מסמך הניתן לעריכה מקבילית -Multi Editor

שם המגיש: אייל לויטון

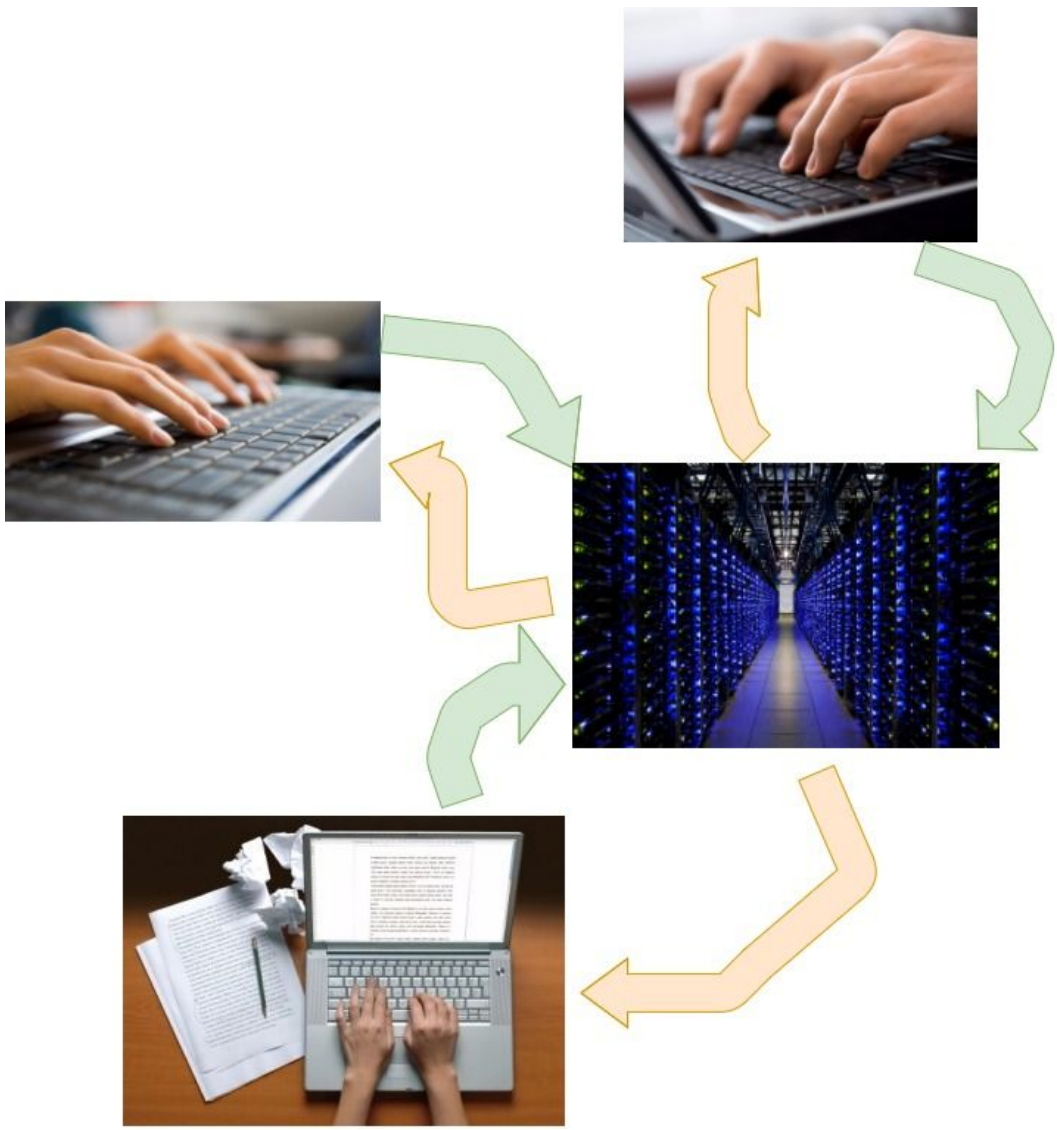
מספר תעודת זהות: 206132417

בית ספר: ליאו באק חיפה

מקצוע: הגנת סייבר

שמות המנחים: אלון בר-לב ושרית לולב

שם הפרויקט: Multi Editor



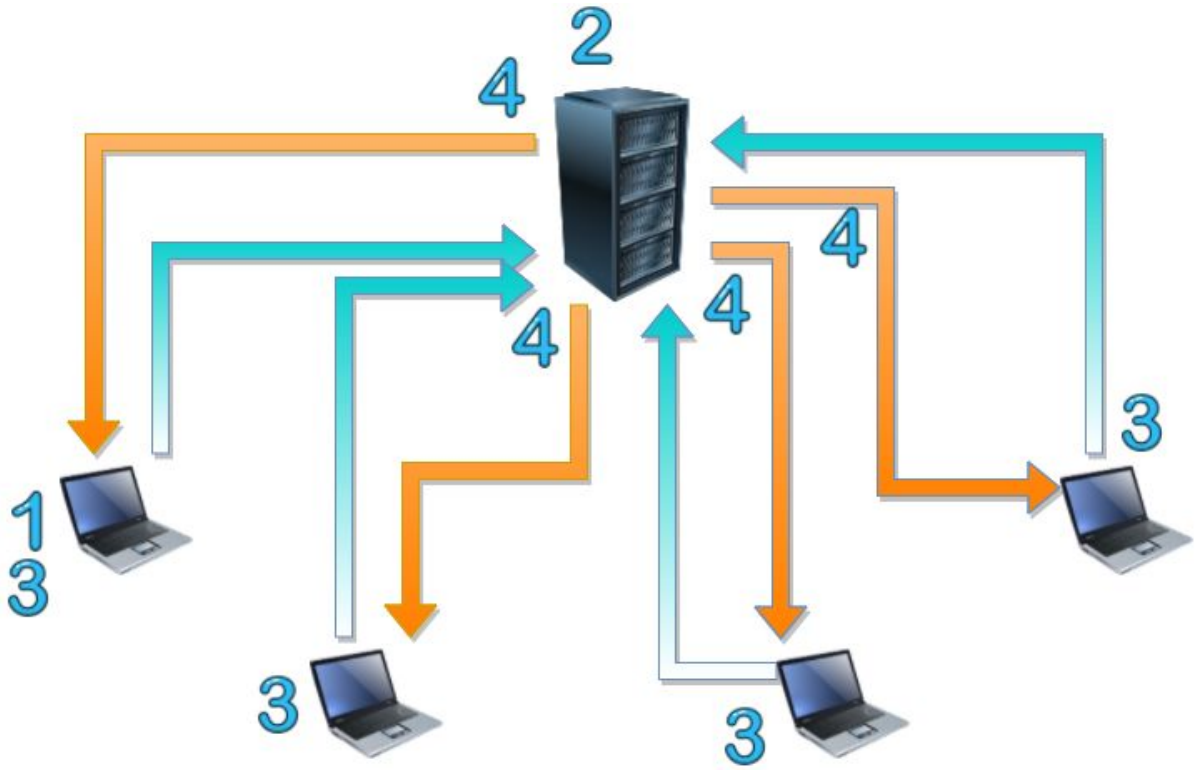
תוכן עניינים

מבוא

הפרויקט שאותו בחרתי לבצע הוא ה-Multi Editor, אפליקציית אינטרנט המדמה את השימוש בגוגל דוקס (google docs), כלומר, אפליקציה שעליה כמות גדולה של אנשים יכולה לכתוב עליה ולערוך אותה במקביל, כאשר הם רואים את השינויים של כל אחד בזמן אמת.

האפליקציה מאפשרת עבודה נוחה על מסמך אחד עם אנשים רבים כאשר לא תמיד ניתן להיפגש ולעבוד יחדיו, וכך ניתן לראות את המסמך באותו הזמן, ולערוך אותו באותו הזמן ומאפשר עבודה נוחה ומהירה על קובץ אחד. לאפליקציה אפשרויות עיצוב בנוסף לכתיבת מסמך פשוט: הדגשה, קיו תחתון, וכתיבה בצבעים (אדום, כחול, ירוק).

ארכיטקטורה



תיאור	שלב
המשתמש לוחץ על מקש במחשבו הפרטי והוא נשלח אל הסרבר.	1
הסרבר מעדכן את המסמך בתוכו ומעדכן את הגירסא שלו.	2
כל המחשבים שולחים בקשה אל הסרבר לקבלת המידע עם הגירסא שלהם.	3
הסרבר שולח את כל המסמך אל כל מחשב שביקש במידה והגירסא אינה תואמת.	4

רקע תיאורטי

