**WeWatch**



**אלון צדקני – 326281367**

**שם בית ספר – הרב תחומי ב' , פתח תקווה**

**מנחה – גולן מור**

**תאריך הגשה – 16.5.2022**

****

תוכן עניינים

[תוכן עניינים 2](#_Toc103384827)

[תקציר ורציונל הפרויקט 3](#_Toc103384828)

[מבוא ורקע כללי 4](#_Toc103384829)

[מטרת הפרויקט 5](#_Toc103384830)

[דרישות מרכזיות 5](#_Toc103384831)

[תרחישי שימוש 6](#_Toc103384832)

[לו"ז לפיתוח המערכת 7](#_Toc103384833)

[ניהול סיכונים 7](#_Toc103384834)

[תיאור תחום הידע 8](#_Toc103384835)

[פירוט יכולות בצד השרת 8](#_Toc103384836)

[פירוט יכולות בצד הלקוח 10](#_Toc103384837)

[שפת התכנות וסביבת העבודה 14](#_Toc103384838)

[ניסוח וניתוח הבעיה האלגוריתמית 15](#_Toc103384839)

[תיאור אלגוריתמים קיימים 16](#_Toc103384840)

[הפתרונות הנבחרים 18](#_Toc103384841)

[(1) כיצד ניתן לצרוך סרטון משותף על האינטרנט? 18](#_Toc103384842)

[(2) איך למנוע את הפרסומות בסרטון היוטיוב? 21](#_Toc103384843)

[(3) איך לשמור את המידע על המשתמש בתוסף? 23](#_Toc103384844)

[(4) כיצד לשמור על קשר בין התוסף לשרת? 25](#_Toc103384845)

[פיתוח הפתרונות בשכלול הקוד עם שפת התכנות 27](#_Toc103384846)

[(1) כיצד ניתן לצרוך סרטון משותף על האינטרנט? 27](#_Toc103384847)

[(2) איך למנוע את הפרסומות בסרטון היוטיוב? 31](#_Toc103384848)

[(3) איך לשמור את המידע על המשתמש בתוסף? 32](#_Toc103384849)

[(4) כיצד לשמור על קשר בין התוסף לשרת? 33](#_Toc103384850)

[תיאור המודלים של מערכת התכנה 35](#_Toc103384851)

תקציר ורציונל הפרויקט

המוצר הינו תוסף בדפדפן הכרום המאפשר צפייה משותפת למספר משתתפים דרך אתר יוטיוב עצמו. המשתתפים יכולים לפתוח חדרי צפייה על כל סרטון יוטיוב הקיים באתר ובאמצעות שיתוף המספר הסידורי של החדר וסיסמתו כל משתמש המזין את הפרטים הנ"ל יכול להצטרף לחדר הצפייה ולחוות צפייה משותפת ומסונכרנת.

**בחרתי לעשות את פרויקט זה מכמה סיבות:**

בחרתי בפרויקט זה מכיוון שפרויקט העוסק בצפייה משותפת הינו נושא שמעניין ומושך אותי. כנער שבזמנו הפנוי מבלה במחשב יצא לי לצפות עם חברים בסרטונים משותפים במספר דרכים שונות כמו שיתוף מסך, ופתיחת אתר חיצוני המריץ סרטון במקביל לא יצא לי להיתקל בתוסף לכרום המאפשר למשתמשים לחוות בצפייה משותפת. עקב עובדה זאת חשבתי שפרויקט הגמר יכולה להיות הזדמנות טובה לנסות לכתוב תוסף כזה בעצמי וכך חשבתי על הרעיון לפרויקט.

בנוסף לכך רציתי לעסוק ולנצל את הזמן של עבודה על פרויקט גמר בכך שאבחר נושא שירחיב את אופקיי מעבר לנלמד בבית הספר במהלך השנים שלי במגמה ואכן יצירת תוסף כרום זה פרויקט שרחוק מאוד מאזור הנוחות שלי וכולו עוסק ב-JavaScript,HTML ו-CSS שהם דברים שפחות יצא לי להתנסות בהם אם בכלל. ראיתי הזדמנות בבחירת פרויקט זה כניסיון בתחום תוכנה חדש ובשילוב עם נושא שמעניין אותי ויש לי מוטיבציה לעבוד עליו בחריצות.

מבוא ורקע כללי

הפרויקט פועל כתוסף כרום, משמע אני פועל מתוך ה-API של chrome extensions שעובד סביב EDA (event driven architecture). התוסף בנוי מכמה חלקים שונים: (1) ***manifest.json***שאחראי להצגת ה-API של תוספי כרום על מבנה האפליקציה והשימושים המרכזיים.  
(2) ***background.js*** שמתקשר עם השרת ,מחזיק מידע על המשתמש ושם מתופעל כל האירועים האפשריים הנכללים באפליקציה בזמן גלישת המשתמש בכרום. (3) ***content.js*** שאחראי על תפעול החדר הצפייה, ובעצם יושב על ה-html של יוטיוב על מנת להוציא מידע על תפעול הסרטון ע"י המשתמש, גילוי וחסימת פרסומות המפריעים למהלך הצפייה ובנוסף לכך יוצר תקשורת מלאה עם ה-background.js ומעדכן אותו על מנת שהמידע הנחוץ יועבר לשאר חברי חדר הצפייה ע"י השרת. (4) ***popups.html*** למיניהם שאחראים על ייצוג מידע נחוץ למשתמש ונתינת אופציות למשתמש להצטרף לחדר צפייה, ליצור חדר צפייה חדש ולהתעדכן במצב חדר הצפייה בין אם זה לראות כמה צופים יש בחדר, ומה פרטי החדר (ID וסיסמא). (5) ***popups.js*** שאחראים על קישור בין המידע המוצג ב-html למידע הקשור למשתמש הנמצא ב-*background.js*.

