# Distributed File Search


שם: ולריה חוטוליוב

ת.ז: 328918966

מקצוע: הגנת סייבר

מנחים: שרית לולב

אלון בר-לב

# תוכן עניינים

# מבוא

בימינו כמות המידע אשר נדרשת בארגונים או חברות היא עצומה. צורת האחסון המועדפת על פני שרת מרכזי היא שימוש בכמה שרתים , אחסון מבוזר. לאחסון המידע בצורה מבוזרת כמה יתרונות: לא תלויים במגבלת נפח האחסון של שרת מרכזי, הקטנת הסיכון לאיבוד כל המידע במקרה של תקלה בשרת , התאמה לביזור הידע (הידע לא נמצא במקום אחד).

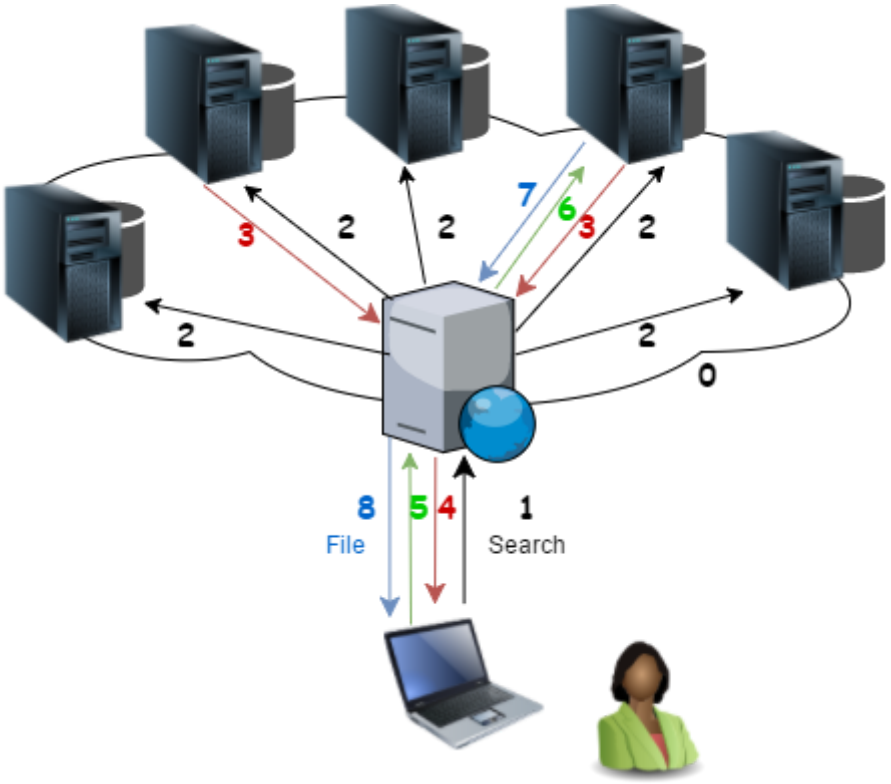
פרויקט זה נועד לאפשר שמירה מבוזרת של המידע על פני שרתים שונים עם ניהול חיפוש קבצים נוח וקל.

מרכיבי המערכת הן: שרתי קצה node servers המכילים קבצים ושרת front end אליו מתחבר המשתמש לחיפוש קובץ במערכת המבוזרת.

למערכת ממשק קל ונוח דרך הדפדפן המאפשר למשתמש לחפש קובץ במערכת המבוזרת לפי שם או חלקי ולקבל את הקובץ לצפייה או להורדה.

# ארכיטקטורה

## Architecture Diagram




## הגורמים במערכת

- Front-end server - שרת העובד מול המשתמש ומטפל בבקשותיו
  - רישום שרתי השרשרת (שרתי קבצים)
  - קבלת בקשת חיפוש - ושליחתה לשרתי השרשרת וטיפול בתוצאות
  - הורדת קובץ
  - צפייה בקובץ

