# **מערכת Full Stack למידע על חנויות**

**אפיון:**

הצגת מידע לגבי חנויות.

לחנות יש:

* קוד
* שם
* קטגוריה
* כתובת

על המערכת להציג את כל החנויות במערכת ולאפשר חיפוש חנות

בנוסף ניתן להוסיף חנות חדשה.

**Database**

מכיל טבלת קטגוריות המכילה:

* קוד קטגוריה
* שם קטגוריה

מכיל טבלת חנויות המכילה:

* קוד חנות
* שם חנות
* קוד קטגוריה
* כתובת

**Backend**

חושף שלושה Routes:

* החזרת רשימת הקטגוריות
* החזרת רשימת החנויות כאשר שם הקטגוריה מופיע בכל חנות
* הוספת חנות

**Frontend**

* Layout המכיל כותרת, תפריט, אזור ראשי
* דף בית – מכיל פיסקת הסבר ותמונה
* דף צפייה בחנויות + אפשרות חיפוש
* דף הוספת חנות חדשה

**הפעולות לביצוע עבור בניית המערכת:**

**Database**

* הרצת XAMPP
* בניית Database חדש. שם: ShopsDB
* בניית טבלת הקטגוריות
* בניית טבלת החנויות
* קישור בין הטבלאות
* הוספת מספר קטגוריות לטבלת הקטגוריות ע"י הממשק של MySQL
* הוספת מספר חנויות לטבלת החנויות ע"י הממשק של MySQL
* ביצוע Export לכל מסד הנתונים בכדי שנוכל לשלוח אותו יחד עם הקבצים

**Backend**

* פתיחת פרויקט חדש בשם Backend (תיקייה חדשה) ע"י VS Code הנמצא בתוך תיקייה בשם Full Stack
* יצירת app.js
* הוספת package.json ע"י הפקודה npm init
* הוספת "start": "node app" לאובייקט ה-"scripts" ב-package.json
* התקנות הדרושות עבור ה-Backend:

npm i express mysql cors

* יצירת קובץ global.json המכיל את הגדרות ההתחברות למסד הנתונים
* בניית שכבת ה-DAL
* בניית שכבת ה-BLL המכילה את הפונקציות הבאות:
  + getAllCategoriesAsync
  + getAllShopsAsync
  + addShopAsync
* בניית שכבת ה-Controllers המכילה את ה-routes הבאים:
  + GET <http://localhost:3001/api/categories>
  + GET <http://localhost:3001/api/shops>
  + POST <http://localhost:3001/api/shops>
* בניית ה-app.js
* הרצה ובדיקה שאין קריסה
* בדיקת כל Route ע"י Postman ובדיקה שאין קריסה ומקבלים את המידע הדרוש

**Frontend**

* סגירת Postman וסגירת VS Code של פרויקט ה-Backend
* התקנת פרויקט React בשם frontend כך שהתיקייה frontend שנוצרת אוטומטית ע"י React תהיה ליד תיקיית ה-Backend. לצורך כך יש לפתוח את ה-Command-Line על תיקיית ה-Full Stack שמעל תיקיית ה-Backend ולהריץ את הפקודה הבאה:

create-react-app frontend --template typescript

אם חסרה התקנת React, יש להתקין ע"י הפקודה:

npm i -g create-react-app

* פתיחת תיקיית ה-Full Stack ע"י VS Code לצורך פיתוח והרצת ה-Backend וה-Frontend ע"י סביבת פיתוח אחת
* פתיחת טרמינל על ה-Backend והרצה
* פתיחת טרמינל על ה-Frontend והרצה
* בניית העיצוב הכללי של האתר בקובץ index.css
* בניית רכיב ה-Layout ושילוב הרכיב ב-index.tsx

אם חסרה התקנת פקודת create יש להתקין ע"י הפקודה:

npm i -g react-cli-snippets

* מחיקת קבצי רכיב ה-App
* הוספת רכיב ה-Header המכיל כותרת.
* הוספת רכיב ה-Menu המכיל את הלינקים הדרושים (תגיות a)
* הוספת שלושה רכיבים ריקים:
  + רכיב עבור דף הבית – מכיל תת כותרת
  + רכיב עבור הצגת החנויות – מכיל תת כותרת
  + רכיב עבור הוספת חנות חדשה – מכיל תת כותרת
* התקנת ספריית ה-routing של React:

npm i react-router-dom @types/react-router-dom

* הוספת רכיב Routing לאזור ה-Layout ובניית הרכיב הזה הכולל את שלושת ה-Routes ו-Default Route.

אם רוצים אפשר גם Page404. לא לשכוח <BrowserRouter> שעוטף את רכיב ה-Layout.

* חזרה לרכיב ה-Menu והחלפת תגיות ה-a בתגיות NavLink בכדי למנוע גלישה לשרת.
* בניית מחלקת Model עבור ShopModel

אפשר להגדיר "strictNullChecks": false ב-tsconfig.json לצורך מניעת שגיאות על שדות המודל

* התקנת Axios ע"י הפקודה npm i axios
* בניית רכיב ה-ShopList: יצירת ה-State, אתחול ה-State, גלישה לשרת ב-componentDidMount והחזרת כל אוסף החנויות והכנסת המידע ל-State, הצגת המידע בטבלה/card וכו'.
* בניית מחלקת Model עבור CategoryModel
* בניית רכיב הוספת חנות: יצירת טופס המכיל תיבת select עבור הקטגוריה, תיבת טקסט עבור שם החנות, תיבת טקסט עבור כתובת החנות, ב-componentDidMount גלישה לשרת, החזרת רשימת הקטגוריות, הכנסה ל-state, עדכון תיבת ה-select כך שתציג את רשימת הקטגוריות, חיבור ספריית useForm לטופס (התקנה: npm i react-hook-form), בלחיצה על לחצן send – שליחת המידע לשרת.
* בדיקה שכל הפרויקט עובד כמו שצריך, אין שגיאות ב-console וכדומה.