAPSHOT, הוא רק ייקרא לפונקציה על הענן שרק בודקת האם יש לו את הרשות לדבר או לא, ורק יחזיר אמת או שקר. זה מאט את הבעיה. אנחנו לא רוצים שהUUID של המנחה יהיה שמור במחשב של כולם כי זה יהיה צרה יותר גדולה.

אפילו אם שינה לTRUE, ברגע שיהיה איזהשהוא שינוי במסמך, התוסף יעדכן את הCOOKIE חזרה לFALSE

יותר מכך, ניתן לשמור את כל מידע של העוגייה במחרוזת אחת, שאותה נוכל להצפין.

בעתיד אעשה SYMATRIC ENCRYPTION שיהיה מפתח שמור במבנה נתונים, ואחד שמור בתוסף. המבנה נתונים יצפין את המידע ששולח, ואת הצופן יישמור בCOOKIE.

כרגע רק אצפין בBASE64.

כל פעם שהמבנה נתונים שומר

### Participant id

התוסף יוצר לכל משתתף(כולל למנחה) UUID שהוא שומר לוקלית בדפדפן.

החלטתי סופית שהוא ייצור UUID ייחודי עבור כל משתתף לכל פגישה חדשה:

אני אשמור בcookie המתוייג לURL של google meet עם meeting id אחריו.

אם הוא מזהה שאין COOKIE לאותה פגישה הוא ייצור COOKIE ספציפית לURL של אותה פגישה וייצור לאותו משתמש UUID שיכניס אם הפרטים שקרא מהDOM.

אם הוא מזהה שיש כבר COOKIE לאותו משתמש המזוהה עם אותה פגישה, אז ייבדוק האם הפרטים של המשתתף(שלא GOOGLE MEET ID שמשתנה כל הזמן) מתאימים לפרטים בעוגייה. אם לא ייעדכן.

בכל מקרה הוא יישלוף את הUUID ששמור בCOOKIE לפגישה.

1. זה יפחית את הסיכויים שיהיו שתי משתמשים עם UUID זהה בגלל שהUUID יימחקו וישומשו למשתתפים אחרים. אם לכל משתתף עלי אדמות יהיה UUID קבוע, זה אפשרי, אבל .
2. הזיהוי זה עבור המשתתף בפגישה הזו ספציפית. זה לא הגיוני ליצור UUID למחשב. זה לא ברור בדיוק מה זה מאפייו
3. רוב המידע משתנה ואקסלוסיבי לאותה פגישה, השם, הGOOGLEID. אז אין סיבה לשמור מידע קבוע שכל המידע חוץ מהUUID משתה כל הזמן. יכול להיות שמשתמש ייתחבר מחשבון אחר עם שם אחר לפגישה אחרת. אז השם יהיה שונה.
4. אפשר להגדיר COOKIE כSecure מה שנותן אבטחה יותר מlocalstorage /sessionstorage
5. הcookie הוא אקסלוסיבי לURL מסוים שנשמע נכון מבחינה רעיונית. אפשר לגשת לפרטי המשתתף רק בטאבים של הפגישה שלו.
6. ניתן להגדיר בקלות תאריך תפוגה לcookie. אני אגדיר אותו ליום כי אין פגישות שיותר מ24 שעות! ככה שמשתמש עוזב, לא אצטרך לעשות שום CLEANUP, המידע יימחק אוטומטית.
7. אני לא בוחר בsessionstorage כי אם משתמש בטעות סגר את חלון הפגישה, הפרטים שלו יימחקו! אז שייחזור, ייצרו לו ID חדש. אני רוצה שעד סוף הפגישה הCOOKIE יישאר אותו דבר.

התוסף ישלוף את פרטי האורח מרשימת האורחים הקיימת מצד ימין(בכל מפגש ליד השם שלך ברשימה יופיע

"(you/אתה)":

1. השם משתמש
2. הgoogle meet ID של המשתתף
3. האם המשתמש הוא המנחה או לא(ברשימה מצוין מתחת לשם המשתתתף המנחה שהוא המנחה. פרט זה לא יישמר לוקלית כדי למנוע ממשתמשים להגדיר את עצמם כמנחים!!)

חיסרון של זה שאיך שהפרטים נשלפים דורש פתיחה אוטומטית של הרשימה כדי להגיע לאלמנטים, מה שיכול להזיק לחוות המשתמש, לכן מעדיף לשמור שם כדי שיהיה נגיש לbackground וcontent script ושישלוף מהtab כל פעם ולא מהDOM.