- Node servers - שרתי שרשרת המכילים את הקבצים ומטפלים בבקשות בשרת ה-front-end

- חיפוש שמות הקבצים המכילים את בקשת החיפוש
- נתינת תוכן הקובץ המבוקש

- המשתמש

| Stage | Description |
|---|---|
| 0 | כל node server שולח כל 2 שנות את כתובתו ואת הפורט אליו מחובר. כל 2 שניות ה- front server מאזין ובונה מילון של ה - node servers עם הכתובות והפורטים.<br>כל node server ממפה את שמות הקבצים הנמצאים אצלו. |
| 1 | המשתמש מכניס את מילת החיפוש ב browser. |
| 2 | ה- front server שולח את מילת החיפוש לכל ה- node servers. |
| 3 | node servers שולח את כל שמות הקבצים המכילים את מילת החיפוש ל- front server. |
| 4 | ה- front server שולח ל- browser (למשתמש) טבלה עם כל הקבצים המכילים את מילת החיפוש. |
| 5 | המשתמש בוחר איזה קובץ מעניין אותו , בוחר אם ברצונו לצפות בקובץ או להורידו. |
| 6 | ה- front server שולח ל- node server המתאים את הבקשה המתאימה לבקשת המשתמש (הורדה/צפייה) |
| 7 | ה- node server שולח ל- front server את הקובץ המתאים. |
| 8 | ה- front server שולח את הקובץ ל- browser להורדה או לצפייה. |

# רקע תיאורטי

Multicast - דרך להעברת הנתונים, כאשר המידע מועבר לקבוצות של מחשבים בעלי אותה כתובת multicast ואותו פורט. דוגמא מחיי היום-יום להעברת נתונים בדרך זו, היא שליחת מייל לקבוצה מסויימת. כמו כן, וידאו טרנסלציה גם משתמשת ב muticast.
multicast ממומש על ידי UDP. הסיבה לכך שלא ממומש על ידי TCP היא שTCP דוגל בחיבור ישיר (נק' לנק'), וUDP אינו דורש חיבור ישיר, וגם אינו דורש הודעה על כך אם ההודעה הגיעה או לא, לכן להעברת נתונים בקבוצות נעדיף את החיבור הלא בטוח במקום חיבור הדורש שכולם יהיו מחוברים. אנו נרצה להשתמש בmulticast כאשר לא נדרשת לנו העברה אמינה של נתונים והתוכנית תעבוד כמו שצריך גם אם הנתונים לא יועברו או שאחד הקצוות נותק.
בתוכנית שלי מומלץ להשתמש בדרך זו של העברת הנתונים כאשר אני רוצה לדעת איזה שרתי שרשרת קיימים אצלי, והתוכנית תמשיך לרוץ גם אם אחד יתנתק.

חיפוש מבוזר - זהו חיפוש, המאפשר חיפוש קבצים המפוזרים במס' מחשבים, ממחשב יחיד ושימוש בקבצים אלו ממנו.
כיום ישנו שימוש בחיפוש זה ב google cloud storage.

## טכנולוגיה

xml - שפת תגיות המשמשת לייצוג נתונים ושליחתם בדרך יעילה ונוחה. בנוסף, גם משמשת כתבנית לשפות תגיות אחרות כגון HTML.

html - שפת תגיות המשמשת לעיצוב דפי אינטרנט ותוכן המוצג בדפדפן.