במהלך הפרויקט למדתי כיצד מספר אנשים מסוגלים לעבוד על אותו המסמך במקביל, כאשר כולם רואים את אותו המסמך לאחר שינוי שיכיל כל אחד מן המשתמשים. בפרויקט שלי עשיתי זאת בכך שלאחר כל לחיצה של משתמש, יעבור האות שנלחצה אל השרת ומיקומה, השרת יכניס את האות במיקום המבוקש ויעדכן את תוכן המסמך, ויעדכן את הגירסא שעליה הוא עובד.
כל לקוח, בכל שנייה יבקש את המסמך שבשרת, במידה והגירסא בין הלקוח והשרת שונה, השרת ישלח את תוכן המסמך.
מכיוון שהשרת הוא סינכרוני, ההודעות מגיעות אחת אחרי השנייה ולכן הסדר של כתיבת ההודעה נשאר זהה, אך דבר זה מוריד בביצועי המהירות של התוכנה.

<u>דרכים נוספות שבהן היה ניתן ליישם את הפרויקט:</u>

- במקום עדכון כל המסמך מחדש בכל פעם שמתבצע שינוי היה ניתן לבצע שינוי רק במקומות שהשתנו. כלומר, במידה ונלחץ מקש במקום A, כל שאר הלקוחות יכניסו במקום A את האות שהוקשה, במקום לעדכן את כל המסמך. לא בחרתי בדרך זו מכיוון שהיא כוללת סיבוכיות לוגית גבוה מאוד וקשיים פרטיים רבים, לדוגמה, במידה ובמסמך נכללו מספר רב של הקשות, לקוח חדש שיכנס אל השרת יאלץ להוסיף כל מקש אחד אחרי השני, דבר שייקח זמן רב מאוד. על מנת לפתור זאת צריכה להיכנס לוגיקה אל מי נכנס מתי אל השרת, ובנוסף צריך לשמור את כל הגרסאות הקודמות, דבר שיכול לתפוס מקום רב בזיכרון.