למרות זאת, כל פעם שייפתח טאב חדש של מפגש או העמוד ייפתח מחדש, הוא ייקרא מחדש. הוא רק קורה ברגע שהDOM נטען כי כך הכי בטוח מdom manipulation.

בעתיד לחשוב על דרך יותר בטוחה לשמור את פרטי המשתמש.

אם המשתתף נמצא בטאב של google meet(איך הPOPUP יידע אם טאב הוא של גוגל מיט?) יציג בפופ אפ את תור הדיון (רק את ה 10 הראשונים בתור או אפשרות לscroll אם הדיון כבר התחיל, אם לא אז בפופ אפ יהיה ריק )

לפי הUUID בCOOKIE התוסף יידע האם המשתתף הוא מנחה הדיון או בעל הרשות דיבור כי רק לשניהם יש את רשות הדיבור מכל משתתפי הפגישה.

איך שהשירות שלי יידע בדרך בטוחה האם המשתמש הוא מנחה הדיון הוא שהUUID של המנחים יישמר במבנה הנתונים וייבקש כל פעם מהמבנה נתונים לאמת האם המשתמש שלו הוא מנחה. לחשוב על דרך יותר בטוחה לעשות את זה.

ולגבי הזמנים שבהם משתמש לחץ על הכפתור דלג, כרגע זה יהיה COOKIE נוסף שיישמר על מחשב כל משתמש כדי לחסוך זיכרון מהמבנה נתונים

אם הוא משתתף, אז יתווסף האזנה לאירוע בו השתנה כפתור ה mute. אם אין להם את הרשות לדיבור או אינם הhost, אז התוסף ישתיק אותם חזרה.

מעבר לכך, התוסף יוסיף למשתתף את הכפתור ל- "הצטרף לתור" ו "[הגב לדובר הנוכחי](#_beep2blbmjpp)".

אם לוחצים עליו ועוד לא הגיע תורם, הכפתור יתחלף בכפתור להסרה מהתור. אם יש להם את הרשות לדיבור הכפתור יהיה לסיים את תורם. התוסף יזהה שהם לחצו על הכפתור אפילו אחרי F5 כי הלוגיקה אומרת, כל עוד ה ID שלהם איפה שהוא בתור, בין אם ממתין או הדובר, להציג את הכפתורים המבטלים ולא המוסיפים.

### מנחה השיחה:

נכון שבGOOGLE MEET מותר שיהיו כמה מנחים אבל אחרי מחשבה רבה החלטתי שעדיף שמתוך כל המנחים רק לאחד יהיה את ההרשות לשלוט בדיון.

אם המערכת זיהתה שאין עוד דיון, היא תיתן לכל מנחה את האפשרות להתחיל את הדיון. המנחה הראשון שיילחץ על הכפתור הוא יהיה מנחה השיחה, יווצר DOCUMENT בFIRESTORE עם ה ID של המפגש ובו רשום הUUID של משתתף זה כמנחה השיחה.

