בסייעתא דשמיא

# מיני פרוייקט בבסיסי נתונים – סמסטר ב' שנת תשע"ח

## בחירת הנושא והסביבה

המוטיבציה – קיימים כמה מאגרים אינטרנטיים שבהם ניתן להשיג ספרי קודש מסויימים. רצינו ליצור מאגר שיאפשר לברר האם ספר מסויים קיים באותם המאגרים. במקרה שכן, ניתן לבדוק באיזה מאגרים הוא נמצא, ומהי הזמינות שלו (למשל – בחינם או בתשלום. כטקסט או כצילום).

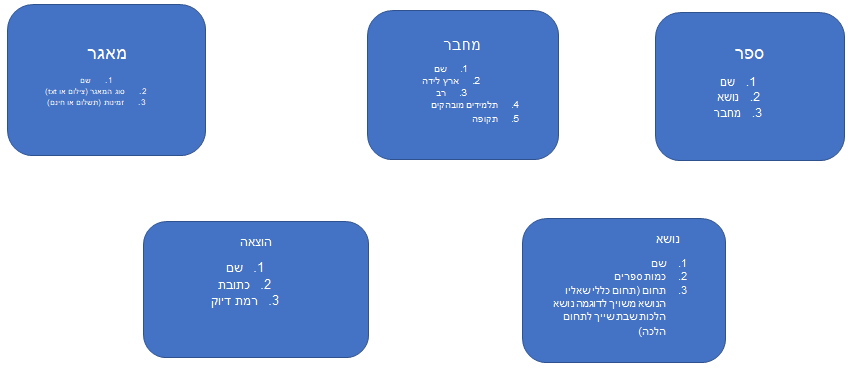
בחרנו ליצור בסיס נתונים שייצג ספרים במאגרי מידע אינטרנטיים – פרטי הספר, פרטי המחבר, פרטים על המאגרים בהם נמצא הספר, ועוד.

## בחירת סביבת העבודה

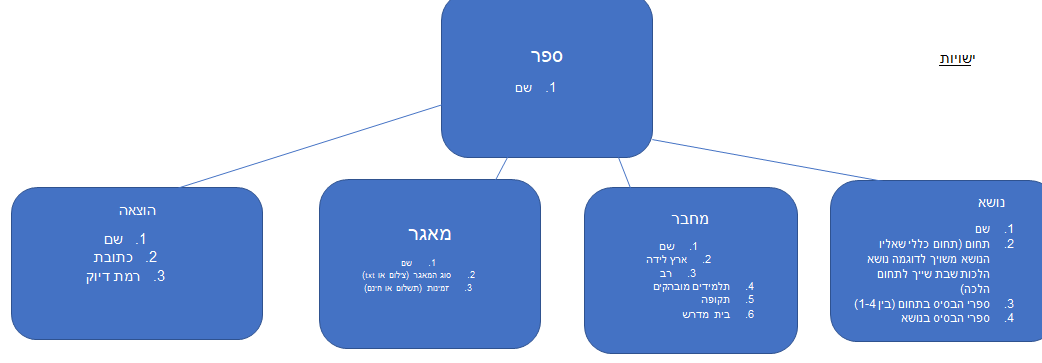
רצינו לעבוד עם סביבה חדשה, שאנחנו פחות מכירים, על מנת לתרגל צורות נוספות ממה שהכרנו עד עכשיו. כיוון שאת סביבת SQL אנחנו מכירים די טוב, רצינו להשתמש בסגנון אחר. החלטנו להשתמש ב Firebase Realtime DataBase, שאותו אנחנו מכירים בצורה בסיסית, אבל רצינו ללמוד אותו טוב יותר. בסיס הנתונים הזה מאחסן את הנתונים בפורמט JSON.

החלטנו ליצור את האינטגרציה איתו ואת הממשק למשתמש בעזרת שפת python, שגם אותה אנחנו מכירים בצורה בסיסית, וכיוון שהיא שפה יחסית פופולרית, רצינו להכיר אותה יותר.