- הכנת שרת אה-סינכרוני היה עלול להאיץ את עבודת השרת ומכאן גם כן את עבודת העדכון של הלקוח, אך לאחר התייעצות עם המנחה, לא היה בטוח בוודאות כי שרת אה-סינכרוני יזרז את תהליכי התוכנה, מכיוון שלמרות שהפעולה היא אה-סינכרונית, קיים רק תהליך אחד שעובד בתוכנה, והיא אינה תומכת בריבוי תהליכים בשרת, על מנת לשמור על אמינות הגרסאות ששמורות בשרת, ולכן על פי דבריו של המנחה, הוספת שרת אה-סינכרוני עלול לקחת זמן רב להכנה, לשינוי לא משמעותי, ולכן העדפנו לא להסתכן מבחינת זמנים וליצור סבר סינכרוני.

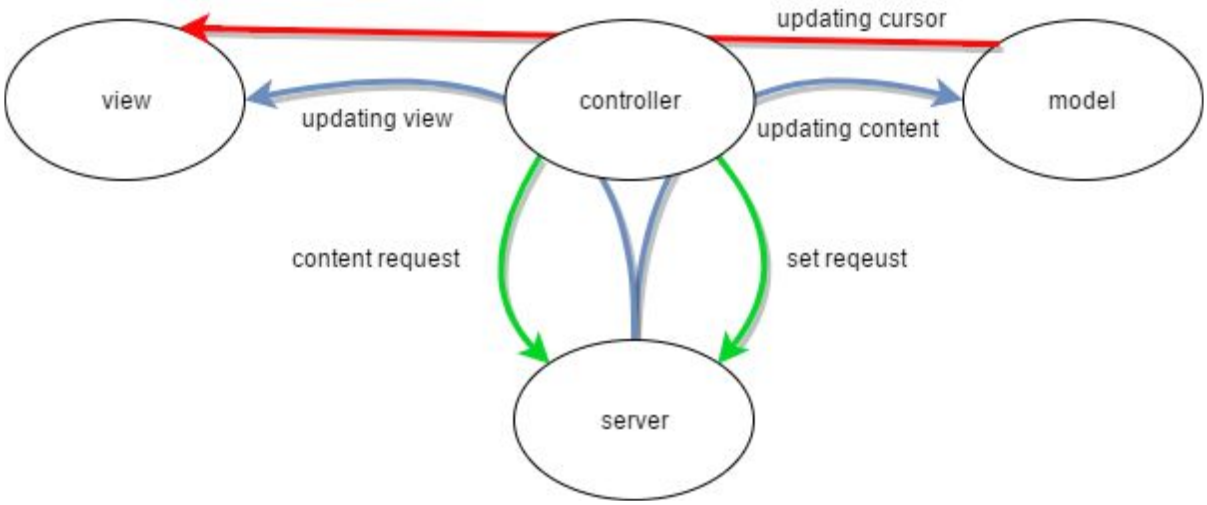
- בפרויקט שלי מבנה הנתונים בו בחרתי להשתמש הוא מערך חד מימדי. בתחילת הפרויקט השתמשתי ב-string ארוך מאוד, אבל במהרה החלפתי אל המערך. שימוש ב-string הוא אינו יעיל מכיוון כאשר משתמשים בסימונים כמו <br>, שה-html משתמש בו על מנת לרדת שורה, המשתמש רואה סימון זה כירידת שורה, וכאשר צריך להעביר את הסמן על גביו יש "לקפוץ" מעל כל המילה, והלוגיקה הכרוכה במציאת המעילה והמעבר מעליה הייתה מאוד מסורבלת. שימוש במערך אחד ארוך פותר בעיה זו בכך שהסמן קופץ בלוק שלם של המערך, וכך שינוי הסמן פשוט מאוד, ומיעל את החיפוש הספציפי לכל "בלוק מיוחד" שכזה.

טכנולוגיות בהם השתמשתי במהלך הפרויקט:

- Python היא שפת תכנות דינמית מהנפוצות ביותר. אחד המאפיינים הבולטים בתחביר השפה הוא השימוש בהזחה להגדרת בלוקים של קוד (ללא שימוש בסוגריים או במילים שמורות לצורך כך, כמו ברוב השפות הנפוצות).
- Javascript היא שפת תכנות דינמית מונחית־עצמים המותאמת לשילוב באתרי אינטרנט ורצה על ידי דפדפן האינטרנט בצד הלקוח. השפה מרחיבה את יכולות שפת התגיות הבסיסית HTML ומאפשרת בכך ליצור יישומי אינטרנט מתוחכמים יותר.
- HTML שפת תגיות לתצוגה ועיצוב דפי אינטרנט ותוכן לתצוגה בדפדפן.
- TCP הוא פרוטוקול בתקשורת נתונים המבטיח העברה אמינה של נתונים בין שתי תחנות ברשת מחשבים.
- HTTP או Hypertext Transfer Protocol הוא פרוטוקול תקשורת שנועד להעברת דפי HTML ואובייקטים שהם מכילים (כמו תמונות, קובצי קול, סרטוני פלאש וכו') ברשת האינטרנט. התקשורת ב־HTTP מתחילה ביצירת שיחה בין השרת ללקוח באמצעות פרוטוקול TCP ונמשכת בסדרה של בקשות (requests) ותשובות (responses) שנשלחות על ידי הלקוח והשרת, בהתאמה.
- MVC או Model-View-Controller היא תבנית עיצוב בהנדסת תוכנה המשמשת להפשטת יישום כלשהו. התבנית מתארת טכניקה לחלוקת היישום לשלושה חלקים, מודל, מבט ובקר, המחוברים ביניהם בצימוד רפוי מונחה אירועים. בדרך זו, התלות ההדדית בין ממשק המשתמש לשאר חלקי התוכנה פוחתת, ואת החלקים השונים ניתן לפתח באופן בלתי-תלוי. בנוסף, קל יותר לתחזק את התוכנה וכן לעשות שימוש חוזר בחלקי היישום שהופרדו.
- server הוא מחשב המריץ תוכנה אחת או יותר, ומספק שירותים למחשבים אחרים.

מימוש

דיאגרמת בלוקים:

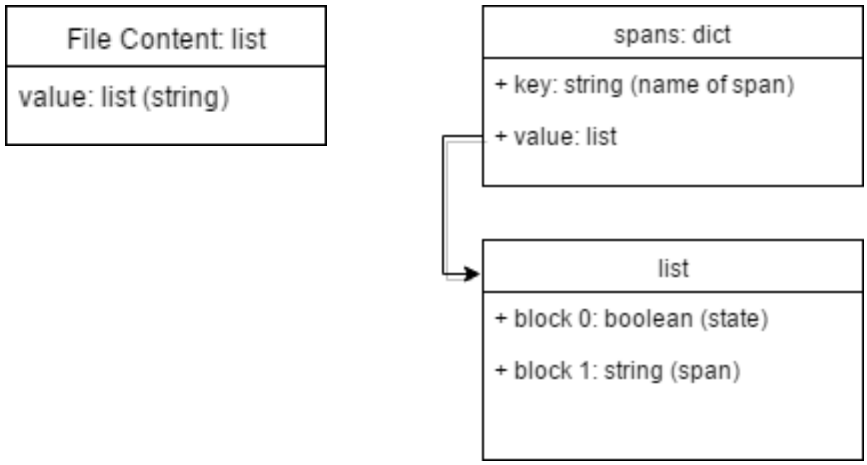


מימוש התוכנה מחולק לארבעה חלקים עיקריים:

- הבקר (controller) הוא האחראי על כל האינטראקציה בין שאר גורמי התוכנה. הבקר מקבל קלטים מן המשתמש, ושולח אותם בהתאם אל שאר מרכיבי המערכת. כאשר מקש מסוים נלחץ, נקראת פעולה מן הבקר הפועלת בהתאם לקלט, כלומר, כאשר נלחץ מקש "רגיל" (אות מסוימת אותה ניתן להציג על המסך), הבקר שולח את מקש זה ומיקומו אל השרת, ומחכה למסמך המעודכן. כאשר מגיע המסמך המעודכן, מעביר הבקר את כל המסמך למודל, והמודל מעדכן את עצמו ואת גירסתו.
  כאשר נלחץ מקש שאינו ניראה על גבי המסמך, כגון חצים או כפתורי HOME\END, כפתורים אלו משנים רק את מיקומו של הסמן, ומכיוון שאין תמיכה של סמנים מרובים, מקשים אלו יעדכנו אך ורק את מיקום הסמן במודל.
  בנוסף לכך, הבקר גם מעדכן את המודל כאשר נלחץ אחד מן כפתורי העיצוב, \bold \underline\ color, ומכניס אל מערך האותיות span פתיחה וסגירה של עיצוב בהתאמה לכפתור שלחץ.
- המודל (model) אחראי על מבנה הנתונים, שינויים בתוכו גירסאת המסמך שבבעלות המשתמש ומיקום הסמן. מבנה בנתונים בו בחרתי להשתמש בפרויקט שלי הוא מערך, כאשר כל תא במערך היא אות או בלוק מסוים, דבר המקל על הזזת הסמן ומחיקת בלוק שלם (כגון span, או
 המדמה ירידת שורה).
- המבט (view) אחראי על התצוגה של המסמך, והוא משנה את הhtml המוצג על המסך ומעדכן אותו מחדש לאחר כל הקשה על מקש, ובנוסף לכך כל שנייה מוקצב טיימר האחראי לבדיקה בשינוי המסמך השמור בשרת, במידה והגירסא שונה, מתעדכן המבט.
  המבט בנוסף לכך גם אחראי על שינוי הסמן. הסמן מיוצג כdiv ארוך, ומכיוון שזהו אובייקט, כל שינוי במסמך, מוחק המבט את הdiv ודוחף אותו מחדש אל המסמך במקום המתאים.