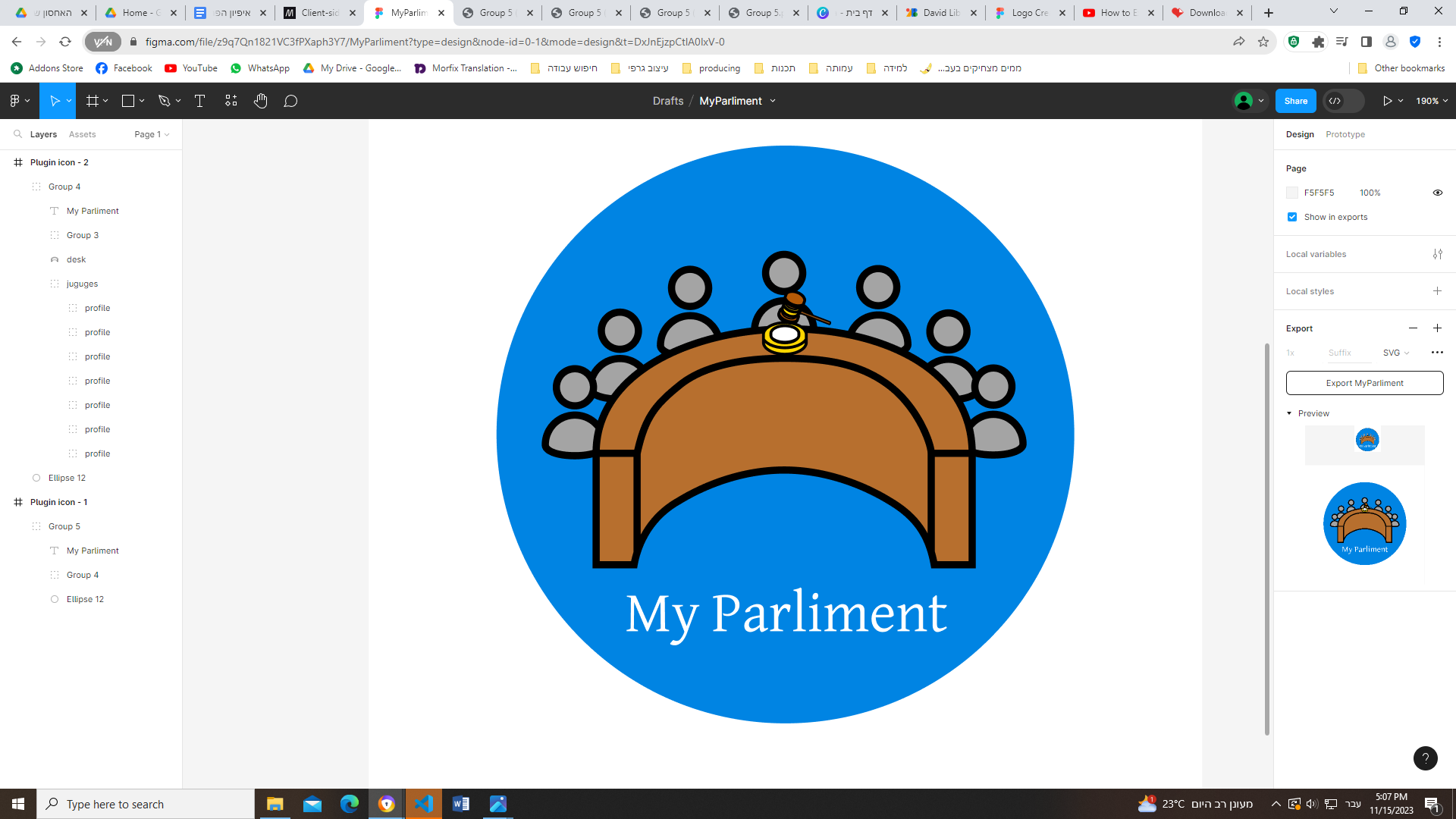
ברגע שקיים דיון האפשרות לפתור דיון או תיכבה או לא תעבוד למנחים אחרים והמנחה הדיון יתווסף ברשימת משתתפים לכל משתתף כפתור להעביר למי שהוא רוצה את רשות הדיבור

אם הוא מנהל (לזהות לפי הכפתור host controls) בפופ אפ יהיה כפתור להתחיל את השיח, שייצור את התור במבנה הנתונים(ולפופ אפ יתווסף בעתיד כפתור ליצור סקר).  
בנוסף לכך, יתווסף לכפתורי הגוגל מיט שלו כפתור שמאפשר לו להעביר את רשות הדיבור למי שהוא רוצה.   
הכפתור יוסיף את המשתתף לראש התור (אם למישהו יש את רשות הדיבור, הוא יעבור אל החדש אפילו אם עוד לא סיים).

## אפיון עיצובי

הייתי רוצה להשתמש בפונט David Libre שכתובים בו מסמכים מקצועיים: ממסמכים של בית משפט עד להצעות חוק של הכנסת!

עיצבתי כבר לוגו לתוסף ה chrome ב- Figma:



## צעדים למימוש הפרויקט

### מחקר

1. מחקר וניסוי-
   1. למצוא תשובה על נעלמים(מה שסימנתי באדום) ועל הנעלמים הללו:
      1. נעלמים על google meet
         1. האם יש קודים אשר עם סימנים מיוחדים? ברור שלא יהיה קו אמצעי באחד משלושת המחרוזות המופרדות על ידי קו אמצעי אבל לגבי פלוס, מינוס וכו
         2. איך לזהות אלמנטים חשובים שנשנה אחר כך(שמות מחלקה וכו)
            1. ה toolbars שנוסיף אליו אחר כך כפתורים