באמצעות כל חלקים אלו הלקוח במערכת יוכל לעקוב אחרי סטטוס המשתמש בגלישתו בכרום – באמצעות events הקיימים ב-API של תוספי כרום וכך תוכל להנגיש למשתמש בהתאם אופציות לתפעול האפליקציה דרך פופ -אפים שמשתנים לפי ה-URL של המשתמש ויעניקו לו אפשרויות בין אם זה לפתוח חדר חדש או להצטרף לחדר הקיים כבר במערכת. לאחר שהמשתמש בחדר צפייה "יוזרק" לתוך עמוד סרטון היוטיוב קוד שעוקב אחרי פעולות המשתמש ומעדכן בהתאם את מצבו בסרטון כמבוקש. בנוסף לכך קוד זה יאתר פרסומות המפריעות לחוויית הצפייה ותחסום אותם אוטומטית.

השרת פועל בעזרת WebSocket וכך מתקשר עם הלקוחות. הוא פועל באמצעות הרצה א-סינכרונית של פונקציה המאזינה להודעות מהלקוחות ופועלת בהתאם, בין אם זה לרשום משתמש חדש למערכת, ליצור חדר צפייה חדש או לעדכן צופים על פקודות הקורות במערכת. השרת מעניק לכל משתמש חדש שמוריד את התוסף ID ייחודי ו-username משלו ורושמת את הפרטים שלו ב-dictionaries שונים. לאחר מכן באמצעות אותו ID ייחודי הוא יכול לנהל את כל המשתמשים בתוסף ולקשר את כל המידע הנחוץ באמצעות כלפי כל משתמש לפי אותם dictionaries.

מטרת הפרויקט

המערכת תעקוב אחרי הגלישה של המשתמש ותבדוק במידה והוא צופה בסרטון יוטיוב למשתמש יינתן אופציה באמצעות פופ-אפ לפתוח חדר צפייה על סרטון זה. כאשר המשתמש פותח חדר צפייה חלון הפופ -אפ ישתנה ויכיל את המספר הסידורי של החדר וסיסמתו ובנוסף כפותח החדר יינתן לו זכויות ה-Host משמע לו יינתן ההרשאות לשליטה בסרטון, להעביר זמן ולהפעיל/לעצור את הסרטון.

אם משתמש רוצה להצטרף לחדר צפייה עליו לפתוח את הפופ-אפ של התוסף ולהזין את המספר הסידורי ואת הסיסמא, במידה והפרטים נכונים יפתח לו סרטון היוטיוב בחדר הצפייה ולו כמשתמש לא תהיה אפשרות לשלוט בסרטון, רק ה-Host ראשי להריץ את הסרטון. המערכת תעקוב אחרי פעילויות המשתמשים ותעדכן את השרת במידה ויוצא הלקוח מחדר הצפייה באמצעות סגירת החלון או מעבר ל-URL אחר.

דרישות מרכזיות

להלן מספר דרישות פונקציונליות ולא פונקציונליות על מנת למקד את עבודתי בפרויקט ויצירת רשימה של נקודות שעליהם אתייחס במיקוד ואדאג בפרט שהם יתקיימו בתוצר הסופי בצורה האפשרית והטובה ביותר.

**דרישות פונקציונאליות:**

**1.** התוכנה תתחבר לתקשורת רציפה עם הסרבר הקיים ותקנה לכל משתמש ID ייחודי.

**2.** התוכנה תזהה אם המשתמש צופה בסרטון יוטיוב ולפיו תעניק לו את ההזמנות לפתוח חדר צפייה בעל שם וסיסמא.

**3.** יתאפשר למשתמשים להצטרף לחדרי צפייה קיימים ע"י הזנת השם וסיסמתם.

**4.** תתקיים סנכרון בסרטוני היוטיוב בין צופי החדר באמצעות הרצה משותפת של פעולות על אלמנט הסרטון באתר.

**דרישות לא פונקציונאליות:**

**1.** הסרטון המשותף ירוץ בצורה חלקה ומתאומת אצל כל צופי החדר.

**2.** דפי ה-html שבפרויקט יהיו אסטטיים ,קלים לתפעול ומובנים למשתמש.

**3.** יתאפשר למשתמש חווית צפייה משתופת בצורה שתרגיש למשתמש נוחה ולא דורשת מאמץ או הכנה מראש.

תרחישי שימוש

**1.** כאשר מותקן התוכנה לראשונה:

**-** ייפתח דף הסבר על השימוש באפליקציה

**-** המשתמש יקבל ID ו-username ייחודי ע"י השרת.

**2.** כאשר המשתמש בסרטון יוטיוב ורוצה לצפות בצורה משותפת עם חבריו:

**-** יפתח לו האופציה לפתיחת חדר צפייה, בעל ID ייחודי וסיסמא.

**-** המשתמש יוכל לשתף את ה-ID והסיסמא לחבריו על מנת שיוכלו להצטרף לחדר הצפייה.

**3.** כאשר משתמש רוצה להצטרף לחדר צפייה משותפת:

**-** יינתן לו האופציה להכניס שם משתמש וסיסמא,

**->** במידה והסיסמא נכונה יפתח למשתמש חדר הצפייה באתר יוטיוב ושאר הצופים יעודכנו על הצטרפותו באמצעות שינוי רשימת הצופים.