- <u>השרת (סרבר)</u> אחראי לשמירת הגירסא העדכנית ביותר. כל המשתמשים שמתחברים אליו שולחים אליו תיו ומיקום, והשרת מעדכן את עצמו, וכאשר משתמש יבקש להתעדכן הוא יקבל את הגירסא העדכנית ביותר. כמו המודל, גם בשרת מיוצגים הנתונים במערך אחד גדול. דבר המקל על תיאום בין מערך הלקוח ומערך השרת, מכיוון שהם בנויים באופן זהה.

מבני נתונים

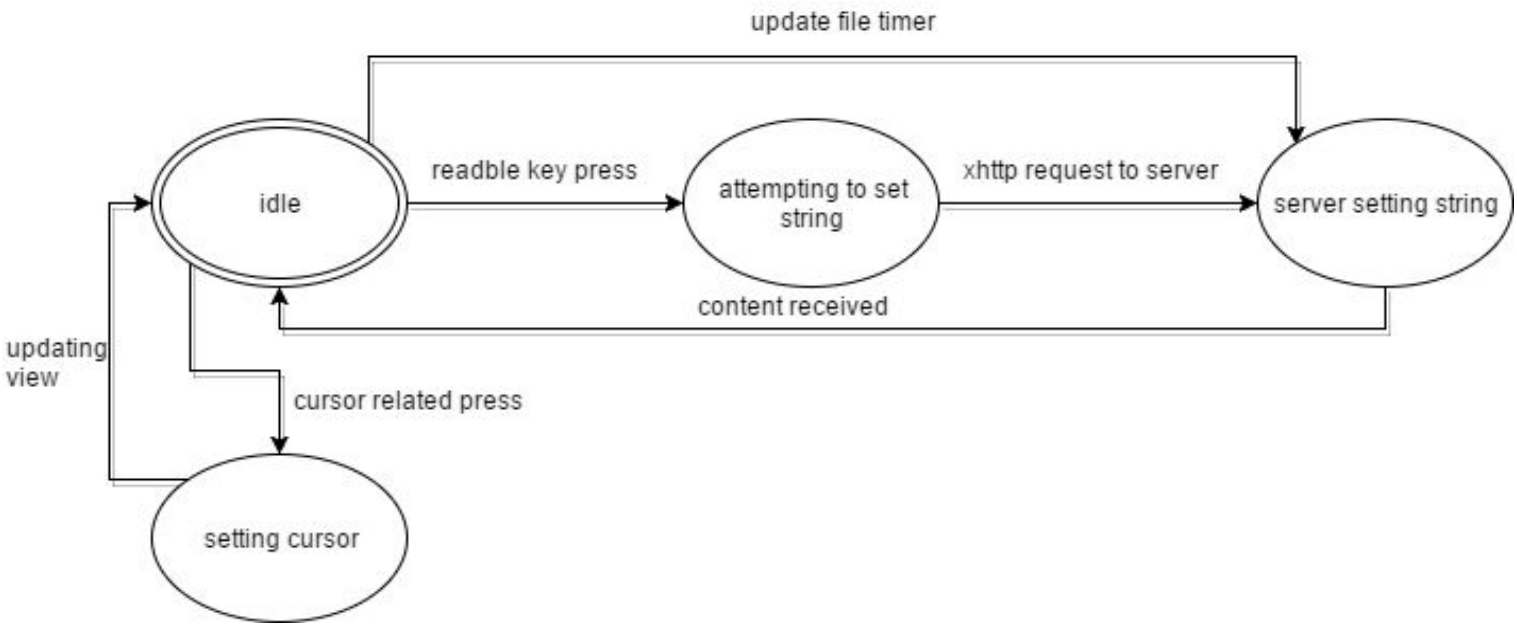


מבנה הנתונים הראשון בו אני משתמש הוא list של string המרכיבים את תוכן המסמך, כאשר כל בלוק מייצג "משבצת" אחת בדף, שאותה יעבור הסמן. לדוגמא, אם בלוק מסוים מכיל "hello", כאשר יגיע אליו הסמן, הוא ידלג על כל מילה ולא רק אות אות, דבר המקל שימוש בspan או <br>, כאשר המשתמש רואה צירופים אלה כבלוק אחד ולכן על התוכנה לדלק על כל הבלוק.

מבנה הנתונים השני הוא הspans. מבנה זה הוא מסוג dict, כאשר הkey הוא שם קבוע של הכפתור המוצג בhtml, והvalue הוא list באורך קבוע של 2, כאשר בבלוק הראשון מוכתב אם הכפתור צריך "להידלק" או לא, כלומר, במידה והסמן נמצא בתוך הspan המתאים.

State Machine

פעולות הclient:



Input events

Input event	Description
readble key press	any key press that's visible (char, space, enter).
cursor related press	any key press that controls the cursor (left, right, home, end).
update file timer	interval set every second to check if file has changed.