כפתור ההשתקה

* + - 1. איך כמנהל לשים תוסף על מפגש של google meet?
      2. ~~איך לדמות שאורח לוחץ CTRL +D כדי להשתיק אורחים~~- [פתרון](https://github.com/jasonketola/Google-Meet-Mute-Toggler/blob/master/sendKeypress.js) צריך להיות keycode, code cancelable:false
      3. איך לוודע שהמסך הוא מסך שכבר חלה שיחה ולא מסך ההצטרפות שבעל אותו URL?
         1. אולי לפי משתנה הסטטוס שאומר "complete
         2. או פשוט לבדוק בצד של היוצר תוכן שהאלנמטים של השיח קיימים? document.querySelector('button[aria-label~="Microphone"]')
      4. האם כל אורח לפגישה צריך להוריד את התוסף כדי שיעבוד לו? לנסות עם תוספים ל google meet שקיימים ולבדוק את השפעתם.
      5. האם תוספי chrome עובדים למשתמשים ניידים(פחות דחוף)
      6. איך להחדיר כפתורים ל toolbar של google meet
         1. כרגע אני שמתי לב לתבנית שבכפתור ההשקט בתכונת הaria-text בכל מצב כפתור המיקרופון(בין אם מופעל, מכובה, או שחסום בידי הדפדפן), מופיעה המילה microphone בטקסט. האם זה בכל מצב, לברר מצבים אחרים של המיקרופון! לברר האם בכל מצב ניתן למצוא את הכפתור
         2. לברר האם לtoolbar השמאל הטא תמיד מחלקה עם classname של "lefkc", בכל מקרה, ככה אני משיג את השליטה
      7. לבדוק לגבי extensions כמו push to talk אשר מפעילות את הכפתור עם space?
      8. הבדל בין UI של מנהל ואורח
      9. מה ה ID של אורחים? מניסוי ראיתי שניתן להתחבר ללא חשבון אימייל, אז איך אוודע מאיזה אורח רץ התוסף? צריך להיות ייחודי וקבוע לכל אורח, לפי הIDS שGOOGLE MEET נותן לאורחים בעצמו.
      10. מה אם לאורח יש שתי פגישות MEET פתוחות בו זמנית? זה לא ייקרה ולא הגיוני!
      11. ~~איך לזהות שהתוסף רץ אצל המנהל?~~ פשוט, למנהלים יש לאלמנט שנקרא host controls. לבדוק אם הוא קיים בDOM או לא כדי לדעת אם הוא המנהל הוא לא
      12. איך לזהות שמשתמש הסיר את השתיקה מעצמו, בין אם לחץ או לחץ על המקלדת ctrl+D,
          1. אפשר להקשיב לאירוע של לחיצה או למקלדת
          2. אפשר דרך mutation observer לזהות שהattributes ישתנו
          3. או פשוט אפשר פשוט כאשר התוסף מתחיל לאורח, לעשות disabled לכפתורים ובנוסף לכך לזהות

g = new MutationObserver(entries =>{

let entFil = entries.filter(ent=>ent.attributeName === "aria-pressed");

if(entFil.length>0)console.log(entFil)})

g.observe($0, {attributes: true})

* + - * 1. יש את הattribute mute
    1. נעלמים על firebase
       1. איך לחבר בין התוסף לfirebase? והאם הם יתחברו מכל מקום?
       2. להתנסות בשמירת נתונים במסמכים כאלה. איך זה עובד
       3. האם הכתיבות לfirebase הם אסכרוניות ואיך למנוע שכאשר כמה אורחים שולחים בקשות בו זמנית שסדר השליחה יישמר כאשר הוספה מתעכבת ולא יתערבבו
       4. איך אפשר להוסיף מסמך חדש לfirebase
       5. איך אני מעדכן תכונות מסויימות
       6. איך אני מנהל תור מתוך מבנה נתונים? אני יכול רק לשמור אובייקט או רשימה. האם יהיה עלי לשלוף את הרשימה כל פעם לאובייקט תור שישנה אותו ויחזיר חזרה. אני רוצה להוסיף איבר לתור מבלי שאצטרך לשלוף ולשים אותו חזרה.   
          לחקור על cloud functions כמו הוספה לרשימה שתתאים לחוקים של תור ותאפשר ניהול של תור במבנה נתונים (First In, FIrst Out- FIFO)
    2. נעלמים על chrome
       1. איך אני מעלה פרוייקט לתוסף
       2. איך אני מעדכן אותו מבלי להעלות כל פעם מחדש?
       3. איך המסמכים עובדים? הbackground.js וpopup.js
       4. באיזה מקרי קצה התוסף לא יעבוד? יהיה אליו לזהות שהמשתמש נמצא על חלון google meet.

