בס"ד

מפרט טכני

****

**מגישים:**

**עמית אביב**

**בן תורג'מן**

**ליאור סילמן**

**מרצה: מארק ישראל**

**מבנה הפרוייקט :**

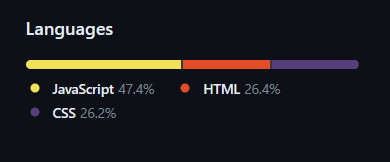
הפרוייקט שלנו מחולק 3 חלקים:

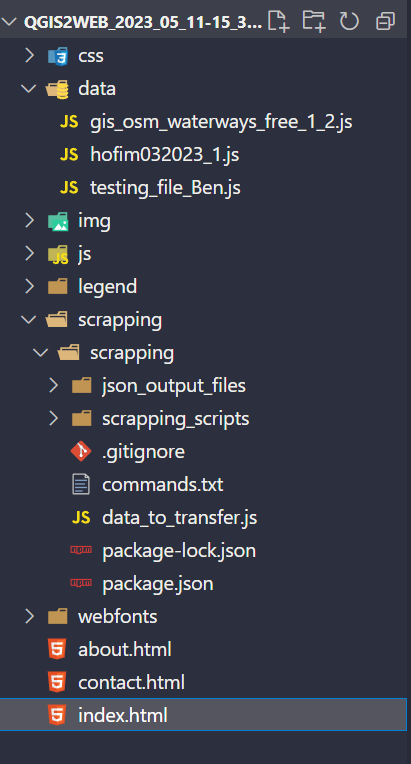
1. האתר(דף נחיתהWEB - ) אשר מכיל בתוכו את המסכים השונים כגון: דף הבית, "קצת עלינו" ו- "צור קשר". דף הבית מכיל בתוכו את המפה שהיא לב הפרויקט