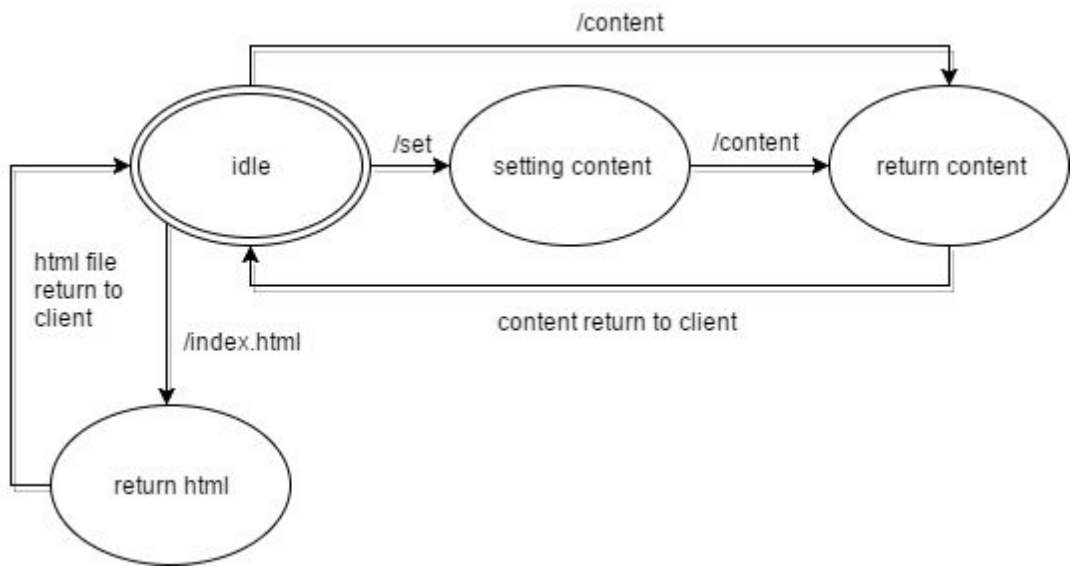
Output events

Output event	Description
xhttp request to server	client request to set or change something in form
content received	server sent answer with new content or empty string if content hasn't changed.
updating view	view gets updated by new cursor position.

States

Output event	Description
Idle	wating for key press or interval.
attempting to set string	client sends key to server.
server setting string	server updates content with giving string, or returning latest content.
setting cursor	setting cursor with new position.

פעולות הserver



Input events

Input event	Description
/set	set request to server. set giving string in the suitible place.
/content	content request to server. returning content if versions are different.
/index.html	return html file from server.

Output events

Output event	Description
content return to client	server return all content to the client.
html file return to client	server return html page to the client.

States

Output event	Description
Idle	server waiting for connection.
setting content	server set content in the correct position within file.
return content	return content to client.
return html	return html file to client.

בעיות ידועות

בעיה ידועה בפרויקט היא האטיות שבין הסרבר לקצב ההקלדה. בגלל בעיה זו הסמן (שאינו מושפע מן הסרבר) מתעדכן מהר יותר ממה שהדף עצמו מתעדכן.
בעיה זו נוצרת משתי סיבות:

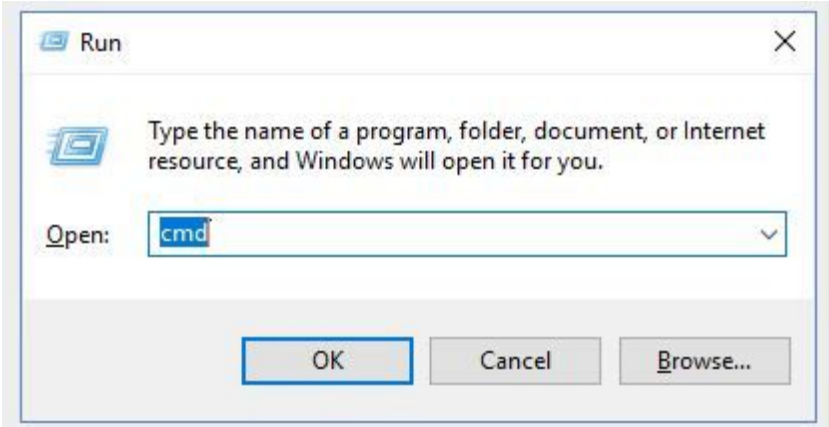
- עדכון כל הקובץ מחדש במקום עדכון חלק מסוים במסמך. שינוי זה מקל על זמן הריצה מכיוון שהתוכנה משנה חלק קטן מן הקובץ ולא תעדכן מחדש את כל הקובץ בכל פעם שקורה שינוי.
- סרבר סינכרוני ולא אה-סינכרוני. הסרבר הסינכרוני דואג לבקשה אחת בכל פעם, ובנוסף לכך הוא מעבד בקשה אחת בכל פעם, לא כמו הסרבר האה-סינכרוני, שאינו נעצר ויכול לעבוד על כמה דברים במקביל, וכך מגיעות הבקשות ביחד והן מטופלות מהר יותר ובאופן אפקטיבי יותר, כך גם התשובה מהירה יותר והתוכנה תגיב מהר יותר.

בעיה נוספת היא באגים עם כפתורי העיצוב. לפעמים לחיצה על הכפתור מוסיפה את הspan אך הסמן אינו נמצא בתוכו, ועל המשתמש להזיז את הסמן ידנית. מכיוון שאפיונים אלו של התוכנה הוספו בשלב יחסית מאוחר של העבודה הם לא תוקנו עד זמן ההגשה, במידה והן היו מוכנסים קודם, היה אפשר למצוא את מקור הבעיה ולתקן.