**->** במידה והפרטים לא נכונים, המערכת תודיע על כך למשתמש ויינתן לו הזדמנות נוספת.

**4.** כאשר המשתמש רוצה לצאת מחדר הצפייה:

**-** כל שאר חברי חדר הצפייה יעודכנו על יציאת המשתמש באמצעות שינוי רשימת הצופים.

לו"ז לפיתוח המערכת

* 30.2 – מחקר על פיתוח תוספי כרום והבנה כללית של הארכיטקטורה של EDA בפיתוח התוסף, בניית "מיני פרויקטים" להבנה רחבה יותר.
* 1.3 – כתיבת תקשורת בסיסית בין גוגל (client) לסרבר.
* 5.3 – לכתוב את כל הפונקציות - event based של התוסף במצבי הגלישה של המשתמש.
* 25.3 – כתיבת ה-content script שמסנכרן את סרטון היוטיוב על פני כל המשתתפים.
* 1.4 – בניית GUI לכל חלונות התוסף.
* 1.5 – אופטימיזציה של כל הקוד הבנוי ותיקוני באגים ופרטים קטנים.

ניהול סיכונים

הסיכונים המרכזיים שלי במהלך הפרויקט זה להתמודד וללמוד את כל המידע החדש שנכלל בהכנת הפרויקט הזה. התקשורת עוברת באמצעות WebSocket שפועל בצורה ובעל אופי שונה מ - socket רגיל שהתרגלתי לעבוד איתו בפייטון. גם כן קיים הקושי להסתגל לעבודה סביב API חדש בעל תנאים ודרכי עבודה ייחודיים משלו (ה-API של תוספי כרום). בנוסף לכך קיים גם הקושי לכתיבת פרויקט ש-70 אחוז ממנו בשפה שחדשה לי (JavaScript).  
עקב סיכונים אלו החלטתי לפני ההתחלה של כתיבת הפרויקט עצמו לעשות מספר מיני פרויקטים שיעזרו לי להבין את כל החומר החדש ותכננתי את הלוז כך שקודם אעסוק ואנסה להתמודד עם החלקים הקריטיים בפרויקט שבהם אני חדש לחלוטין וחסר ניסיון ורק לאחר שאצליח להתמודד עם כל הנקודות בהם רציתי להתמקד ולהבין כיצד לכתוב אותם בסביבת העבודה החדשה שלי רק אחר כך אחבר בין כל הנקודות והקוד שכתבתי לבניית פרויקט מוגמר.

תיאור תחום הידע

להלן פירוט היכולות של הפרויקט הן בצד השרת והן בצד הלקוח.

פירוט יכולות בצד השרת

* הרשמה למערכת - משתמש חדש

רישום משתמש חדש לשרת המוריד את התוסף לראשונה בכרום.

* קליטת התחברות WebSocket חדש לשרת.
* נתינה ללקוח ID ייחודי.
* נתינה ללקוח username ייחודי שישמש אותו לאורך כל השימוש ורישומו במערכת.
* רישום פרטי ה-ID ,ה-username וה-WebSocket הנוכחי שבאמצעותו אותו לקוח תקשר עם הסרבר.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* התחברות מחדש של לקוח לשרת

במידה וה-WebSocket של הלקוח התנתק והלקוח רוצה לשלוח הודעה חדשה לשרת.

* נוצר בדיקה של ה-ID שאליו משתייך הלקוח ובמידה והוא קיים רושם את הלקוח מחדש במערכת עם ה-WebSocket החדש שדרכו התחבר.
* במידה ואין את אותו ה-ID המדובר השרת מעניק ללקוח ID ו-username חדש ורושם אותו במערכת.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* פתיחת חדר צפייה חדש

במידה והלקוח רוצה לפתוח סרבר הוא פונה לשרת על מנת לפתוח ולרשום את החדר.

* יצירת ID וסיסמא לחדר.
* רישום פרטי החדר במערכת: ID, סיסמא, משתתפים, URL של הסרטון, הזמן האחרון שה-host צופה הפיץ לשרת פקודה.
* שליחת ה-ID והסיסמא ללקוח.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* הצטרפות לקוח לחדר חדש

המידה ולקוח רוצה להצטרף לחדר צפייה הוא שולח את פרטי החדר לשרת.

* אימות פרטי החדר שהלקוח הכניס בפופ-אפ, במידה והפרטים נכונים, מוסיף אותו לחדר במערכת ובמידה ולא שולח ללקוח שהפרטים לא נכונים.
* מודיע לשאר חברי החדר שמשתמש חדש הצטרף אליו.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* יציאה של לקוח מהחדר

במידה והלקוח יוצא מהחדר הוא מודיע על כך לשרת.

* הורדה של הלקוח מהמערכת ברשימת הצופים בחדר הצפייה.
* העברת המסר על יציאת הלקוח לשאר המשתתפים בחדר.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* יציאה של לקוח שהוא ה-Host של חדר הצפייה

במידה וה-Host של חדר הצפייה יוצא הוא מודיע על כך לשרת.

* הורדה של הלקוח מהמערכת ברשימת הצופים בחדר הצפייה.
* העברת המסר על יציאת הלקוח לשאר המשתתפים בחדר.
* העברת הרשאות ה-Host ללקוח הצטרף לחדר אחריו.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* סגירת חדר צפייה

במידה ויוצא הלקוח האחרון מחדר הצפייה.

* השרת מוודא שאכן אף אחד לא שנשאר בחדר הצפייה.
* מחיקת חדר הצפייה מהמערכת.
* אובייקטים נחוצים: תקשורת, מבנה נתונים
* העברת broadcast לחברי החדר

במידה והלקוח עושה פעולה על הסרטון (מפעיל, עוצר, מעביר זמן) הוא מודיע על כך לשרת.