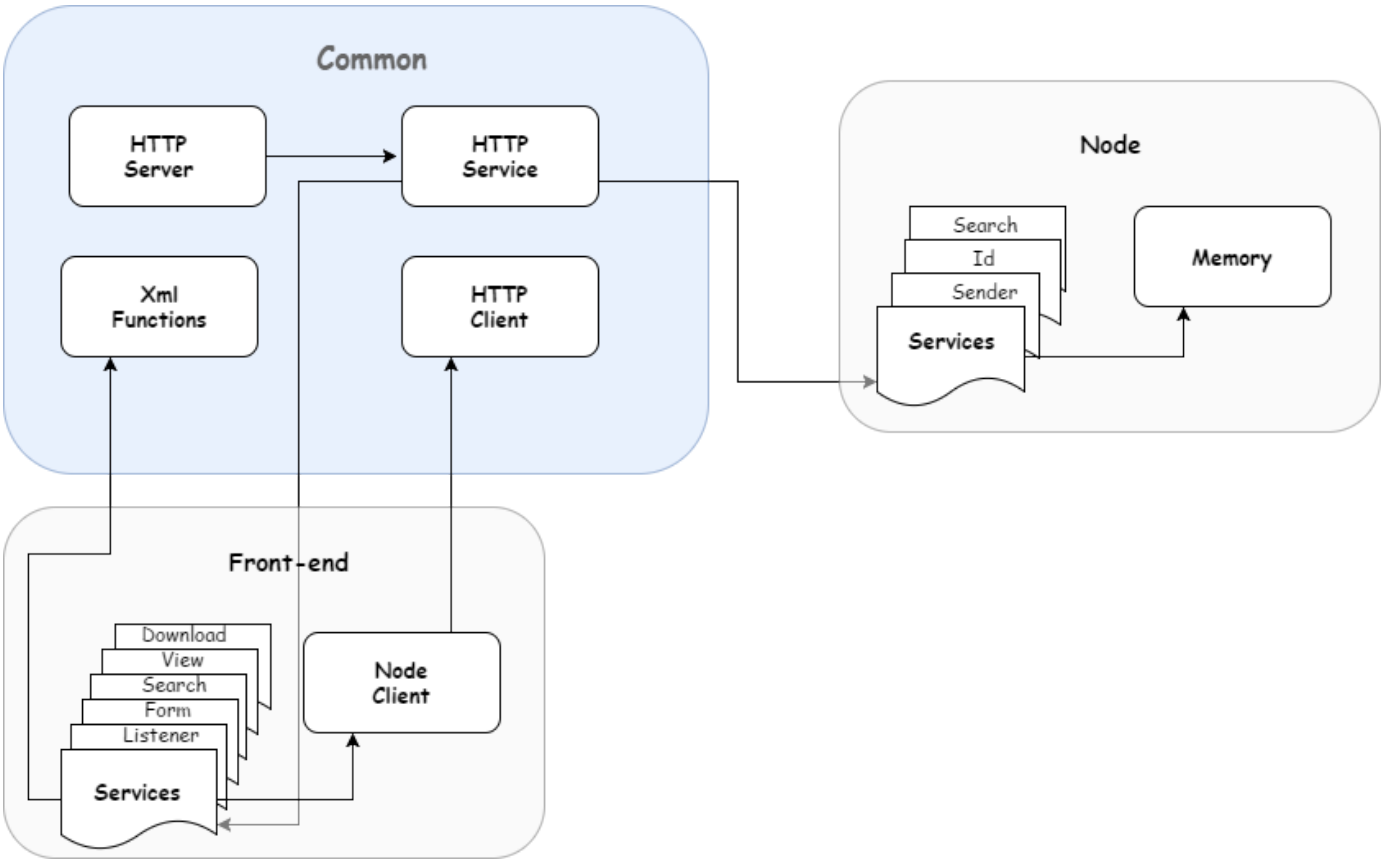
פרוטוקול http - פרוטוקול תקשורת המשמש להעברת דפי html ואובייקטים של מדיה, כגון תמונות וכו'.

tcp - פרוטוקול בתקשורת נתונים, כל העברות הנתונים בפרוטוקול זה אמינות, ישנו אישור שלקבלת נתונים במלואם והחבילות המידע מגיעות באותו סדר בו נשלחו. התקשורת בין התחנות היא באמצעות חיבור מקושר.

udp - פרוטוקול בתקשורת נתונים המאפשר מעבר נתונים שאינו אמין (אין אישור על הגעת הנתונים). כמו כן, החבילות בפרוטוקול זה אינן מגיעות בהכרח בסדר שליחתן.

# מימוש

## Block Diagram




**common** - מודול שבו ישנן פונקציות וקלאסים שגם ה- front server וגם ה- node server משתמשים בהם.

- HTTP server - שרת על פרוטוקול HTTP. השרת משתמש ב class של השירותים של הפרוטוקול, על ידי כך שקורא לעצמים השונים וכך עובד בשירותים השונים שהתוכנית מבצעת.
- HTTP Service - שירותים שהתוכנית נותנת, class זה הוא משמש רק כעזר לשרת על מנת לגשת לשירותים השונים של התוכנית.

- Xml functions - מכיל את הפונקציות המתעסקות עם xml.
- HTTP Client - אחראית על התחברות לשרת HTTP.