השפות בהם השתמשנו לפרוייקט זה: Javascript ,HTML, CSS

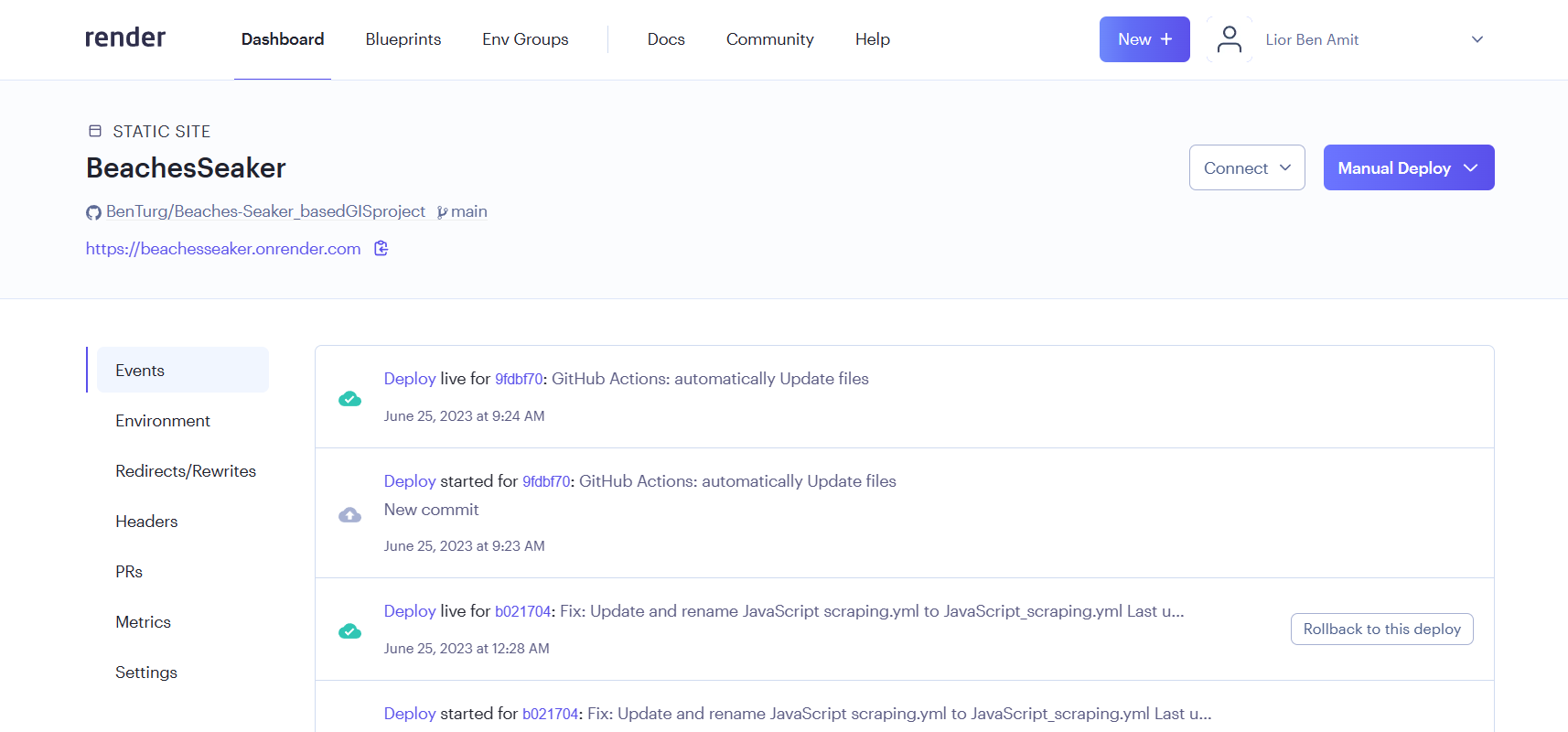
1. המפה (MAP) – אשר מציגה את כל מפת ישראל וחופי ישראל המסומנים כפוליגונים, כל החלק הזה נבנה באמצעות התוכנה של QGIS ולאחר ייצוא לפרויקט HTML הקוד נערך ושונה על ידינו.
2. Backend – למעשה מדובר בכמה רבדים ביניהם:
   1. סקריפט המלקטים מידע מהאינטרנט באמצעות nodeJS וספריית puppeteer
   2. ביצוע deploy והעלאת האתר לרשת באמצעות render.com ו-GitHub
   3. אוטומציה באמצעות GitGub action אשר מריץ פעם ב-6 את הסקריפטים ב-node.js ובאופן אוטמטי render מבצע עדכון לאתר.

כמו שניתן להבין אתר כולו נשמר ב-repository בגיטהאב ומופץ ברשת ע"י render שגם מנפיק url וגם מתעדכן על commit/push ב- repository.