התקנה ותפעול

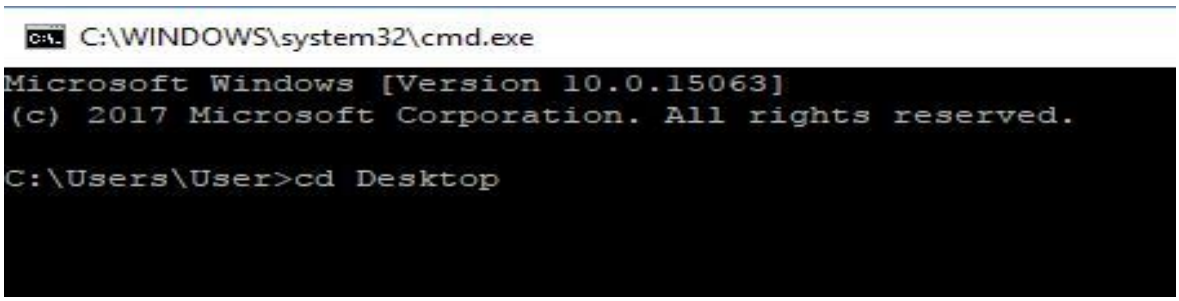
הוראות להרצת הסרבר

*הערה: הסרבר רץ רק על סביבת ווינדוס
פתיחת command prompt על ידי לחיצת win key + r ובחלון שנפתח לרשום cmd.



בcommand prompt יש להקליד את תיקיית האב של הפרויקט (המקום בוא נמצאת תיקיית הפרויקט). בדוגמה למטה, התיקייה נמצאת בdesktop, ולכן אעביר את סביבת העבודה שלי לשם על ידי הקלדת:

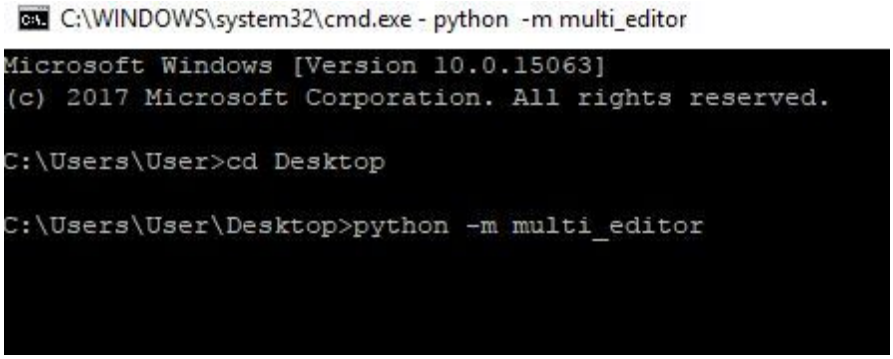
> cd [location of multi_editor parent]



לאחר מכאן, יש להקליד python -m multi_editor על מנת להריץ את הסרבר. ניתן לשנות ip ו-port על ידי הקלדת:

--address [ip address]
--port [port]

לדוגמה: python -m multi_editor --address 127.0.0.1 --port 8080



הוראות להתחברות אל הסרבר

*הערה: ניתן להתחבר אל הסרבר רק דרך דפדפן כרום.

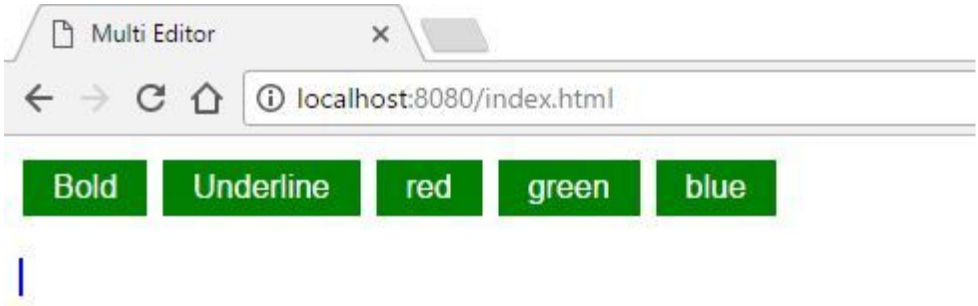
על מנת להתחבר יש לכתוב בשורת הip של הדפדפן:

http://[ip]:[port]/index.html

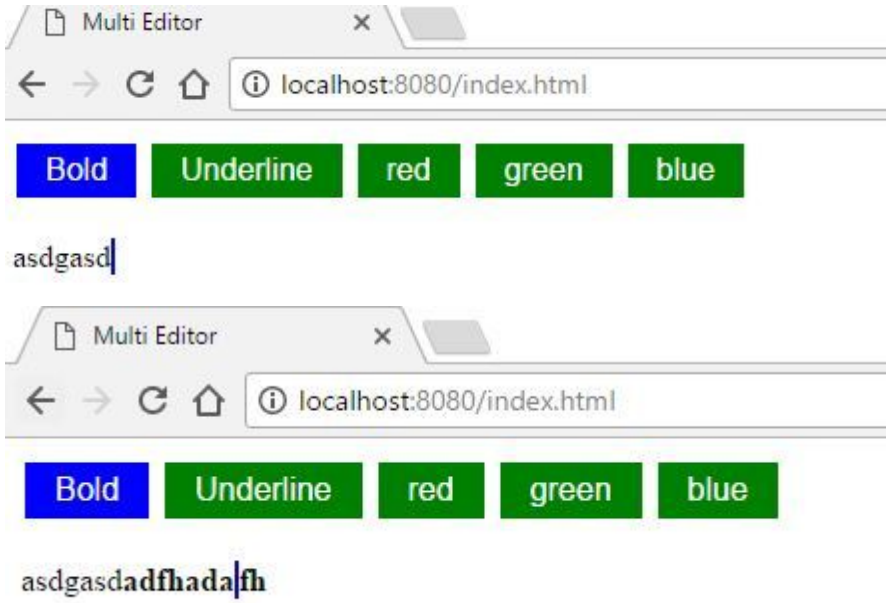
לדוגמא:

שימוש בתוכנה

כאשר אתה מתחבר אל הסרבר ולא נכתב בו כלום יוצג המסך הבא.



כשהמשתמש יחל להקליד במקלדת, האותיות יוצגו אל המסך. בנוסף ניתן ללחוץ על כפתורי העיצוב על מנת לעצב את הטקסט.



על מנת להפסיק את השימוש בעיצוב הטקסט, יש ללחוץ פעם נוספת על הכפתור המודגש. כאשר הכפתור אינו מודגש, הטקסט יכתב כרגיל, ללא עיצוב.

תוכניות עתיד

ניתן להוסיף אפשרויות אל התוכנה, על מנת לשפר אותה, כגון:

- תמיכה בכמה סמנים תשפר את התוכנה, תוכל לדעת איפה מקליד כל אחד מן המשתמשים האחרים ובנוסף תוכל לדעת כמה אנשים מחוברים אל הסרבר, דבר המשפר את חווית השימוש בתוכנה.

- תמיכה בכמה עיצובי טקסט במקביל, ועיצובים נוספים. כרגע התוכנה אינה מאפשרת יותר מעיצוב טקסט אחד ברגע נתון, כלומר, לא ניתן לכתוב בטקסט מודגש אדום לדוגמה. תוספת זו תאפשר ליצירת מסמכים יפים יותר ונאים יותר לעין כאשר עובדים על משהו חשוב.

- שמירת המסמך בסרבר. כרגע כאשר הסרבר מתאפס, גם כן המסמך, דבר הגורם לאיבוד העבודה על המסמך הנוכחי כאשר הסרבר נסגר. ניתן אפילו להוסיף גרסאות שמורות, ובמידה ונעשתה טעות או שינוי לא טוב, ניתן לחזור לגירסא ישנה יותר של המסמך.

- אבטחה. במצב הנוכחי, כל אדם המחובר ברשת, בעל הכתובת של הסרבר יכול להיכנס ולשנות את הקובץ כרצונו. ניתן להוסיף סיסמא למסמך מסוים וכך אדם ללא סיסמא אל הקובץ אינו יוכל לפתוח את הקובץ ולשנותו.

פרק אישי

הפרויקט שלי הוא פרויקט המדמה google docs, מסמך טקסט שניתן לערוך אותו במקביל. כאשר היה צריך לבחור פרויקטים לא עלה במוחי רעיון, ולכן המנחה עזר לי לבחור את הפרויקט שלי. הפרויקט שבחרנו הייתה משחק מחשב של הכה את החפרפרת שמשלב רשתות. לאחר חודש הבנתי שאיני מתלהב מן הפרויקט והייתי צריך לבחור פרויקט חדש, וכאשר עברתי על כל האפליקציות שלי ראיתי את אפליקציית google docs, ואהבתי את האתגר שהצבתי בפני.

להכין את הפרויקט כלל קשיים רבים, הרבה עבודה לבד, ולמידה עצמית, כאשר אני מקבל הכוונות כלליות מן המנחים אך נאלץ ללמוד את הרוב לבדי, דבר שתורגלנו אליו כבר בתחילת הלימודים של לימודי הסייבר, אך כעת הוא היה במסגרת גדולה יותר בעל משקל גדול יותר. עבודתי העצמית בתחילת העבודה על הפרויקט הייתה איטית ומייגעת, והיה מאוד קשה להגיע אל משהו קטן שעובד ונותן רעיון לאיך לבצע את הפרויקט כך שהוא יעבוד. בכל פעם שהתחלתי לעבוד נתקלתי בעוד קשיים ועוד באגים שלא חשבתי עליהם לפני, וכל דבר כזה הוסיף עוד הרבה זמן עבודה, דבר שהיה מאוד קשה לבצע במקביל לחיים החברתיים והלימודיים במקצועות הנוספים.

בערך חודש וחצי לפני הגשת הפרויקט התחלתי לשבת המון זמן על הפרויקט ו"לתת גז" על מנת לסיים אותו עד לתאריך ההגשה ולתת לעצמי אפשרות לסיים את שלושת שנותיי במגמה, ואם תמיכה רבה מן המנחים, חברים ומשפחה, הצלחתי להגיע לפרויקט עובד שאני גאה בו וחושב שבזמן שהיה לי נתון עשיתי את המקסימום והגעתי לפרויקט טוב.

דבר שהמגמה עזרה לי להבין זה את היכולת שלי לשבת וללמוד משהו אם זה מעניין אותי, ולא לוותר למרות שניראה שאין אפשרות לסיים.