* יצירת זמן UTC שבא כל הלקוחות יבצעו את הפעולה שה-Host יזם.
* העברת מסר באמצעות שליחת הפקודה לWebSocket הרשומים כלקוחות בחדר הצפייה.
* אובייקטים נחוצים: תקשורת

פירוט יכולות בצד הלקוח

* התחברות לשרת ולמערכת

קבלת ID מהשרת להצטרפות הלקוח החדש

* בדיקת ה-Event שבא הלקוח הוריד את התוסף לראשונה.
* יצירת WebSocket חדש וחיברו לשרת.
* קבלת ה-ID וה-username מהשרת ורישומם כמשתנים גלובליים.
* אובייקטים נחוצים: תקשורת
* התחברות מחדש לשרת

במידה וה-WebSocket של הלקוח מתנתק.

* יצירת WebSocket חדש ושליחת בקשה להתחברות מחדש באמצעות ה-ID הקיים.
* קבלת ההודעה מהשרת וקבלת ID ו-username חדש במידת הצורך או קבלת אישור להתחברות מחדש למערכת.
* אובייקטים נחוצים: תקשורת
* פתיחת חדר צפייה חדש

במידה והלקוח נמצא בסרטון יוטיוב ומעוניין לפתוח חדר צפייה

* הופעת פופ-אפ המציע למשתמש לפתוח חדר צפייה.
* במידה והמשתמש מעוניין ולוחץ על כפתור הפתיחה נשלח הודעה לשרת עם ה-URL של הסרטון ובקשה לפתיחת חדר צפייה.
* נפתח למשתמש פופ-אפ חדש שמכיל את ה-ID של החדר, סיסמתו ורשימת המשתתפים הקיימים בחדר.
* נפתח למשתמש הלקוח הראשות Host בחדר הצפייה.
* אובייקטים נחוצים: תקשורת, ממשק משתמש
* שימוש בחדר צפייה כ-Host

במידה והלקוח הינו Host בחדר הצפייה הנוכחי

* שליטה מלאה על זמן הצפייה.
* שליטה מלאה על כפתור ההפעלה.
* במידה והמשתמש סוגר את החלון או עובר על לאתר אחר הוא מוצא אוטומטית מן החדר.
* אובייקטים נחוצים: web scraping
* שליחת פקודות של ה-Host לשרת

כאשר המשתמש עושה פקודה על הסרטון באתר היוטיוב שלו.

* באמצעות content script שהוזרק לתוך האתר נאסף מידע על הפקודות שמשתמש מבצע.
* ה-script מודיע ללקוח על הפקודה ולאחר מכן היא מועברת לשרת על מנת לבצע broadcast לכך לשאר צופי החדר.
* אובייקטים נחוצים: web scraping, תקשורת
* שימוש בחדר צפייה כלא Host

במידה והלקוח אינו Host בחדר הצפייה הנוכחי.

* אין שליטה כלל על אף פרט בסרטון חוץ מהווליום – כל כפתורי התפעול נעלמים מן המסך.
* במידה והמשתמש סוגר את החלון או עובר על לאתר אחר הוא מוצא אוטומטית מן החדר.
* אובייקטים נחוצים: web scraping
* הצטרפות לקוח לחדר חדש

במידה ומשתמש חדש הצטרף לאותו חדר הצפייה של הלקוח

* קבלת הודעה מהשרת דרך ה-WebSocket של הלקוח על הצטרפות משתמש חדש לחדר.
* עדכון הפופ-אפ והוספת המשתמש החדש לרשימת הצופים בחדר.
* אובייקטים נחוצים: ממשק משתמש, תקשורת
* יציאה של לקוח מהחדר

במידה ומשתמש יוצא מחדר הצפייה של הלקוח

* קבלת הודעה מהשרת דרך ה-WebSocket של הלקוח על יציאה של משתמש מהחדר.
* עדכון הפופ-אפ והסרת המשתמש מרשימת הצופים בחדר.
* אובייקטים נחוצים: תקשורת
* חסימת פרסומות

במידה ובמהלך הסרטון מופיע פרסומת שעוצרת את הסרטון.

* קליטת שינוי באובייקטים באתר.
* מציאת אובייקט ה-DOM של הפרסומת.
* דילוג ישיר על הפרסומת בחזרה לסרטון.
* אובייקטים נחוצים: web scraping

שפת התכנות וסביבת העבודה

במהלך כתיבת הפרויקט שלי עסקתי בכמה שפות תוכנה שונות למספר שימושים שונים. להלן כל שפה רשימת השפות שהשתמשתי בהן, השימוש שנעשה בהן ומדוע בחרתי בשפה זו.

(1) **פייטון – Python**, בשפה זו השתמשתי אך ורק ליצירת השרת בפרויקט. השרת מנהל את כל הלקוחות במערכת ,את מצבי החדרי הצפייה ומתקשר עם ה-backend של תוסף הכרום. בחרתי בשפה זאתי לשרת מכיוון שיש לי ניסיון רב בשפה ככלל ובפרט בבניית שרתים. עקב סיבה זאת מצאת לנכון לכתוב את השרת בשפה שבה אוכל לכתוב בצורה הכי טובה וארגיש בנוח איתה לאורך כל הפרויקט.

(2) **HTML**, בשפה זו השתמשתי על מנת לכתוב את העמוד פתיחה שנפתח ברגע שהמשתמש מוריד לראשונה את התוסף ואת הפופ-אפים המופיעים למשתמש במהלך השימוש בתוסף. בחרתי בשפה זו כי היא השפה ה-default לכתיבת אתרים וקלה מאוד ללמידה ויישום בתוך הפרויקט שלי.