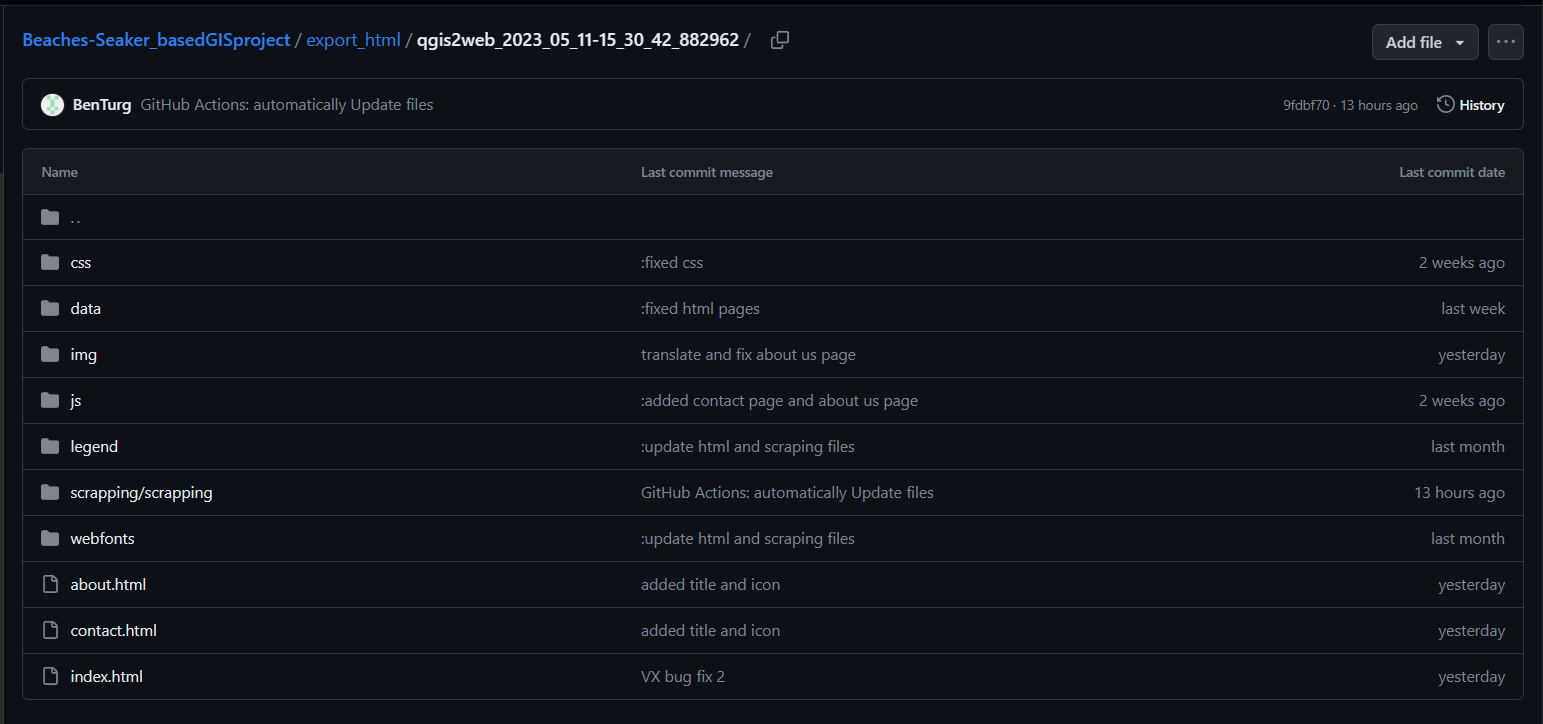
**התפלגות הקוד בפרויקט:**  


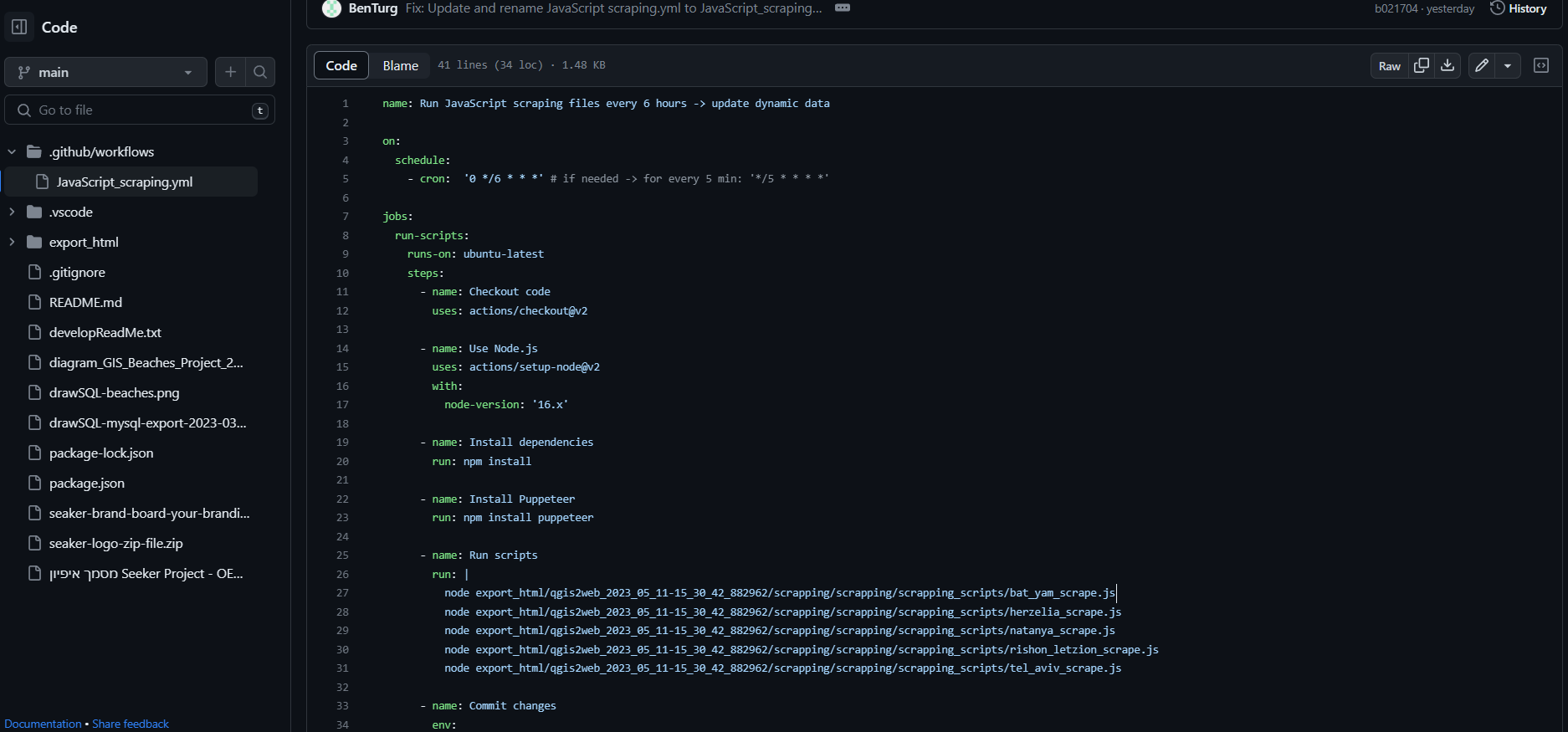
**מבנה הקבצים בפרויקט.**  


**תוכנת הפצה ואחסון האתר – Render.com**



**GitHub Repository**



**CI/CD by GitHub Action  
**

ניתן לראות את האוטומציה שגיטהאב מבצע ע"י הגדרת טכנולוגיות, ספריות, גרסאות וכמובן סקריפט שנכתבו מראש.

**Scrapping**

const puppeteer = require("puppeteer");

const *fs* = require("fs");

const *path* = require("path");

async function scrapedData() {

  try {

    const browser = await puppeteer.launch();

    const page = await browser.newPage();

    await page.goto(

      "https://www.tel-aviv.gov.il/Visitors/Pages/TelAvivBeach.aspx"

    );

    const h3Elements = await page.$$("h3.ng-binding");

    const resultArray = [];

    for (const h3Element of h3Elements) {

      const reversedText = await h3Element.evaluate((node) =>

        node.textContent

          .split("")

          .reverse()

          .join("")

          .replace(/(\d{1,3}) - (\d{1,3})/g, (match, p1, p2) => `${p2} - ${p1}`)

          .replace(/(\d{1,3})-(\d{1,3})/g, (match, p1, p2) => `${p2} - ${p1}`)

      );

      resultArray.push(reversedText);

    }

    await browser.close();

    const exportArray = {

      Height: resultArray[0],

      Temperature: resultArray[2],

      WindDirection: resultArray[3],

      WindSpeed: resultArray[4],

    };

    const reverseArray = {

      beaches\_wave\_height: reverseTextDirection(resultArray[0]),

      beaches\_water\_temp: reverseArrayTemperature(resultArray[2]),

      beaches\_wind\_direction: reverseTextDirection(resultArray[3]),

      beaches\_extra: reverseTextDirection(resultArray[4]), *//beaches\_extra is wind speed and wave direction*

    };