## הגדרת הישויות

ניסינו לחשוב מה הישויות החשובות, והחלטנו קודם כל להגדיר את הישויות הבאות – ספר, מחבר, מאגר, נושא, הוצאה. 

בהתחלה הגדרנו את הישויות כישויות מקבילות, ורצינו לייצר מודל של ישויות- קשרים, כמו בSQL. אבל לאחר בחינת הנושא הזה, ולימוד ההבדלים בין JSON לSQL, החלטנו שהצורה המתאימה היא, ליצור ישות אחת שהיא השורש, וכל שאר הישויות מתייחסות אליה.

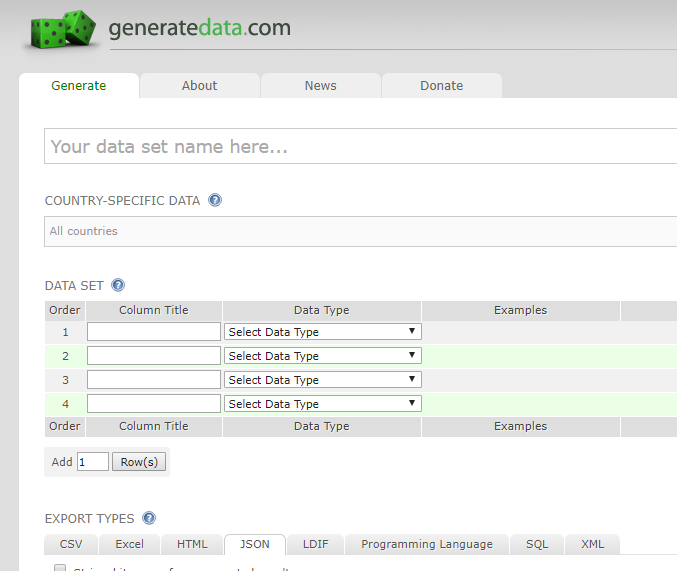


הגדרת הנתונים הסופית היתה כזאת:

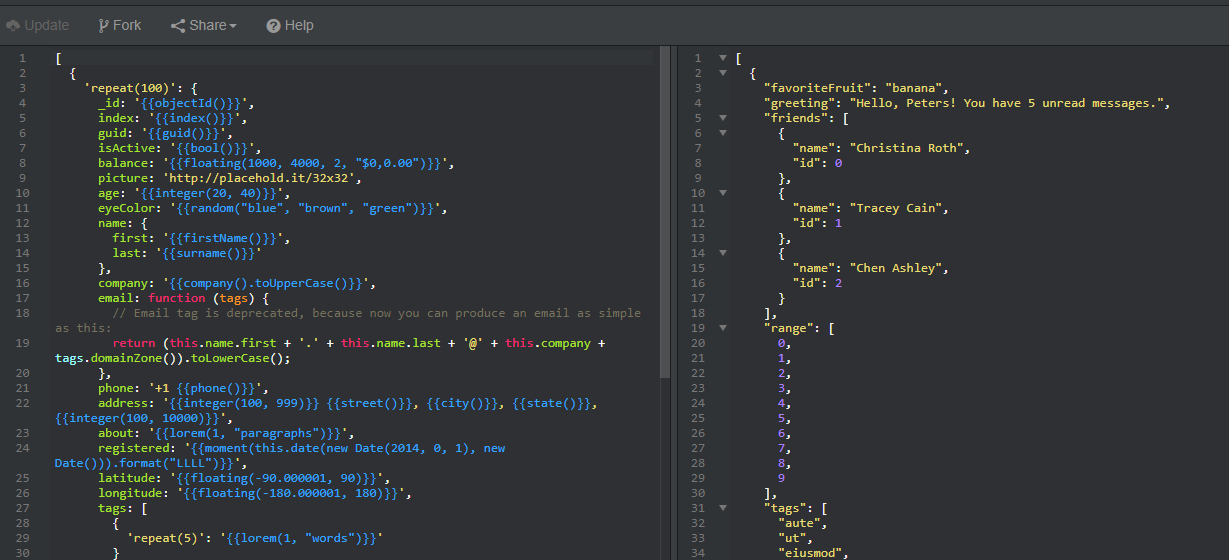
## יצירת הנתונים

בהתחלה התלבטנו באיזו דרך לייצר את הנתונים.

