node.js → Sap

node -v ← בודק גירסה

node index.js ← מריץ את הקובץ

```
const fs = require("המודול שרוצים להשתמש")
```

למשל fs (file system מערכת קבצים)

```
fs.writeFile(file, data, callback)
```

- file – מה שנוצר
- data – [illegible]
- callback – הפעולה ← (err) => {

טיפול נכשל

```
(err) => {
  if (err) throw err;
  console.log("good");
```

```
fs.readFile("המסלול לקובץ", "utf-8", (err, data) => {
    if (err) throw err;
    console.log(data);
});
```

npm → node package manager

```
npm install <something>
```

או i

השם של החבילה ← אפשר כמה חבילות

הוא יופיע אחרי זה בקובץ json

תחת הקטגוריה dependencies

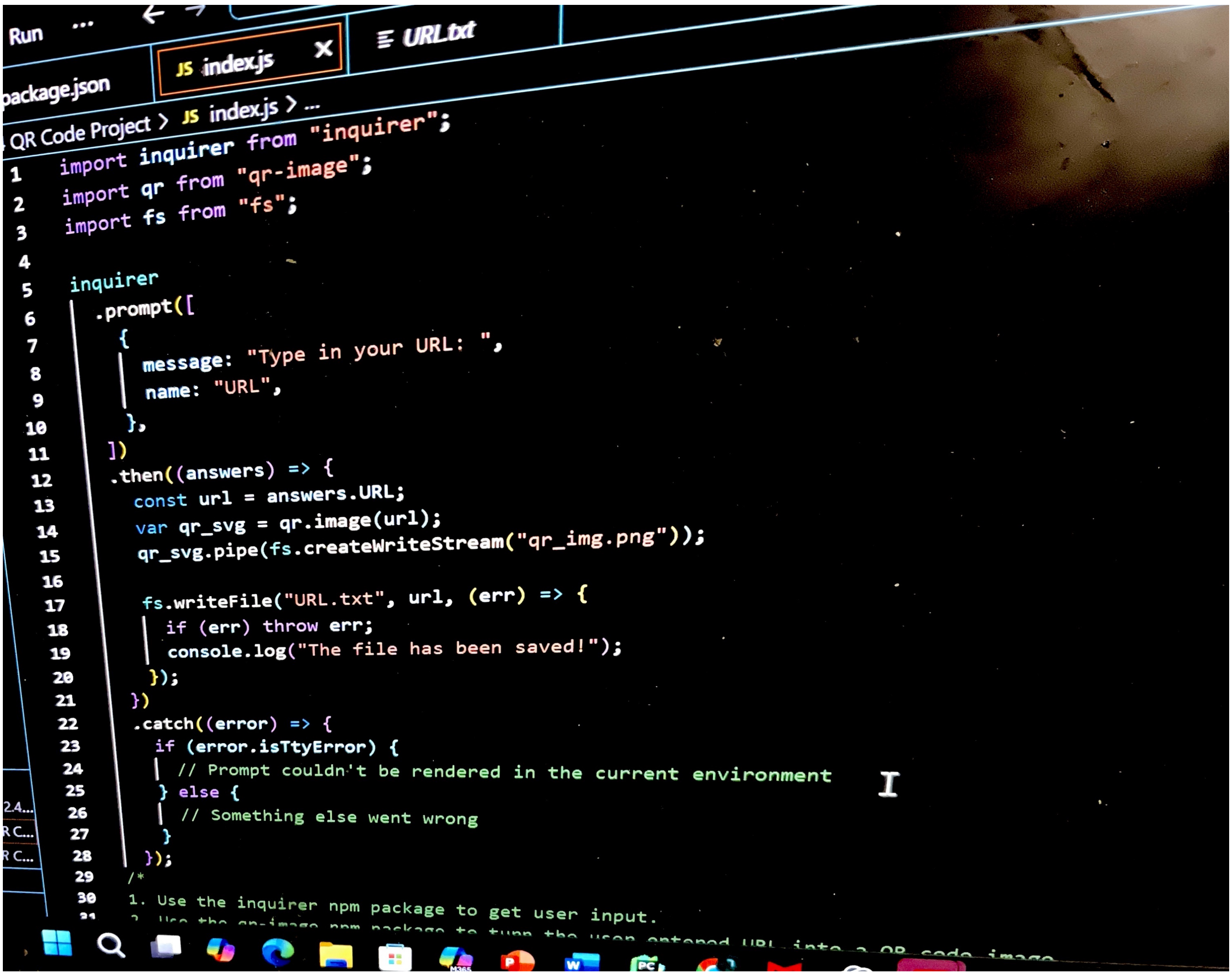
console.log(`My name is ${name}.`)