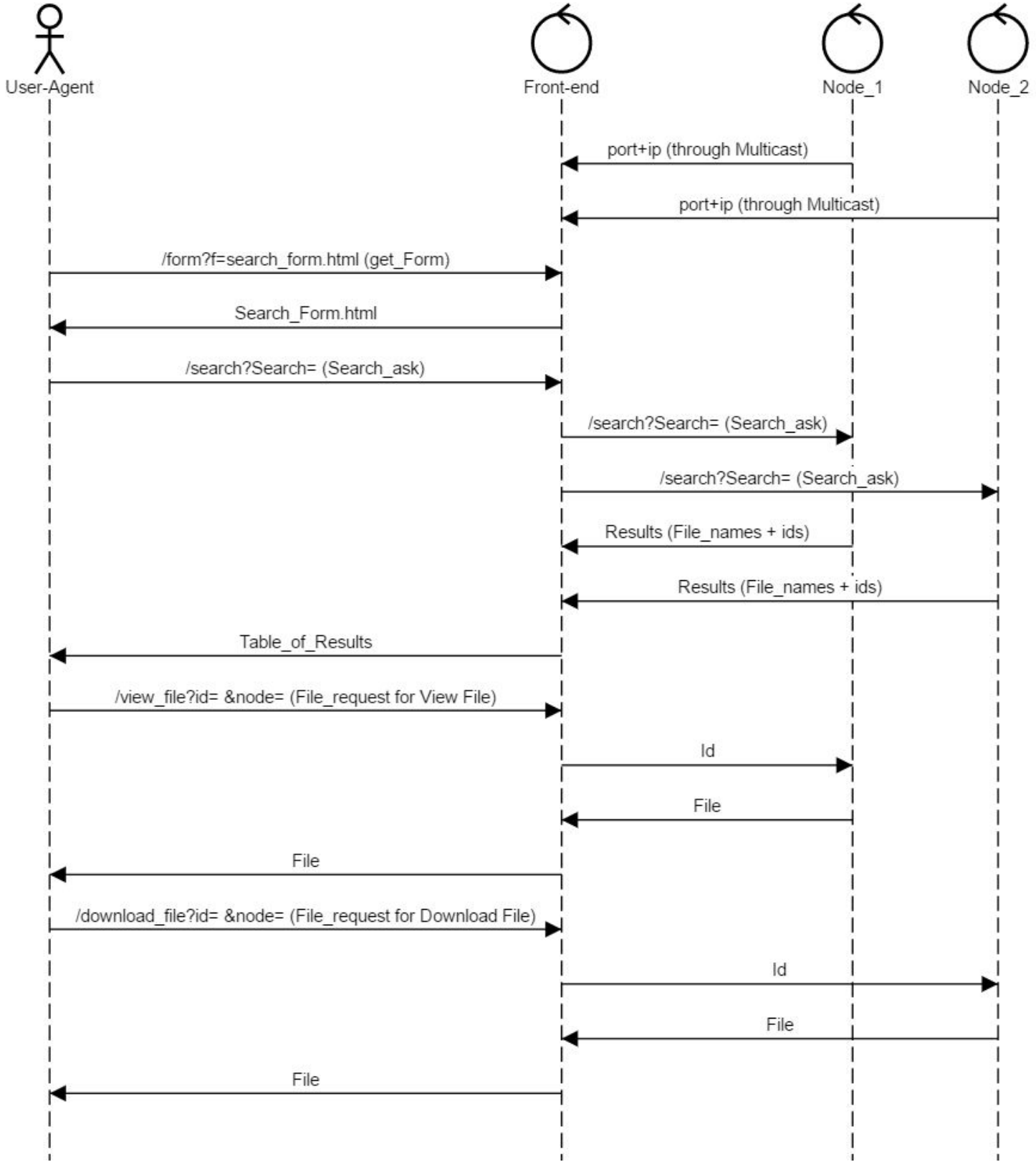
**front-end** - מודול המכיל את כל הפונקציות שהשרת שעובד מול המשתמש צריך.

- services - מכיל קלאסים של השירותים שה front server אמור לספק. כל הקלאסים שבתוכו יורשים מ- HTTP Service על מנת שהשרת HTTP יוכל לגשת אליהם.
- node client - על ידי שימוש ב-HTTP Client ומתחבר ל שרתי השרשרת.