(3) **ג'אווה סקריפט – JavaScript**, בשפה זו השתמשי ברוב כתיבת הקוד של התוסף בין אם זה ב-backend שלה או ב-frontend. בנוסף לכך השתמשתי בה בשביל לכתוב סקריפטים של אתרי ה-HTML המופיעים בתוסף. בחרתי בשפה זו מכיוון שהיא השפה ה-default לכתיבת תוספי כרום והיא נועדה לעבודה עם html ואתרי אינטרנט ומכיוון שכל הפרויקט שלי עוסק בעבודה סביב אתר היוטיוב כתיבת הקוד ב-JS הינה הפתרון ההגיוני והטוב ביותר.

(4) **CSS**, בשפה זו השתמשתי על מנת לעצב את עמודי ה-HTML שנמצאים בפרויקט שלי. בחרתי בשפה זו מכיוון שהיא עובדת בצורה טובה עם HTML , היא שפה קלה ללמידה ולתפעול ויכולה להעניק לעיצוב הבנאלי של כתיבת HTML עיצוב ייחודי שנותנת הרגשה של עמוד יותר מוגמר ונעים לעין על מנת לשפר את חווית המשתמש.

(5) **JSON**, בשפה זו השתמשתי על מנת לכתוב את ה-manifest של תוסף הכרום. ה-manifest מציג ל-API של תוספי כרום את הסקריפטים הקיימים בתוסף, ההרשאות המיוחדות שהתוסף דורש, תמונות, כותרות, פופ-אפים וכדומה. בחרתי בשפה מכיוון שעל פי הנחיות ה-API של תוספי כרום זה השפה הדרושה לכתיבת ה-manifest של תוספי כרום.

חלוקת השפות בפרויקט שלי הינה:   
**70%** - *JavaScript* , **17%** - *Python* , **8%** - *HTML* , **5%** - *CSS*

ניסוח וניתוח הבעיה האלגוריתמית

במהלך הפרויקט נתקלתי במספר בעיות אלגוריתמיות שלאחר מכן ביססו לי את הפרויקט ובנו אותו לצורה שבה שהוא מוצג בתוצר הסופי.

**(1) כיצד ניתן לצרוך סרטון משותף על האינטרנט?**   
 - כלומר מה הדרך המועדפת לשיתוף סרטון בתוסף על מנת לקבל חוויית צפייה משותפת נוחה, מסונכרנת ומהירה.

**(2) איך למנוע את הפרסומות בסרטון היוטיוב?**   
 - במהלך הסרטון יכולים להופיע פרסומות העוצרות את סרטון היוטיוב ומנעות את הסנכרון בין צופי החדר.

**3)) איך לשמור את המידע על המשתמש בתוסף?**

- ישנה בעיה ב-API של התוסף בכך שה-backend של מאתחל את עצמו מחדש כל פעם שהמשתמש לא נוגע בכרום למספר שניות על מנת לשמור על ביצועי המחשב, כך כל פעם מחדש פרטי המשתמש מאתחלים ולא נשמרים במערכת.

**(4) כיצד לשמור על קשר בין התוסף לשרת?**

-מכיוון שהתוסף בנוי על JavaScript ופועל בעצם על כרום אז התקשורת הבנאלית שמתאפשר דרך שימוש בתוסף הינו רק דרך WebSocket. הבעיה בשימוש תקשורת זאתי היא שהיא אינה תקשורת רציפה ולאחר כמה שניות שה-socket אינו בשימוש הוא סוגר את עצמו ואז כאשר אני רוצה לשלוח הודעה לשרת מהלקוח אך ה-socket סגור ההודעה לא תשלח והתקשורת הרציפה תיעצר. דבר נוסף לבעיה היא שכאשר הלקוח בחדר הצפייה והשרת רוצה להעביר פקודה ללקוח וה-socket שלו נסגר אז ההודעה לא תועבר וייווצר תקל בפעילות חדר הצפייה.

תיאור אלגוריתמים קיימים

**(1) כיצד ניתן לצרוך סרטון משותף על האינטרנט?**

- הפתרון הראשון שעלה לראשי זה להוריד את סרטון היוטיוב ל-mp4 וליצור אתר נפרד יוטיוב באינטרנט שרץ על שרת בעל חדרים והלקוחות שרוצים לצפות בסרטון יפתחו את חדר הצפייה ושם "יוקרן" להם סרטון היוטיוב שהשרת הוריד לזיכרון שלו.

- הפתרון השני שחשבתי עליו זה לעבוד עם embedded video player משמע גם כן ליצור אתר שהשרת מתפעל אך במקום להוריד את הסרטון לזיכרון השרת הוא משתמש ב-URL מיוחד של סרטון יוטיוב המאפשר באמצעות HTML להציג את הסרטון באתר נפרד בשליטת השרת וכך ניתן להריץ את הסרטון בצורה משותפת בצורה דומה ואף יותר פשוטה מהפתרון הראשון.

- הפתרון השלישי זה להריץ את הסרטון בצורה משותפת על אתר יוטיוב עצמו. משמע לשלוט את כל לקוח דרך התוסף בהתקדמות הסרטון וכך להריץ בצורה משותפת לכל הלקוחות שבחדר הצפייה. כך שכל לקוח באתר יוטיוב בסרטון מעצמו אך הצפייה נעשית בצורה משותפת לחלוטין.

**(2) איך למנוע את הפרסומות בסרטון היוטיוב?**

- הפתרון הראשון שחשבתי עליו הינו פשוט להוריד את הסרטון ל-mp4 וכך למנוע לחלוטין הופעת פרסומות בסרטון היוטיוב – פתרון זה הולך יד ביד עם הפתרון הראשון בבעיה הראשונה.

- הפתרון השני הינו לדרוש מהלקוח להוריד תוסף החוסם פרסומות ביוטיוב, קיימים עשרות אופציות לחוסמי פרסומות באינטרנט ככלל ביוטיוב בפרט ולכן פתרון זה ניתן ליישום בקלות ופותר את הבעיה בצורה מלאה.

