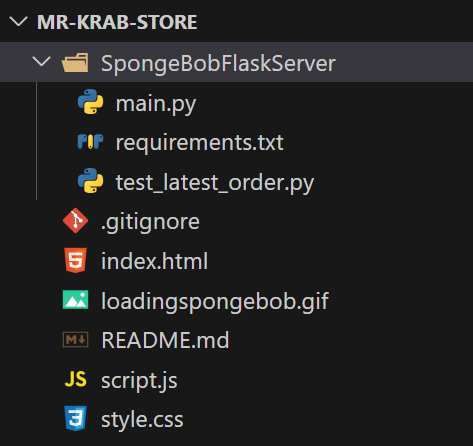
*Async JS Is Easy*

**בובספוג רוצה לנהל חנות ופונה אליכם לעזרה!**

**מטרת התרגיל**

היכרות עם Promise, async/await והתנהגות א-סינכרונית אמיתית. היכרות עם fetch.

**דרישות**

המבנה ההתחלתי שלכם נראה כך:  
  
תחילה יש להריץ את השרת לוקלית.

נפתח טרמינל בנתיב של SpongeBobFlaskServer

יש ליצור venv באמצעות הפקודה python -m venv venv

יש להפעיל את הסביבה הוירטואלית, בווינדוס:  
. 

לאחר מכן יש להתקין את כל הספריות הנדרשות.

pip install -r requirements.txt

על מנת להריץ את השרת יש להריץ את הקובץ main.py

כעת נתחיל עם התרגיל. להלן ההנחיות:

המטרה של התרגיל היא להציג את הידע שרכשתם ב-JS ומימוש פונקציות א-סינכרוניות לצד שימוש בקוד סינכרוני, ויכולת לטפל בבקשות ל-API באמצעות fetch. את המידע הרלוונטי יש להציג ב-DOM.

1 – יש לטעון את התפריט של המוצרים לעמוד, הבקשה תיקח לפחות 5 שניות, השרת מדמה שליפה כבדה מ-DB ולכן עד שהתוכן לא מוחזר יש להציג את ה-gif כל עוד הבקשה pending.

לאחר שנטענו המוצרים התפריט צריך להיראות כך:



2 - כל מוצר צריך להיות מ-0 יחידות עד 5 יחידות בלבד. ההצגה של המחיר צריכה להיות בפורמט של currency.

3 - פירוט ההזמנה צריך להתעדכן בלייב באופן הבא. שימו לב שאין לשלוח את הבקשה עד הלחיצה על הכפתור submit. מבנה כל שורה בסיכום ההזמנה נראה כך: (בדיוק)  


4 – אם הכמות חזרה ל-0 יש להוריד את המוצר מסיכום ההזמנה.

5 – אם יש 0 מוצרים סה"כ יש להציג את הטקסט empty order

4 - הכפתור submit ישלח את הבקשה לשרת ותהיה הודעת הצלחה או כישלון בהתאם.

Get last order יבקש את ההזמנה האחרונה מהשרת. בנוסף, יש להציג בחלון כל שגיאה רלוונטית שחוזרת מהשרת.

5 – יש למיין את המוצרים לפי מחיר לפני ההצגה ב-DOM.­

יש לעשות שימוש גם ב-async await וגם ב-promise-ים עם הסינטקס המלא.