**node** - מודול המכיל את כל הפונקציות שהשרת שרשרת צריך.

- services - מכיל קלאסים של השירותים שה node server אמור לספק. כל הקלאסים שבתוכו יורשים מ- HTTP Service על מנת שהשרת HTTP יוכל לגשת אליהם.
- memory - אחראית על מיפוי הזיכרון של המחשב ולחיפוש הקבצים בתוכו.

## Sequence Diagram



כל שרת node שולח בmuticast את הפורט אליו מחובר.

לאחר מכן המשתמש שולח בקשה לשדה החיפוש לשרת הראשי, אשר מחזיר לדפדפן קובץ html.

המשתמש שולח לשרת הראשי את בקשת החיפוש והשרת מעביר אותה לכל שרתי השרשרת.

שרתי השרשרת מעבירים את תוצאות החיפוש לשרת הראשי אשר מארגן אותן בטבלא ומחזיר לדפדפן המשתמש. בשלב הזה המשתמש בוחר קובץ ואם ברצונו להוריד אותו או לצפות בו בדפדפן, השרת הראשי מעביר בהתאם לזאת בקשה לשרת שרשרת המתאים, אשר מחזיר לו את תוכן הקובץ.

## מימוש החיפוש

כל שתי שניות שרתי הקבצים (שרשרת) שולחים ב muticast את פורט החיבור שלהם.

כל שתי שניות השרת הראשי שעובד מול המשתמש, מאזין ל muticast ומחדש את המילון בו שמורים אצלו כל שרתי השרשרת.

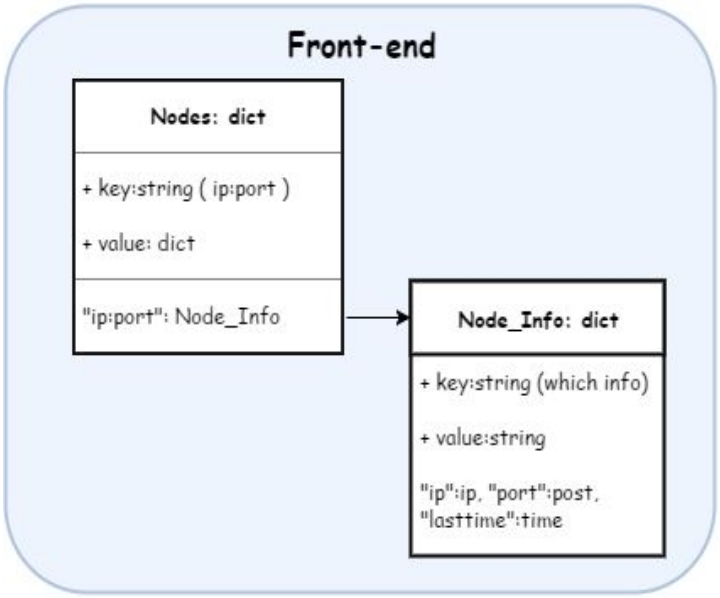
החידוש מתבצע כך שכל שרת קבצים מאופיין בצורה ip:port והמידע עליו הוא כתובתו, הפורט אליו מחובר, וזמן התגובה האחרון. לאחר חידוש המילון, ישנו מעבר על המילון ובדיקת הזמנים של כל שרת שרשרת, כל שרת שלא הגיב כ-10 שניות או יותר, נמחק מהמילון.

