התחברות ל DB

עד עכשיו: sqlite3, "התחברנו" או בעצם פתחנו לקובץ. בעצם רק מפתח אחד יכול להתחבר.

עכשיו יש לנו DB POSTGRES שאי אפשר לפתוח אותו כקובץ.

אפשר רק דרך הרשת, חיבור מרחוק. זה מאפשר להרבה משתמשים להתחבר ל DB הזה.

ה DB רץ בתוך שרת עם כתובת IP, פורט, יוזר וסיסמא.

אם נדע הפרטים האלה (כתובת IP, פורט, יוזר וסיסמא) אז נוכל להתחבר ואז נוכל להריץ SQL.

סיכום ענן

1. Docker
   1. Dockerfile
   2. בניית image
   3. דחיפת ה image ל docker hub
   4. הרצת container מה image
2. Hosting
   1. Pythonanywhere
   2. VM בשירות hosting
      1. EC2
      2. יש הרבה hosting providers שמוכרים שירות של הרצת VM
         1. Godaddy
         2. Digital-ocean
      3. ברגע שיש לנו VM אפשר להתקין שם docker ולהריץ את האפליקציה
3. דגשים:
   1. צריכים מחשב עם כתובת IP ציבורית
   2. Firewall לדאוג שפורט 80 פתוח להריץ שרת ווב (בפרודקשן פורט 443 HTTPS)

שלב הבא: בניית אתר לעסק קטן ב FLASK ופרסומו כ docker image ודחיפה לענן

אפיון:

BACKEND

1. API לשירותים\מוצרים ולהרשמה לאתר
2. API לקביעת פגישות
3. DB
4. בדיקות Unit testing כדי לוודא שהכל תקין לפני שמעלים ל docker

FRONT

1. חלק סטטי HTML, CSS
2. JS שמתחבר ל API ובונה את רשימת השירותים\מוצרים, נרשם, קובע פגישות
3. בונוס: מי שרוצה אפשר ב react

לארוז את כל זה ב docker image ולפרסם כך שירוץ עם כתובת IP ציבורית

פירוט:

BACKEND

לבחור איזה סוג של עסק: שיעורי יוגה, חנות פרחים, חוגים, משרד עורך דין, סוכנות ביטוח...

1. API לשירותים\מוצרים ולהרשמה לאתר

GET /api/services

{

"services":{

"consulting",

"תביעות",

"contacts"

}

}

POST /api/services

{

"services":{

"consulting",

"תביעות",

"contacts"

}

}

POST /api/service

1. API לקביעת פגישות

GET /api/meeting

{

"meetings":{

{"date":"3/3/2024", client:"Levi"},

{"date":"13/3/2024", client:"Cohen"},

}

}

POST /api/meeting

{"date":"3/3/2024", client:"Levi"},

1. DB
2. בדיקות Unit testing כדי לוודא שהכל תקין לפני שמעלים ל docker