אפשר להשתמש ב import במקום require

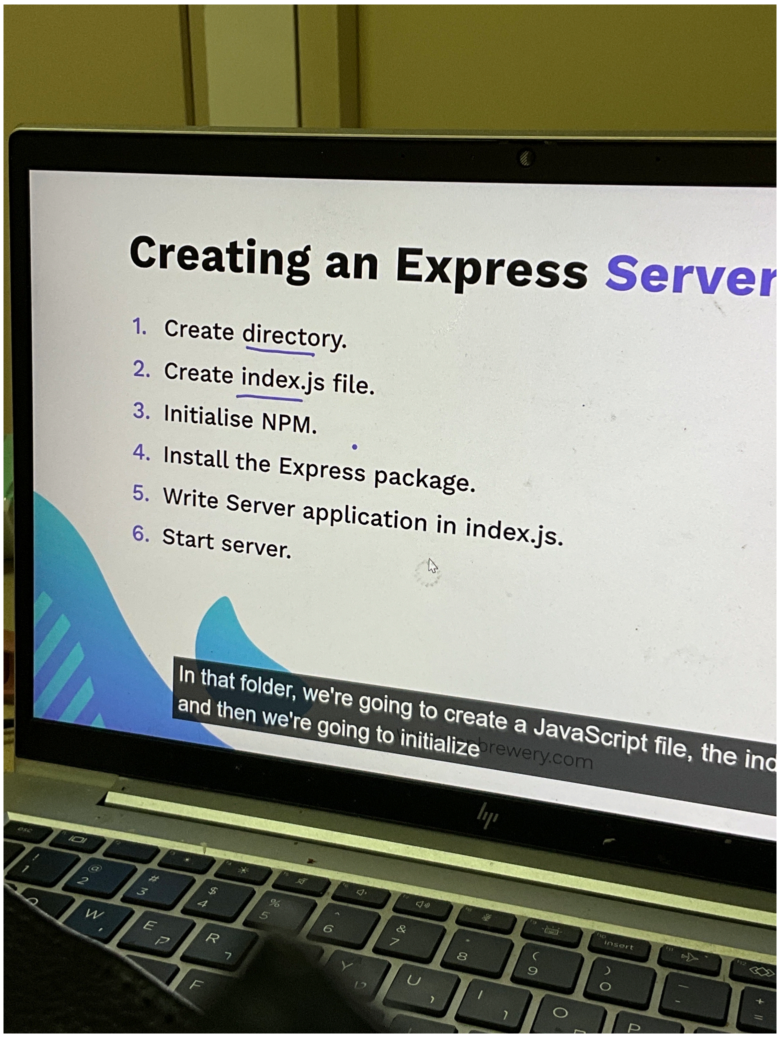
מתחת ל main בקובץ json נוסיף type="module"

ערך של המילה ← qrproj.js

מקום



express



הסבר על השלבים:

1) יצירת ה directory באמצעות ("שם הספריה") mkdir ואז נכנס אליו באמצעות cd

2) יצירת קובץ index.js באמצעות touch

3) -y init npm ← ניצור את הקובץ json

4) express i npm ← התקנה של express לקובץ

ונוסיף "type": "module", אחרי ה main בקובץ json

5) בקובץ ה index.js נרשום:

```
import express from "express";
const app = express();
app.listen(3000, () => {
    console.log('server running on port 3000.');
});
```