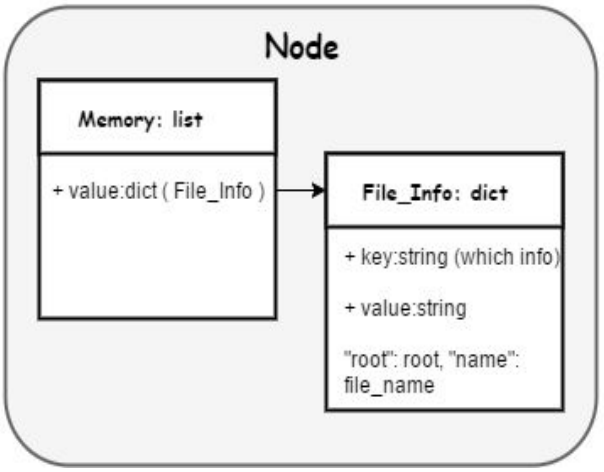
**החיפוש עצמו**

המשתמש מחיל את השימוש בחיפוש על ידי בקשה לשדה חיפוש.
לאחר שהשדה נפתח, המשתמש יכול להקליד את בקשת חיפושו. לאחר מכן, במידה וישנה תקלה, ואין שרתי קבצים, תוצג נודעת מתאימה. אם אין שום תקלה, השרת הראשי יעביר את הבקשה לשרתי השרשרת, והן יחזירו לו xml ובו שמות הקבצים המכילים את בקשת החיפוש ואת מס' הזיהוי שלהם.
השרת הראשי יטפל בתוצאות ויציג למשתמש טבלא.
מתוך הטבלא, המשתמש יוכל לבחור קובץ שמעניין אותו, ולבחור מה ברצונו לעשות - לצפות בקובץ בדפדפן או להורידו.
בשני המקרים השרת הראשי ישלח לשרת השרשרת המתאים את מס' הזיהוי של הקובץ, אך שמות השירותים יהיו בהתאם לרצון המשתמש.
בשלב זה שרת השרשרת יקח את הקובץ מזכרונו וישלח לשרת הראשי את התוכן, ואם הודעת http מתאימה (הודעה לצפייה או להורדה) השרת יעביר את התוכן לדפדפן.

# מבני נתונים

Data Diagram




## פרוטוקולי תקשורת

הפרוטוקול הראשי הוא פרוטוקול HTTP

בקשת http

GET Service HTTP/1.1
Host: Url+uri

העברת הנתונים מתבסס בנוסף על שפת XML

## Front-end

Form service

הסבר

השירות פותח שדה חיפוש בדפדפן

בקשה

/form?f=search_form.html

תשובה

קובץ html

Search service

הסבר

מקבל בקשת חיפוש, שולח לשרתי השרשרת

מכין טבלת html מהתוצאות שהתקבלו משרתי השרשרת.

בקשה

/search?Search=searchAsk

תשובה

```
<html>
<head>
</head>
<body style="background: rgb(255,255,255)">
<center>
<td align="middle" ;="" style="background-color:white">FILE_NAME</td><a
href="/form?file=search_form.html">
&lt;Back&gt;</a><table style="width:35%" ;="" border="2px solid #dddddd">
```

```
<tbody><tr>
<th align="left" ;="" style="background-color:aqua"> Filename
</th>
<th align="left" ;="" style="background-color:aqua">Option</th>
</tr> <tr>
<td align="middle" ;="" style="background-color:white">
<a href="/download_file?id=6&amp;node=10.0.0.1:8070">
&lt;download&gt;</a>
<a href="/view_file?id=6&amp;node=10.0.0.1:8070">&lt;view&gt;</a>
</td>
</tr></tbody></table>
</center>

</body>
</html>
```

View service

הסבר

מקבל id וnode, שולח את ה id לשרת השרשרת המתאים

מקבל את תוכן הקובץ המבוקש ומציג אותו בדפדפן

בקשה

/view?id=0& node=ip:port

תשובה

תוכן הקובץ

Download service

הסבר

מקבל id וnode, שולח את ה id לשרת השרשרת המתאים

מקבל את תוכן הקובץ המבוקש ומוריד אותו למשתמש

בקשה

/download?id=0& node=ip:port

תשובה

תוכן הקובץ

**Node**

Search service

הסבר

מקבל בקשת חיפוש ומחפש במיפוי הזיכרון שלו

את כל הקבצים המכילים בתוכם את הבקשה

בקשה

/search?Search=searchAsk

תשובה

```
<root>
<result id="0" name=" " />
<result id="5" name=" " />
</root>
```

### Id service

הסבר

מקבל id ומחזיר את תוכן הקובץ הנמצא במקום ה id
במיפוי הזיכרון

בקשה

/id?id=0

תשובה

תוכן הקובץ

# בעיות ידועות

- התוכנית פותחת בדפדפן רק את תוכן הקבצים, כלומר היא אינה תפתח תמונה, אלא את הקוד שלה.
  - דרך פתרון- ניתן לשנות את הודעת http כך שיפתח תמונות.
- כל הקבצים המורדים הם מסוג txt.
  - דרך פתרון - לשמור את שם הקובץ כפרמטר ולהעבירו בהודעת ההורדה