### פיתוח

1. יצירת אבטיפוס
   1. לפתח תוסף שרק ישקיט את כולם חוץ מהמנהל ו ID של משתמש שהוא hardcoded.
   2. יצירת פרויקט עם כל המסמכים החשובים לתוסף chrome
   3. יצירת ה firebase ולחבר לפרויקט
   4. לבדוק שההשפעה חלה לא משנה מה, אם לוחץ F5, פותח בטאב חדש, התוסף עוד קיים
2. להוסיף אינטראקציה!
   1. להוסיף כפתור מהמשתמשים לtoolbar שלהם במפגש ולבדוק שרואים אותו לא משנה מה.
   2. שהכפתור ידפיס את ה ID שלהם
3. אינטגרציה של המבנה נתונים
   1. ליצור את מבנה הנתונים firebase ולחסק
   2. להוסיף UI בסיסי לpopup שבכל חלון אחר שלא meet יראה מסר
   3. להוסיף כפתור למנהל בpopup להתחיל את הדיון: ליצור מסמך חדש ב DB(כפתור התחלה הדיון) שיהיה לו את ה ID של המפגש ושמור בו רק הזמן בו התחיל הדיוןׁׁ(שבו נלחץ הכפתור)
   4. שהכפתור שהוספנו למשתמשים פשוט יוסיף את ה ID שלהם לרשימה.
   5. להוסיף שהכפתור משתנה שמשתמשים לא יוכלו להוסיף את ה ID שלהם יותר מפעם אחת. כל עוד השם שלהם ברשימה הם לא יכולים לשנות.
   6. להפוך את הרשימה לתור
   7. להוסיף בpop up שיהיה אפשר לראות את התור
   8. בסוף, לשנות את ID המשתמש אם רשות הדיבור מהקבוע לשם שבראש התור
4. הוספת יכולות חסרות
   1. להוסיף למנהל את היכולת להוסיף רשות דיבור לאחרים
   2. להוסיף את יכולת העקיפה
      1. את הכפתור "הגב לדובר" לאורחים
      2. לתכנת את הפונקציונליות שלו שישפיע על התור במבנה הנתונים
      3. להוסיף את ההגבלה שכל 20 דקות(להעתיק קוד שהוסיף לרשימה IDS של אורחים מהשלב הקודם

### 

### מבנה הפרויקט

#### Content Script

התסריט שפועל במסגרת DOM של דף המפגש.

הוא אחראי על

* שליפת נתונים על המשתמש מהHTML
  + האם הוא אחד ממנחי השיחה?(HOST)
  + מה השם של המשתתף בשיחה(אם יש לו אימייל זה יהיה השם פרטי ומשפחה שהגדיר בחשבון
  + הוא ייזהה לפי הסימן MUTE ליד השם שלו הרשימה האם הוא משותק או לא.
  + החלטתי במקום לבדוק לשינויים באלמנט כפתור ההשתקה עצמו להאזין לסימן שהמשתתף מושתק ברשימת משתתפים בגלל שהאלמנט הזה פחות משתנה ושהאזנתי לכפתור ההשתקה הMUTATION OBSERVER זיהה 3 שינויים ויותר לכל לחיצה. ועוד מזהה רק שינוי אחד כאשר הסימן הופך ממושתק ללא מושתק

#### Background

התסריט ברקע

אחראי על כל עניין הCOOKIES: האחרים שולחים לו בקשה ליצור ולגשת לCOOKIES(כי רק לBACKGROUND יש גישה לCOOKIES API של CHROME וגם זה נכון יותר מבחינת מבנה שהBACKEND ייצור את הUUIDS)

שולף מהמבנה נתונים

מי המנהל

את התור

כל בקשה למבנה נתונים תטופל בBACKGROUND, וכל בקשה יבקשו ממנו והוא ידאג לשלוף מהCACHE אם כבר קיים.

#### Popover

רוב הUI של התוסף

ישלח בקשה לbackgroud מהמבנה נתונים לגבי פרטי המשתמש: האם הוא המנהל וכו…

### אלגוריתם

טאב חדש עם url של Google meet נפתח מה שמעורר את התוסף להחדיר לDOM ולהפעיל את תסריט התוכן אשר:

1. מחכה שאלמנט כפתור השייך לUI של פגישת MEET פתוחה יופיע בDOM. כך יודע שכבר לא ממתין להצטרפות, אלא הוא אשכרה נכנס נמצא בחלון של פגישה פתוחה
2. שולף את הנתונים של המשתמש מחלון המשתתפים
   1. אם טאב הרשימת משתתפים לא פתוח, פתח וחכה שנמצא בDOM
   2. אם לא נלחץ על הרץ שפותח את הרשימה, פתח וחכה שנמצא בDOM
   3. כעת שהאלמנטים עם מידע המשתתפים נמצאים בDOM, תמצא אלמנט הspan ממחלקה "NnTWjc" שזו מחלקה של אלמנט בו כתוב "(אתה)"/"(you) אחרי השם של המשתתף המשתמש. כך נשיג את מיקום המשתתף המשתמש ברשימה
   4. מהאלמנט הזה נווט בעץ הDOM כדי להשיג את
      1. שם המשתתף(אח קודם לאותו הורה) המשתמש בפגישה
      2. הGOOGLE MEET ID (הנמצא בdiv אב קדמון עם role="listitem)
      3. האם המשתתף המנחה או לא: למנחים ולמנחיי משנה(co host), ליד כפתור הactiviies יופיע כפתור שלא מופיע למשתמשים רגילים, עם אייקון של מנעול שקוראים לו host control. אז אני מזהה אם הכפתור הזה קיים
         1. אם הכפתור HOST CONTROLS קיים, אני אוסיף לו כפתור להתחיל דיון
   5. מבקש מהBACKGROUND את הUUID של המשתתף :
      1. מקבל מהFRONTEND את הID של הפגישה
      2. אם אין COOKIE שבה שמור הUUID של המשתמש, הBACKGROUND ייצור UUID וישמור אותו בCOOKIE עם PATH של ID המפגש (הCOOKIE עם URL ספציפי למפגש)
      3. שולח לב
      4. מעדכן את הפרטים שנשמרו קודם(אם לא יוסיף) עם הMEET ID ושם שנשלפו עכשיו מהDOM ומצטרף לפרטים את הUUID החדש ושומר את הפרטים באחסון על הדפדפן תחת המפתח "participant"(לא כולל האם הוא המנחה או לא)
   6. שולף את הפרטים מהאחסון ולברר האם יש לו את רשות הדיבור
      1. שולח את השם משתתף והUUID שלו לתסריט הרקע
         1. כרגע לתסריט הרקע שמור UUID קבוע של המנחה ולמי יש רשות דיבור אז הוא משווה את הUUID שנשלח לו ומחזיר האם הוא שווה לUUID של בעל רשות הדיבור או המנחה
      2. שולף מהDOM את כפתור הMUTE
      3. מקבל תשובה האם יש לו רשות לדבר או לא ולפי זה קובע האם כפתור הMUTE יהיה פעיל או לא(אם בעל רשות, הכפתור היה לחיץ, אם לא הכפתור יוגדר כבלתי לחיץ)
      4. אם גם אין למשתתף משתמש רשות לדבר ואינו מושתק(data-is-muted=false), אז יידמה לחיצה של CTRL+D(לבדוק האם הדימוי הזה עובד גם בMAC)
   7. מגדיר mutationObserver שיאזין לשינויים של אלמנט כפתור ההשתקה, ספציפית שינויים של ה attribute שמשקף האם המשתמש מושתק או לא: data-is-muted
      1. אם מזהה שערך מאפיין הזה השתנה והמשתמש איכשהוא הצליח לשנות את כפתור ההשתקה(בטח דרך המקלדת כי כיבוי הכפתור לא יימנע הדלקתו דרך הקיצור CRTL+D), לברר האם יש לו את רשות הדיבור(להריץ את בלוק g)
   8. אם המשתתף אכן מנחה, אז יוסיף ברשימה לכל משתתף כפתור לתת רשות דיבור