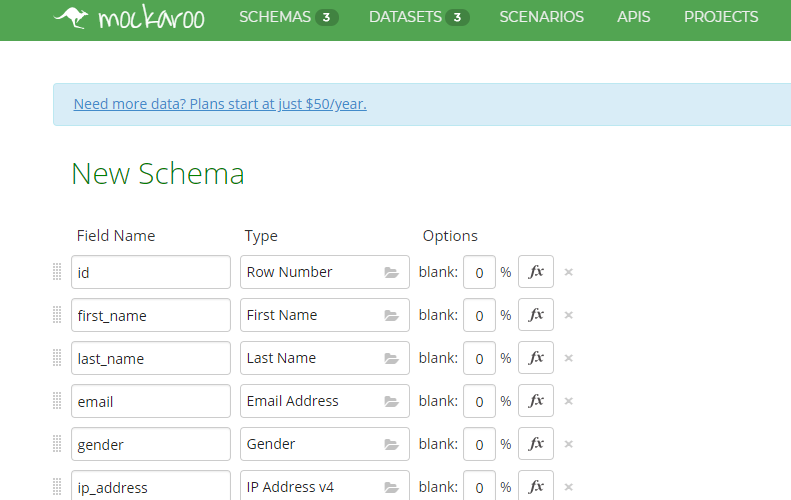
ניסינו כל מיני אתרים, כמו למשל האתר הזה:



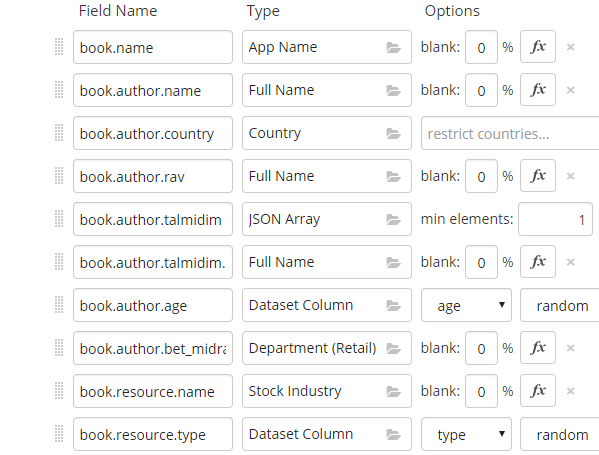
, אבל ראינו שצורת יצירת הנתונים לא כל כך מתאימה, גם אם הם נוצרים בפורמט המתאים. ניסינו גם אתר ייעודי לקבצי JSON:



לצערנו, גם זה לא הצליח כל כך הצליח. ניסינו גם באתר mockaroo:



גם שם נתקלנו בקשיים, כי לא ידענו איך לייצר את הנתונים בצורה שתתאים להיררכיה שהגדרנו, ושקלנו פשוט לייצר את הנתונים בצורה מתאימה לSQL, ואחר כך לעשות להם התאמה לJSON. לאחר מאמצים, הבנו שצריך להגדיר לכל משתנה בעץ, את כל המסלול שלו מתחילת העץ, ואז הצלחנו להשתמש באתר mockaroo ולייצר נתונים. למשל:



מאחר שרצינו לייצר נתונים שידמו את הספרים, הגדרנו פרמטרים קצת דומים ליצירת הנתונים. לדוגמה, על מנת להגדיר את מידת הדיוק של ההוצאה, בחרנו לייצר נתונים שמגדירים מידות של חולצה...