*// Function to modify Temperature format*

    function reverseArrayTemperature(text) {

      const temperatureValue = text.match(/(\d+)\s\*°[CF]/);

      if (temperatureValue && temperatureValue.length > 1) {

        return `${temperatureValue[1]}°C`.replace(":םימה תרוטרפמט", "");

      }

      return text.replace(":םימה תרוטרפמט", "");

    }

*// Reverse function*

    function reverseTextDirection(text) {

      return text

  .split("")

        .reverse()

        .join("")

        .replace("גובה הגלים:", "")

        .replace("כיוון הרוח:", "")

        .replace("מהירות הרוח:", "");

    }

    const output = JSON.stringify(reverseArray);

*// Write the output to a file*

    const filePath = *path*.join(

      \_\_dirname,

      "export\_html/qgis2web\_2023\_05\_11-15\_30\_42\_882962/scrapping/scrapping/json\_output\_files/rishon\_letizon\_output.json",

    );

*fs*.writeFile(filePath, output, (err) => {

      if (err) {

        console.error("Error writing to file:", err);

      } else {

        console.log("Output saved to output.txt");

      }

    });

    console.log(exportArray);

    return exportArray;

  } catch (error) {

    console.log("error");

    console.error("An error occurred:", error);

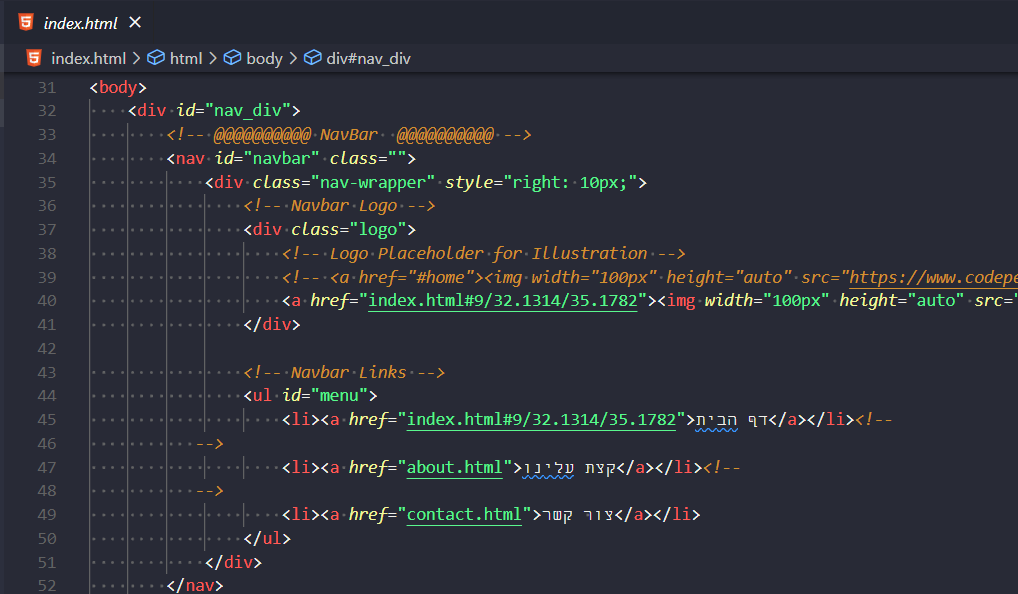
    return null;