# התקנה ותפעול

- על מנת להריץ את התוכנית נפתח את הcmd
- נעבור לתיקייה בה נמצאת תיקיית הפרויקט Dseach

cd C:\folder_name

- נריץ את ה front server

python -m Dsearch-master.front --params

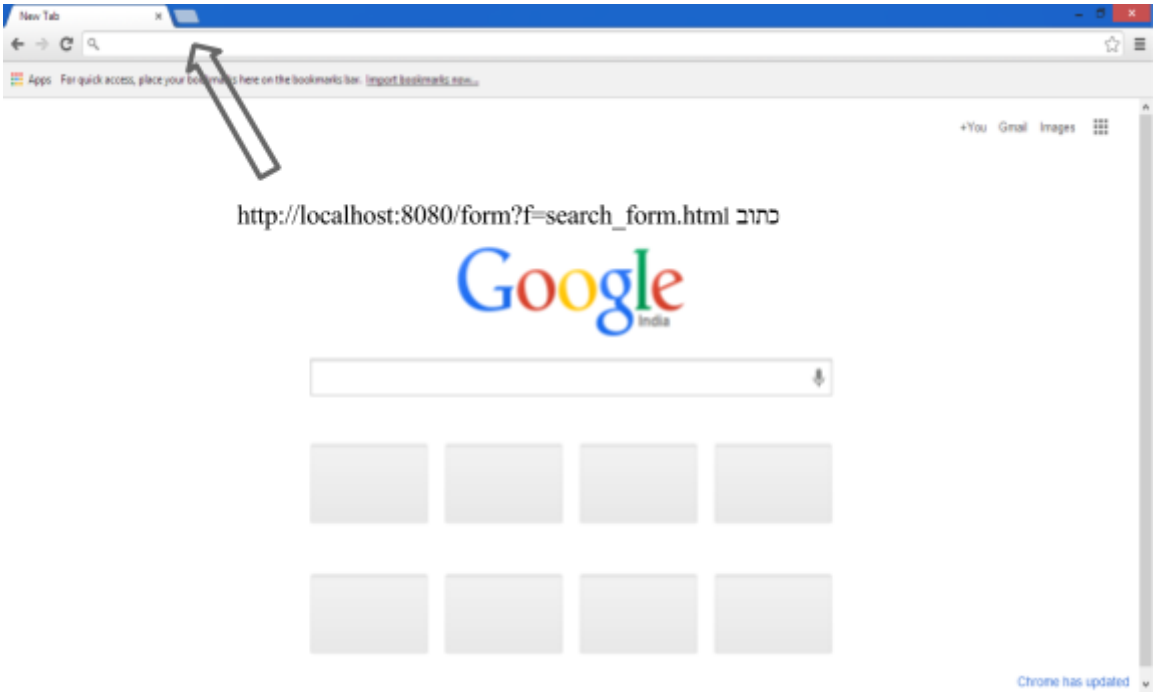
  - Dsearch-master התיקייה בה נמצא הפרויקט
  - --url ( http://localhost:8070/\): כתובת הדפדפן למשל
  - --bind-port פורט אליו נרצה להתחבר
  - --bind-address 0.0.0.0 -לרוב כתובת אליה נרצה להתחבר
  - --base מקום ממנו נרצה לייבא קבצים - לרוב אין צורך לכתוב