- הפתרון השלישי זה ליצור בעצמי חוסם פרסומות ליוטיוב, כך שכל פרסומת המונעת התקדמות בסרטון ייחסמו אוטומטית.

**(3) איך לשמור את המידע על המשתמש בתוסף?**

- הפתרון הראשון זה לשמור את כל המידע על האובייקטים בקובץ נפרד וכל פעם שה-backend חוזר לתפעול הוא קורא ממנו את המידע על המשתמש וכך מעדכן את המערכת. פתרון זה קל ליישום ופשוט.

- הפתרון השני זה לעבוד בסביבת עבודה של chrome.storgae.local ודרכו לשמור את כל המידע על המשתמש. סביבת עבודה זו היא דרך ה-API של תוספי כרום כך שהיא דורשת מאמץ למידה ויישום בתוך הפרויקט.

**(4) כיצד לשמור על קשר בין התוסף לשרת?**

- הפתרון הראשון שחשבתי עליו הינו לשלוח הודעות מהלקוח לשרת כל כמה שניות על מנת להשאיר את הקשר פעיל וכך ה-WebSocket של הלקוח לא ייסגר אף פעם לאורך פעילותו עם השרת.

- הפתרון השני הינו לעשות בדיקה לפני כל שליחת הודעה לשרת על מצב ה-WebSocket של הלקוח ובמידה והוא נסגר אז לחדש את הקשר עם השרת לטווח פעילות נוסף עד שדבר זה קורה שוב.

הפתרונות הנבחרים

(1) כיצד ניתן לצרוך סרטון משותף על האינטרנט?

החלטתי לבחור בפתרון השלישי שחשבתי עליו שהוא עבודה דרך אתר יוטיוב עצמו. כלומר בחרתי שהדרך המועדפת עלי בפרויקט לצרוך סרטון בצורה משותפת הינה דרך מניפולציה של הסרטון באתר יוטיוב בצורה מסונכרנת בין צופי החדר. בחרתי דווקא בפתרון זה מכמה סיבות:

- הסיבה ראשונה לכך היא שרציתי לתת לצופה חווית צפייה משותפת פשוטה וקלה לתפעול במלי שהמשתמש ירגיש שהוא צריך להתאמץ יותר מידי בשביל לחוות סרטון עם חבריו. יצירת אתר חיצוני שדרכו הם יצפו בסרטון הרגיש לי שנדרש למשתמש יותר מאמץ לתפעול ולפתיחת חדר צפייה ובנוסף לכך הייתי מודע למספר אתרים שמעניקים את שירות זה. רציתי לצאת מחוץ לקופסא ולחשוב על פתרון ייחודי ויצירתי לבעיה.

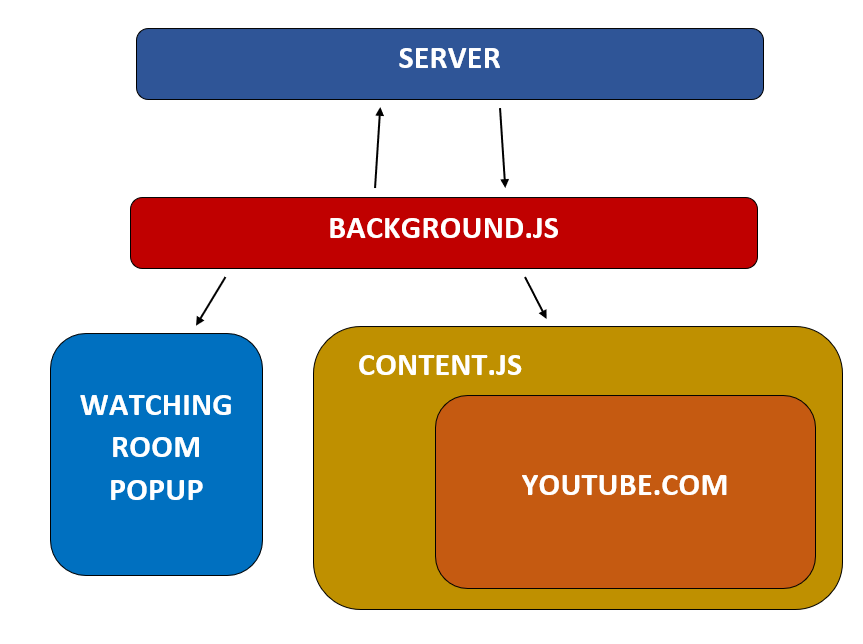
- הסיבה השנייה לכך היא שרציתי לנצל את שימושי בתוסף כרום בצורה המלאה ביותר. מכיוון שהחלטתי כבר שאני עוסק בסביבת העבודה של תוספי כרום רציתי לנצל את שימושי בתוסף בצורה המלאה ביותר. לאחר מחקר לא קצר גיליתי שבעזרת התוסף אני יכול לשלוט על אובייקטים באתרים חיצוניים ולבצע עליהם מניפולציות כך שאוכל לתת למשתמש חוויית צפייה משותפת נקייה דרך אתר יוטיוב עצמו. מכיוון שפתרון זה עוסק בסביבת העבודה של תוסף כרום רציתי לנצל כבר את ההזדמנות לכך שאני עובד על תוסף ולנצל את יכולותיו על כרום בצורה המקסימלית.