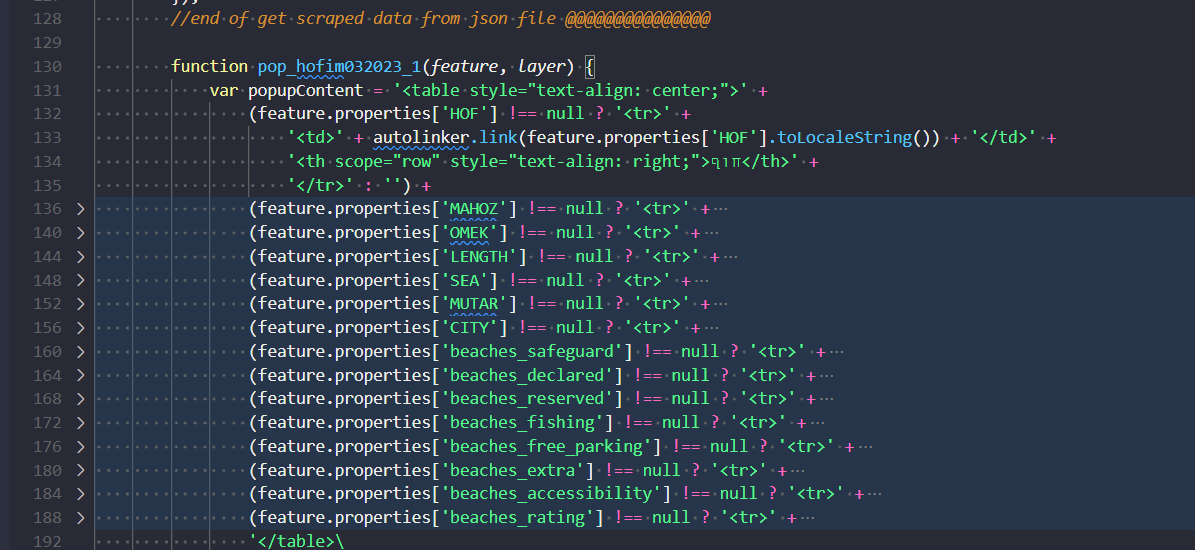
  }

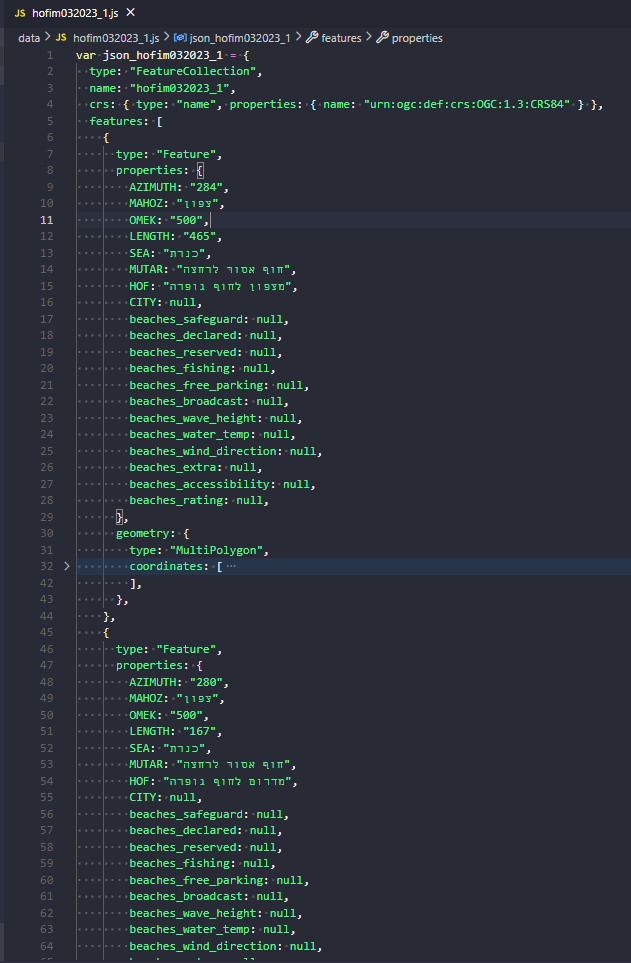
}

scrapedData();

ניתן לראות איך הסקריפט עובד ע"י חיפוש הנתנוים שבחרנו, הכנס למערך וייצוא לקובץ JSON

**Menu bar from index.html (base html)  
**

**יצירת החלון הקופץ ע"י חיפוש במשתנה מסוג אובייקט המכיל את כל הייצוא של הנתונים והחופים**

**המשתנה מסוג אובייקט המכיל את כל הנתונים על החופים (12852 שורות) :  
**