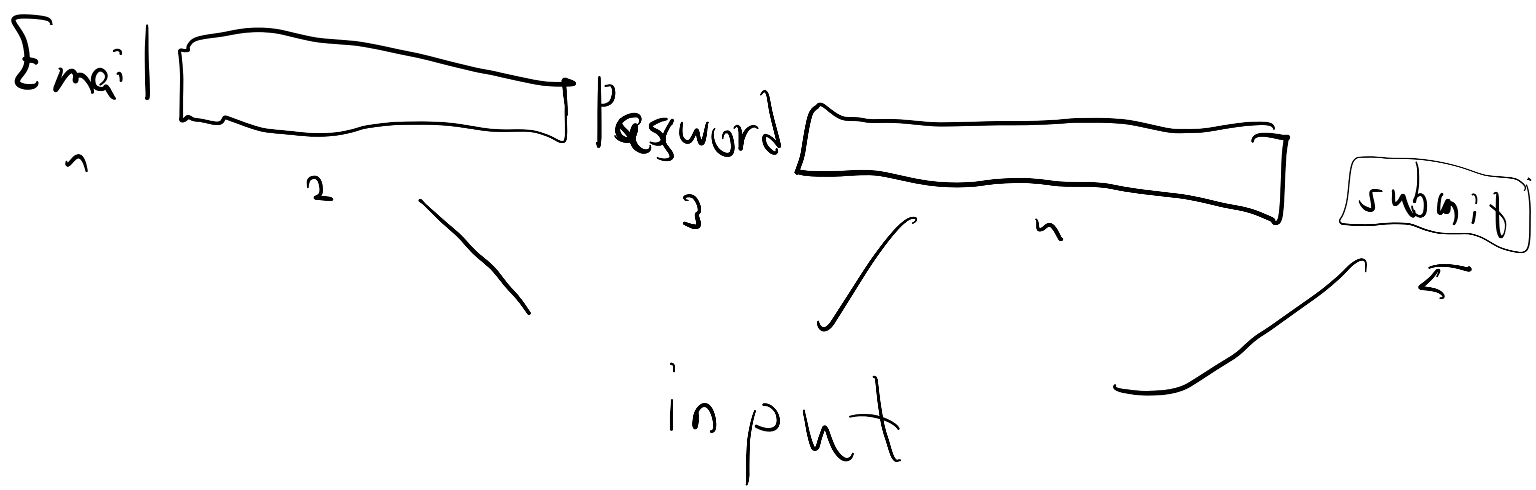
port callback

HTTP	SQL
GET	select
post	insert
put	update
patch	update
delete	delete

Middlewares

```
<form action="/login" method="POST">
    <label for="email">Email</label>
```

```
                                              label>
2 <input type="text" name="email" required>
3 <label for="password">Password</label>
4 <input type="text" name="password" required>
5 <input type="submit" value=Sumit>
</form>
```



מה לעשות בשביל לקבצים לקח html לקוד שמגן כאשר
כשאנו מריצים מהשרת

```
import {dirname} from "path";
import {fileURLToPath} from "url";
const _dirname = dirname(fileURLToPath(import.meta.url));

app.get ...
    res.sendFile(_dirname + "html/ הנתיב");
```

morgan

```
install morgan

import morgan from "morgan";
app.use(morgan("dev"))
```

יש כבר middlewares מוכנים ←

```
app.use((req, res, next) => {
    console.log("Request method: ", req.method);
    next();
});
```

תרגול קוד לפני המבחן

דרישות מערכת

1) ניהול מלאי ירקות

לכל ירק:

שם

קטגוריה יחודי

כמות במלאי

מחיר לקילוגרם

← אפשר להוסיף, למחוק, לעדכן ולחפש לפי שם/id

שליפה של הירק הכי זול/יקר ב O(1)

2) ניהול לקוחות

לכל לקוח:

מזהה ייחודי
שם
מספר טלפון
רשימת רכישות

← הוספת לקוח, חיפוש לפי שם / id, עדכון פרטים
שליפת הלקוחות שביצעו מספר הקניות הכי גבוה

ג) ניהול הזמנות
לקוח יכול לבצע
רשימה וכמויות

הזמנה שכוללת ← בפריטים צריך לבדוק אם
הכמות המבוקשת זמינה?
במלאי > הצלחה
לפי הכמות

ד) דוחות ושאילתות

הפקת רשימה של הירקות שנמכרו הכי הרבה
לקוחות עם הכי הרבה הזמנות
ירקות עם מלאי נמוך

כללי ברזל ל API

5

מעולה, הנה טבלה מסכמת שתעשה לך סדר בראש לקראת המבחן 🚀

📊 טבלת שימוש ב־HTTP Methods

Method	מטרה ראשית	מזהה ייחודי (מי) → params	פרמטרים אופציונליים → query	נתונים חדשים/עדכון → Body
GET	שליפה / קריאה	✅ כן (לדוגמה: vegetables/:id/)	✅ כן (לדוגמה: ?num=5&sort=asc)	❌ אסור (אין body)
POST	יצירה / פעולה חדשה	❌ לרוב לא	❌ לא חובה	✅ חובה (נתוני הישות החדשה)
PUT	עדכון מלא (מחליף אובייקט)	✅ כן (מי לעדכן)	❌ לא חובה	✅ חובה (הנתונים החדשים)
PATCH	עדכון חלקי (שדה אחד/שניים)	✅ כן (מי לעדכן)	❌ לא חובה	✅ חובה (רק השדות לשינוי)
DELETE	מחיקה	✅ כן (מי למחוק)	❌ כמעט תמיד	❌ אין body

📌 דוגמאות קצרות

+ Ask anything

ChatGPT can make mistakes. Check important info.