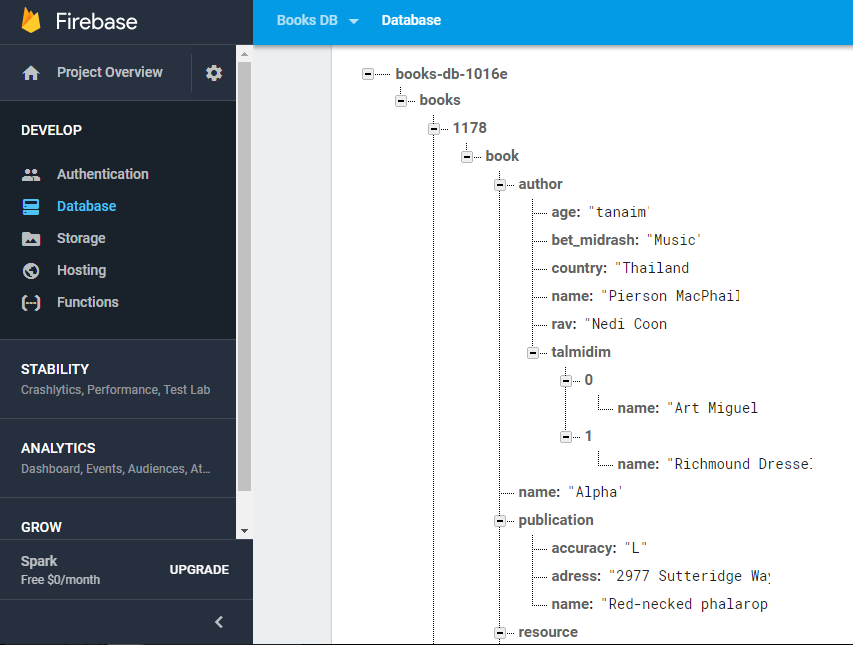


## בסופו של התהליך, קיבלנו קובץ JSON שמכיל את כל הנתונים, אבל לא בעיצוב הרגיל:

פתחנו את הקובץ בעזרת visual studio, ושם עשינו לו עיצוב לפי הפורמט המקובל בJSON:

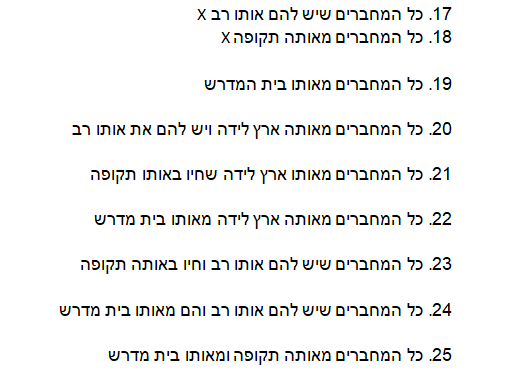


לאחר שסיימנו לייצר את הנתונים, העלינו אותם לfirebase:



## הגדרת השאילתות

יצרנו קובץ ובו הגדרנו לעצמנו את החיפושים שבהם נהיה מעוניינים:

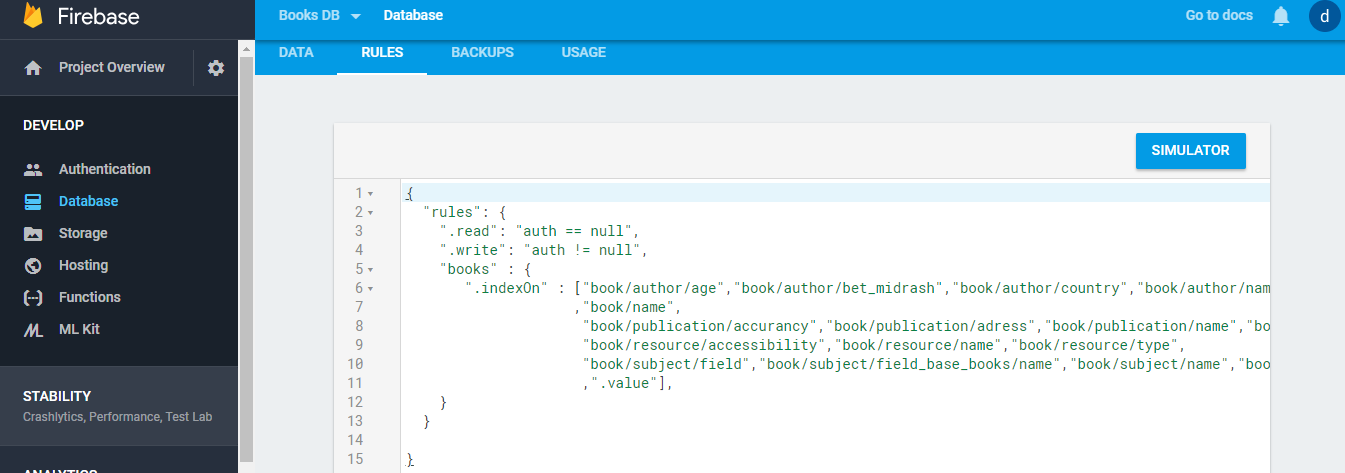


## כתיבת השאילתות

לאחר שהגדרנו לעצמנו באילו חיפושים אנחנו מעוניינים, ניגשנו לשלב של כתיבת הקוד של השאילתות. ראשית, היינו צריכים קוד שיבצע עבורנו את הגישה לfirebase. הורדנו את המודול firebase\_admin שמאפשר גישה לנתונים, ויצרנו קוד בסיסי שיבצע עבורנו את הגישה:



לאחר מכן, ניסינו להוציא את כל הנתונים. אחרי כמה נסיונות וחיפושים, הצלחנו למצוא דרך להוציא את הנתונים. רצינו לעבור לשלב הבא, של חיפושים, אבל פה התחלנו ממש להסתבך, כי לא ידענו מה בדיוק קורה. לכן, יצרנו מאגר קטן יותר, שמכיל רק כמה ספרים בודדים, על מנת שנוכל לעקוב אחרי מה שקורה. חיפשנו הרבה איך צריך להגדיר לfirebase את הדברים שעליהם ניתן לבצע חיפוש. בסופו של דבר, הגענו לתוצאה הזאת:



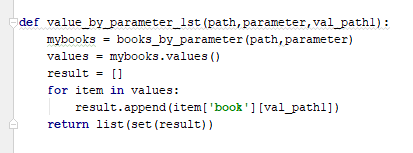
עכשיו יכולנו לגשת לכתיבת השאילתות. בהתחלה כתבנו שאילתות פשוטות, בסגנון:

ref.order\_by\_child('book/subject/field').equal\_to(XXX).get()

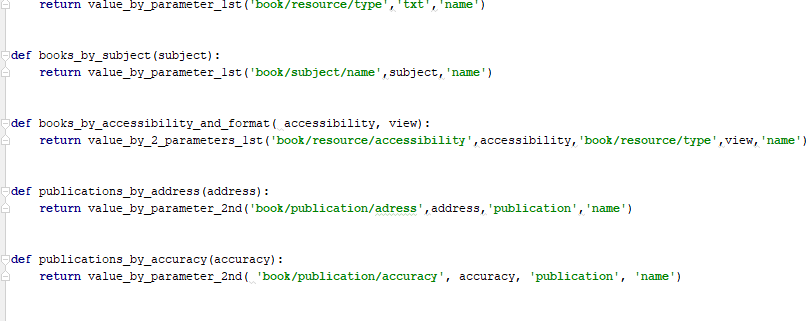
אחר כך יצרנו גם שאילתות מורכבות, לחיתוכים וכדומה. בשלב הבא, ניגשנו לבנות פונקציית חיפוש גנרית, שכל החיפושים משתמשים בה:



וקראנו לה בכל מימוש. אחר כך ראינו שהשאילתות האלה מחזירות לנו את כל הנתונים על הספרים הנבחרים, בזמן שאנחנו מעוניינים רק בנתון יחיד, למשל שמות הספרים. לכן יצרנו שאילתות קצת יותר מורכבות, שמאפשרות לדרוש נתון ספציפי יחיד לגבי הספרים שנמצאו בחיפוש:



וכל פונקציות החיפוש בפועל, פשוט קראו לפונקציה הגנרית המתאימה:



## יצירת הממשק למשתמש

עכשיו ניגשנו לשלב של יצירת הממשק למשתמש