**מבנה המערכת:**

אפשרתי את חווית הצפייה המשותפת דרך יוטיוב בכך שבניתי מספר פונקציות הבנויות על events ב-backend של תוסף הכרום כך שברגע שאותר שהמשתמש בחדר צפייה הוא "מזריק" את ה-content.js לתוך אתר היוטיוב ובעזרת התקשורת בין ה-background.js וה-content.js ניתן ליצור חווית צפייה מסונכרנת בין כל המשתתפים. ה-background.js מקבל הודעות לגבי פעולות שצריך לבצע על הסרטון מהשרת ולאחר סינון הודעות שחוסך עומס על אתר היוטיוב הוא מעביר את זה הלאה ל-content.js שמבצע את הפעולות הללו על אובייקט הסרטון ש"לקח" מהאתר. מכיוון שכאשר שמזריקים קוד לתוך האתר בצורה זו יש לך גישה לכל התוכן באתר, עקב כך יכולתי להשיג את האלמנטים הרצויים שכוללים את הסרטון עצמו וכפתורי ההפעלה של האתר ובעזרתם יכולתי להריץ בקוד events שעוקבים אחרי פעולות שהמשתמש עושה על הסרטון, בין אם זה לעצור, להפעיל או להעביר את הזמן. לאחר שנקלט אחד מן הפעולות הללו במידה והלקוח הינו ה-host של חדר הצפייה, רק למשתמש שיצר את החדר יש שליטה מלאה על תפעול הסרטון. במידה והמשתמש אינו host מתבטל לו האפשרות לשלוט על הסרטון בכך שנעלם כפתורי ההפעלה והאופציה ללחוץ על הסרטון.

מתאפשר חווית צפייה משותפת בכך שכל המחשבים מבצעים את הפעולות בו זמנית. אני מוודה שאכן כל המחשבים מריצים את הפעולה על הסרטון בו זמנית בכך שהשרת שולח לכל המחשבים את הפקודה שעליהם לעשות על הסרטון ובנוסף לכך נותן להם את ה-UTC (Coordinated Universal Time). לאחר מכן שההודעה מועברת ל-content.js על ידי ה-background.js הוא מחשב תוך כמה זמן הוא צריך להריץ את הפעולה וכך גם מבצע אותה על השנייה המתאימה. כך מתאפשר חוויית צפייה מסונכרנת בין צופי החדר.

תרשים תקשורת כללית בין המרכיבים במערכת:



**SERVER**

**BACKGROUND.JS**

**WATCHING ROOM POPUP**

**CONTENT.JS**

**YOUTUBE.COM**

תרשים זרימה לפעולה של ה-host על סרטון היוטיוב והשפעתו על שאר חברי הקבוצה:

(2) איך למנוע את הפרסומות בסרטון היוטיוב?

החלטתי לבחור בפתרון השלישי לבעיה זו. כלומר מצאתי לנכון להתמודד עם בעיה זו בכך שאני אכתוב בתוך התוסף שלי חוסם פרסומות למניעת הפרסומות המפריעות להרצת הסרטון במהלכו. בחרתי בפתרון זה מכיוון שראשית בחרתי בבעיה הראשונה בפתרון של הרצת הסרטון המשותף על אתר יוטיוב ולכן הורדת הסרטון ל-mp4 היא אינה אפשרות זמינה כעט. בנוסף לכך החלטתי לא לדרוש מהמשתמש להוריד חוסם פרסומות חיצוני מכיוון שבתוך יוצר מוצר ללקוח פחות מתאים או מקובל לדרוש ממנו להוריד מוצר חיצוני על מנת לשפר את חווית המוצר שאותו אני יוצר. ולבסוף עוד סיבה לכתיבת חוסם הפרסומות בעצמי היא מכיוון שאני כותב קוד על אתר יוטיוב יש לי כבר בסיס טוב לכתוב עליו חוסם פרסומות מבלי להתאמץ יותר מידי מכיוון שכל התשתית כבר בנויה מסביב לפונקציה עצמה שתחסום פרסומות. עקב הסיבות הללו החלטתי לבחור בפתרון השלישי ואכן לכתוב חוסם פרסומות לאתר יוטיוב.

**מבנה המערכת:**

חוסם הפרסומות שבניתי בתוסף רץ על ה-content.js שאחראי על הרצת פקודות בסרטון המשותף ביוטיוב. הוספתי בקובץ זה MutationObserver שהוא בעצם event שמריץ פונקציה כל פעם שמתקיים שינוי ב-DOM של האתר. לאחר מכן בפונקציה אני מנסה למצוא אלמנטים של פרסומת היכולים להתקיים. אם נמצא האלמנטים המוודים שאכן ישנה פרסומת המפריעה להרצת הסרטון יש שני אפשרויות לפי סוג הפרסומת המופיעה. (1) פרסומת בעלת כפתור דילוג: במידה וזה המצב אז כפתור הדילוג נלחץ אוטומטית עוד לפני שהוא מופיע בכלל על מסך המשתמש. (2) אם זאת פרסומת ללא כפתור דילוג אז נשיג את אלמנט של סרטון הפרסומת עצמו ונגדיר ב-timeline שלה שהיא בסוף הסרטון וכך זה ידלג לחלוטין על הפרסומת. פעולות אלו קוראות במילי-שניות ככה שהמשתמש בקושי שם לב אליהם וכך חווית הצפייה המשותפת אינה נפגעת כלל.

תרשים זרימה לחוסם הפרסומות בסרטון היוטיוב:

(3) איך לשמור את המידע על המשתמש בתוסף?