### בדיקה

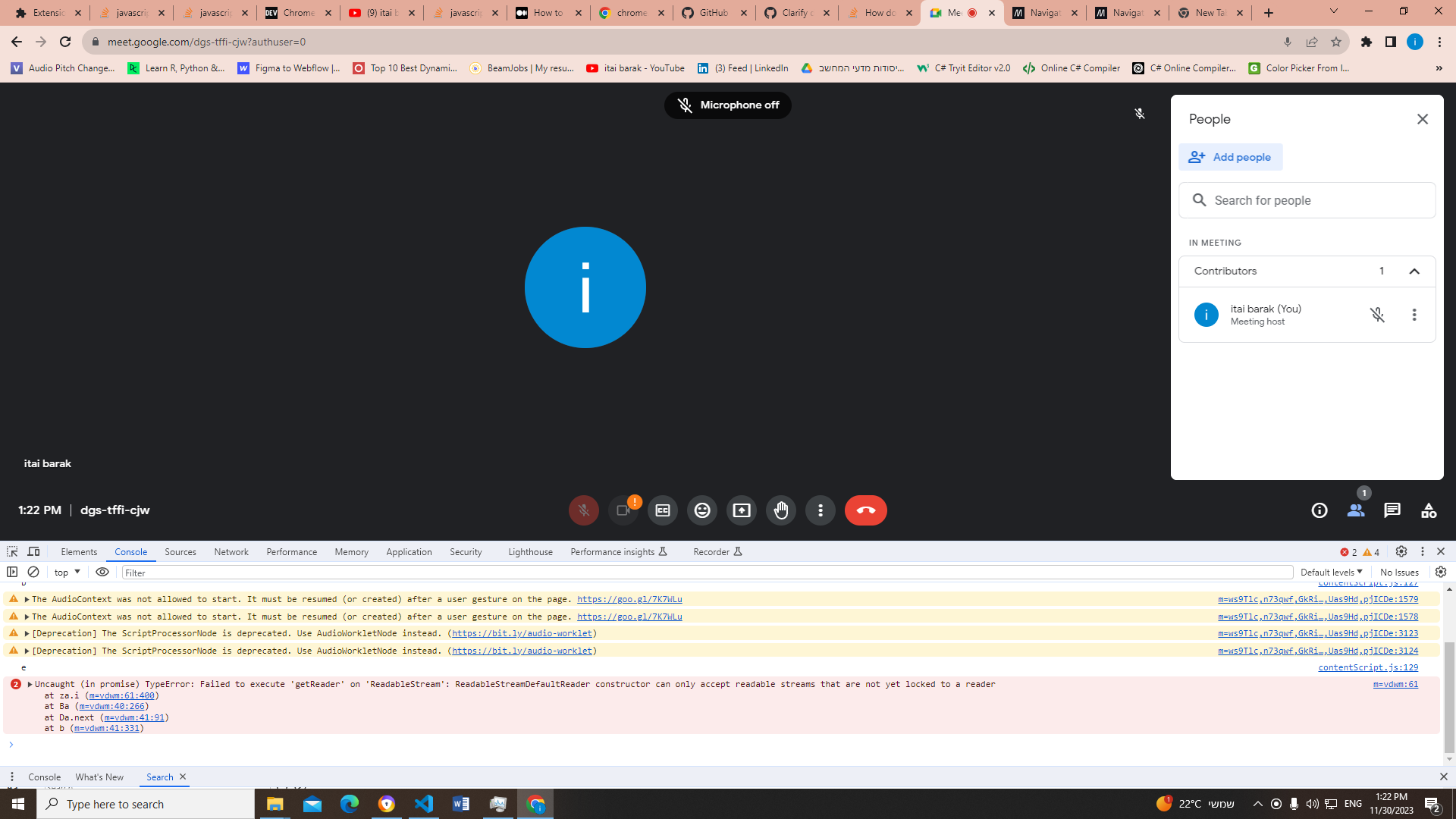
#### דרכי בדיקה

כרגע אני לא יכול לבדוק במחשבים אחרים כי לאמא יש קורונה אז לא אוכל לבדוק אצלה, אולי בלפטופ שלי או במחשב בספרייה.

במחשבים אחרים לבדוק בהדרגה, גרסה גרסה, מהאבטיפוס עד לגרסה הסופית

#### מקרים לבדוק:

מקבל שגיאה כאשר משתתף לוחץ CNTL+D ומושתק על ידי התוסף כל 4 פעמים. אולי בגלל שגם המשתמש לוחץ את הקיצור והמכונה מדמה את לחיצת הקיצור משפיע על הקורא



m=vdwm:61 Uncaught (in promise) TypeError: Failed to execute 'getReader' on 'ReadableStream': ReadableStreamDefaultReader constructor can only accept readable streams that are not yet locked to a reader

at za.i (m=vdwm:61:400)

at Ba (m=vdwm:40:266)

at Da.next (m=vdwm:41:91)

at b (m=vdwm:41:331)

#### cookies

**חשוב בעתיד לבדוק האם המשתמש ביטל COOKIES לאותו דף.**

**להרתיע את המשתמש שעליו להרשות COOKIES לGOOGLE MEET**

**לבדוק את CHROME.COOKIES לראות מה קורה ששומר עוגיות במחשב שאסר על עוגיות.**

**אם משתמש ניקה את הCOOKIES וCACHE שלו, אז צריך לבדוק כל קריאה האם הCOOKIE קיים או לא**

בעיה: אם מישהו ייתחבר לפגישת google meet בטאב אחד ובטאב אחר ייתחבר לפגישה אחרת, הוא לא יידע של מי לשמור:

פתרון: פשוט להציב את הID בURL של הcookie. ככה הוא ייצור עוגייה לא רק לURL של goole.meet אלא למפגשים נפרדים.

לשים לב איך לעשות wildcard שלא משנה מה יבוא אחרי הID של הפגישה זה לא יישנה

לשים לב כאשר משתמש מתחבר לאותה פגישה מאותו מחשב אבל מחשבונות שונים. הcookie יהיה זהה.