- נריץ את ה node server

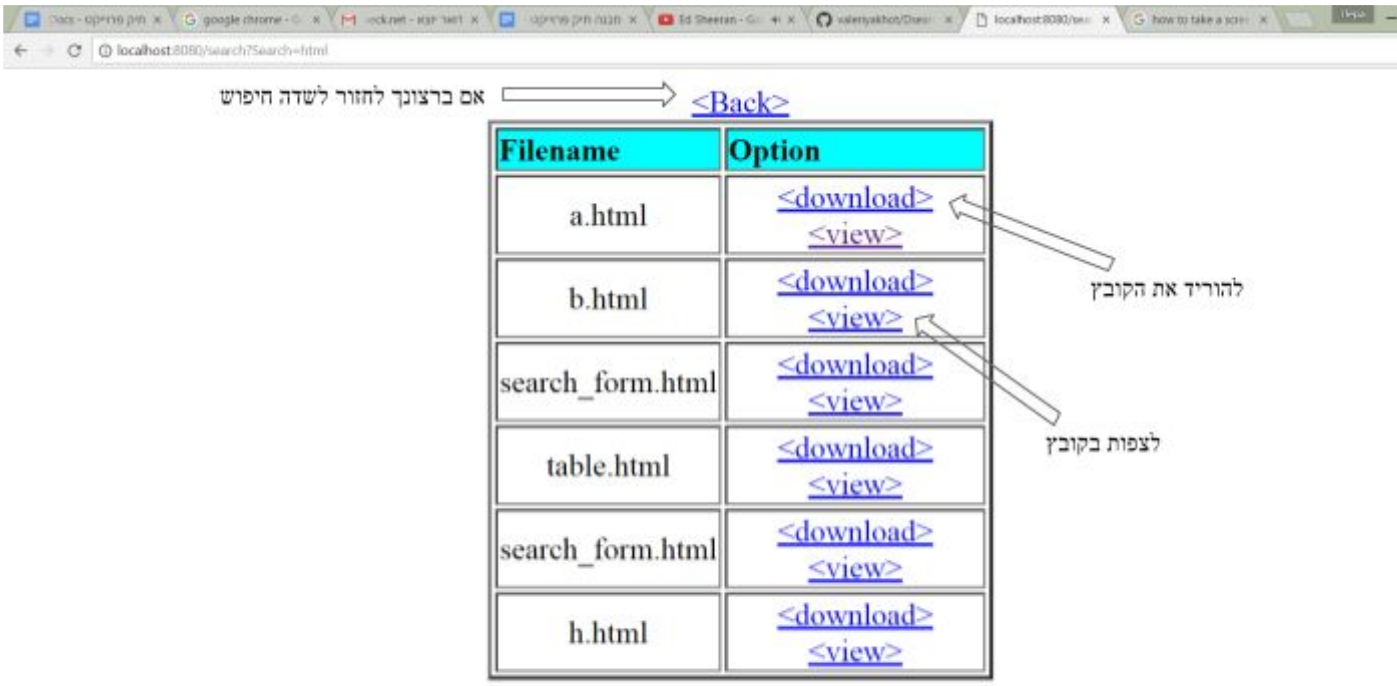
python -m Dsearch-master.node --params

  - Dsearch-master התיקייה בה נמצא הפרויקט
  - --bind-port פורט אליו נרצה להתחבר
  - --bind-address 0.0.0.0 -לרוב כתובת אליה נרצה להתחבר
  - --directory מקום ממנו נרצה למפה את הזיכרון

- נכנס ל google chrome




localhost:8080/form?f=search_form.html
Distributed File Search
Type your ask :
Search
רשום בקשת חיפוש
לבסוף לחץ


localhost:8080/search?Search=html
אם ברצונך לחזור לשדה חיפוש
<Back>
Filename
Option
a.html
<download>
<view>
b.html
<download>
<view>
search_form.html
<download>
<view>
table.html
<download>
<view>
search_form.html
<download>
<view>
h.html
<download>
<view>
להוריד את הקובץ
לצפות בקובץ

## תוכניות עתיד

- ניתן לייעל את מנוע החיפוש על ידי הכנת קובץ המכיל את שמות הקבצים בצורה ממוינת
- ניתן להרחיב את החיפוש לתוכן הקבצים

## פרק אישי

במהלך עשיית הפרויקט נתקלתי בקשיים רבים, כגון ניסיון חלוקת הפרויקט לקלאסים ולקשר ביניהם. כמו כן, התקשיתי במימוש ה multicast, ובמהלך הקישור בין בקשות המשתמש לבין שרתי ה node .

כל הפרויקט ומימושו היוו אתגר לא קטן עבורי.

אך בזכות האתגר שהפרויקט היה עבורי, למדתי המון דברים ורכשתי מיומנויות. למדתי לחפש חומרים באינטרנט בצורה יותר יעילה. נוסף על כך, למדתי כיצד להרכיב פרויקט וכיצד לגבור על שגיאות בקוד בצורה יעילה ומהירה. כמו כן, רכשתי מיומנות לתאר את הקוד שכתבתי בדיאגרמות שונות.