הפתרון שבחרתי לבעיה זו הינו הפתרון השני שהוא עבודה עם storage.local של כרום כמקום השמירה למשתנים הגלובליים של המשתמש. בחרתי דווקא בפתרון זה לעומת עבודה עם קובץ נפרד המכיל את המידע על המשתמש מכמה סיבות. ראשית ניתן לשמור באכסון של כרום אובייקטים ולא רק מילים, כך ניתן לשמור בצורה יותר נוחה את המידע כמו ה-WebSocket של הלקוח. בנוסף לכך שמירת המידע באכסון כרום יותר מאובטח מקובץ חיצוני ושמור בצורה פנימית בתוך הדפדפן לאורך כל השימוש. לבסוף העדפתי להשתמש באכסון של כרום מכיוון שהיא פועלת בצורה הרבה יותר נוחה סביב הממשק הבנוי של תוספי כרום ולכן העבודה איתה היא יותר אינטואיטיבית ותואמת לפרויקט עצמו.

**מבנה המערכת:**

כאשר לראשונה המשתמש מוריד את התוסף נבנה אובייקט המכיל את כל המשתנים של המשתמש כגון: ID, username, WebSocket, פרטי חדר צפייה וכו'. לאחר מכן אני מציב את האובייקט הזה בתוך ה-local storage של כרום ולבסוף בכל פעם שאני משנה את האובייקט של המשתמש אני דואג לעדכן אותו באכסון הלוקלי כך שאם ה-background.js עושה unload ברגע שהוא יחזור לתפעול הוא יוצא מחדש את האובייקט המשתמש המעודכן וכך נשמר כל המידע החשוב על המשתמש לאורך השימוש בתוסף. בנוסף לכך הצבתי עוד פקודה שאמורה לוודא כי כל פרטי המשתמש נשמרו והיא קליטת ה-event שברגע ש-background.js עומד לעשות unload הוא מעלה את פרטי המשתמש לאכסון הלוקלי וכך מאבטח אכסון של כל פרטי המשתמש במצב בו הקוד עומד לעבור אתחול.

תרשים זרימה לשמירת וביצוע פרטי המשתמש ב-local storage של כרום.

(4) כיצד לשמור על קשר בין התוסף לשרת?

הפתרון שבחרתי לבעיה זו הינה ערבוב בין שני הפתרונות הקיימים בהתאם לסיטואציה של התוסף. הגעתי למסקנה שישנם חסרונות ויתרונות לשני הפתרונות ועל מנת לפתור את הבעיה ולחשוב על פתרון מתאים צריך להשתמש בשני הפתרונות בכך שכל פתרון בסיטואציה שונה. כאשר מתקיים קשר רגיל של שאלה תשובה בין הלקוח לשרת אז לפני כל פעם שאני שולח הודעה דרך ה-WebSocket אני בודק ראשית את מצב החיבור. אם מצבו טוב אז הוא ממשיך ושולח את ההודעה אך אם הוא לא אז נשלח הודעת חיבור מחדש לשרת עם אובייקט WebSocket חדש, לאחר שדבר זה מתבצע השרת שומר מחדש את פרטי הלקוח ל-WebSocket העדכני והחדש ולאחר מכן מאשר חיבור מחדש ללקוח. לעומת זאת במצב שהלקוח החדר צפייה אז הוא צריך להיות מוכן בכל שנייה לקבל הודעה מהשרת ולכן אי אפשר להסתמך על יצירת קשר חדש כל פעם שהקודם נופל ולכן אני שולח אשלח הודעה כל כמה שניות על מנת לשמור על החיבור בין הלקוח לשרת.

**מבנה המערכת:**

במצב כאשר הלקוח אינו בחדר צפייה אז לפני כל פעם שהלקוח עומד לשלוח הודעה לשרת נבדק מצב ה-WebSocket של הלקוח, במידה והתקשורת פתוחה הוא שולח הודעה לשרת ומתקשר כרגיל. במידה והקשר בניהם במצב סגירה אז הלקוח פותח מחדש את התקשורת עם WebSocket חדש באמצעות ה-ID של הלקוח – השרת מוודא שה-ID אכן נמצא במערכת ואם הוא אכן נמצא אז מעדכן באכסון השרת ה-WebSocket של הלקוח ואם לא נמצא ה-ID אז נשלח ללקוח ID חדש ורושם אותו במערכת בצורה מחודשת. במידה והלקוח בחדר צפייה אז כל כמה שניות נשלח לשרת הודעה דרך ה-WebSocket של הלקוח על מנת לשמור על קשר פתוח בין השניים במידה והשרת רוצה להפיץ פקודה ללקוח בנוגע לסרטון.

תרשים זרימה לתקשורת בין התוסף לשרת לפי מצב.

פיתוח הפתרונות בשכלול הקוד עם שפת התכנות

(1) כיצד ניתן לצרוך סרטון משותף על האינטרנט?

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי"הזרקת" הקוד לתוך סרטון היוטיוב, קבלת מידע מהמשרת ותפעול החדר מ-background.js

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטימעקב ה-content.js על הסרטון ועדכון סטטוס

קליטת פקודות של המשתמש על הסרטון במידה והמשתמש הוא ה-host

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

ביטול שליטה על הסרטון במידה והמשתמש אינו host

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

(2) איך למנוע את הפרסומות בסרטון היוטיוב?

האלגוריתם שחוסם פרסומות ב-content.js

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

(3) איך לשמור את המידע על המשתמש בתוסף?

אתחול האובייקט user באכסון הלוקלי של כרום

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיפונקציה לעדכון האובייקט בשמירה הלוקלית ופונקציה לקבלת האובייקט מהאכסון

(4) כיצד לשמור על קשר בין התוסף לשרת?

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיבדיקת החיבור לפני שליחת הודעה לשרת כאשר לא מדובר על תקשורת של חדר צפייה

במצב שבוא המשתמש בחדר צפייה ונשלח הודעה לשרת כל כמה שניות על מנת לשמור על החיבור קיים זה פונקציה שרצה כל כמה שניות ומוגדרת בבעיה מספר 1 בפונקציה (run room process)

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תיאור המודלים של מערכת התכנה