עוד הפעם זה מקרה קצה מאוד מטומטם. המערכת פשוט תתייחס לשתי המשתמשים כאותו משתמש. ככה זה יימנע ספאם שמישהו יכול ליצור כמה חשבונות. רק צריך לבדוק שזה ייקרה

אם משתמש מסיים פגישה ומייד מצטרף לאחת אחרת. יהיה לה ID שונה אז הURL יהיה שונה וייצור COOKIE חדש.

אם משתתש עוזב פגישה וחוזר, אז הCOOKIE יישאר זהה, כי הפרטים שלו לא השתנו

אבל מה יוצא מפגישה ואז מתחבר לאותה פגישה עם חשבון אחר?

השם שלו יהיה שונה, אבל בCOOKIE המייוחס לאותה פגישה יש את השם מהחשבון הקודם שלו.

הוא פשוט ייבדוק את הCOOKIE אם היא קיימת והאם הפרטים מתאימים. ברור שהGOOGLE ID יישתנה אבל כל פעם שיש REFRESH ייבדוק בעוגייה האם השם השתנה. זה לא משנה לי אם המשתנה יישנה בגםצ את השם שלו, כי מה שחשוב זה הUUID

רעיון יותר טוב לCOOKIES וUUID:

<https://security.stackexchange.com/questions/161425/whats-stopping-people-from-modifying-cookies-to-change-who-theyre-logged-in-as>

That's not how session cookies work. The cookie shoudn't just contain the ID of the current user since that would obviously allow an attacker to tamper with the value.

Instead, a common approach is that the web application issues a sufficiently long, random session ID to the user and internally associates this session ID with the actual user account. Since a user is then assigned a new session ID after each login, it won't be feasible to predict the session IDs of other users or manipulate user-specific values by modifying the cookie.

Generally, you shouldn't have to invent your own session management scheme. If you think about how to store a user ID in a cookie, you're probably doing it wrong. Many frameworks already provide abstractions for cookie-based sessions. E.g., PHP has built-in support for [session handling](https://secure.php.net/manual/en/book.session.php). Also have a look at the OWASP [Session Management Cheat Sheet](https://www.owasp.org/index.php/Session_Management_Cheat_Sheet) to learn about security considerations.

#### Firebase

לבדוק איך אני מונע קריאות מיותרות, לדוגמא כאשר משתמש לוחץ שוב ושוב על CTRL+D אני לא רוצה שכל פעם ישאל את המבנה נתונים, גם כדי לשפר ביצוע וגם כדי לחסוך כסף וקריאות מיותרות.

צריך לבדוק האם בברירת מחדל כפי שמבטיח [הוא חוסך קריאות כאשר המידע לא השתנה בכך ששומר את התשובה בCACHE](https://firebase.google.com/docs/firestore/manage-data/enable-offline)

לבדוק כאשר כמה מנחים לוחצים על כפתור התחלת הדיון בו זמנית

לבדוק כאשר כמה משתתפים לוחצים זה אחר זה להצטרף לתור, איך

### שיפור:

שיפור אבטחה סביב שליחת מסרים בין החלקים בתוסף:

למנוע ממשתמש להריץ chrome.runtime.sendMessage לתוסף מה שיכול לפגוע בו.

להוסיף בדיקה של ארגומנט הsender שכולל doumentID שממנו באה ההודעה כדי לוודע שהוא בא ממסמכים של התוסף

להשתמש בsen

<https://developer.chrome.com/docs/extensions/reference/runtime/#method-sendMessage>

יצירת גרסה טובה יותר

הוספת יכולות כמו התראה למי שדיבר פחות

להוסיף כפתור בצירוף של cohost שיוסיף למבנה נתונים.

הmutation observer מופעל פעמיים ומופעל פעם אחת אקסטרה בהתחלה כי אני בודק לפני שאני מתחיל את הMUTATION OBSERVER. אז אם הוא משתיק ואז אני מפעיל אותו, אז האירוע נקלט ובודק למרות שלפני שנייה בדק.

ברור לי שהוא קורא גם שינוי שהcontentscript עצמו עושה !

,זה ממש יכול להשפיע על ביצוע, לתקן שרק פעם אחת עם יש לו רשות או לא

document.querySelector("div.m3Uzve.RJRKn div.VfPpkd-aGsRMb span.NnTWjc").closest('div[role="listitem"]').querySelector('div[jscontroller="mUJV5"]')

בכל מקרה אני מצפה שחלק מהמשתמשים ינסו ללחוץ כמו שיותר מהר CTRL+D מתוך תסכול, אז אני במילא צריך למצוא דרך לחסוך READS על ידי שמירת תוצאות עם המיידע לא השתנה(onsnapshot)