# Python Full Stack Web Development

# Project – Part III



# כללי

**זהו החלק השלישי של הפרויקט המתגלגל.**

**בחלק זה עליכם להוסיף למערכת החופשות שבניתם, REST API שמנגיש סטטיסטיקות עבור המערכת ומאפשר לצפות בהן ע"י משתמש Admin בלבד. כמו כן על הפרויקט לרוץ ב-Cloud ע"י Docker.**

**טכנולוגיות פיתוח:**

* מסד הנתונים: MySQL.
* צד השרת: Django.
* צד הלקוח: Native או React.
* הרצה ע"י Docker
* העלאה ל-Cloud

**סוגי הפרויקטים:**

* צד השרת: פרויקט Django המשתמש ב-Django Rest Framework לבניית REST API.
* צד הלקוח:
  + אפשרות א': בניית צד הלקוח ע"י טכנולוגיות Native כגון HTML, CSS ו-JavaScript הקיימים בפרויקט ה-Django ומשתמשים בפניות AJAX ל-REST API לצורך קבלת והצגת המידע.

**שימו לב: עבור אופציה זו הציון המקסימלי של הפרויקט יכול להיות רק 90.**

* + אפשרות ב': בניית צד הלקוח ע"י פרויקט React נפרד, המשתמש ב-REST API שבניתם בפרויקט ה-Django לצורך קבלת והצגת המידע.

# מסד הנתונים

אותו מסד נתונים MySQL שבניתם בחלק השני של המערכת ללא כל שינוי.

# פרויקט ה-Backend

* יצירת פרויקט Django חדש.
* שימוש ב-Django Rest Framework לבניית REST API.
* התחברות ל-MySQL שבניתם באחת משתי דרכים לבחירתכם:
  + Code-First: בניית מודלים דרכם אפשר לקרוא את המידע ממסד הנתונים (מומלץ).
  + בניית DAL, Logic ו-Façade המאפשרים לקרוא את המידע ממסד הנתונים ע"י שאילתות SQL.
* על ה-REST API להכיל את ה-Routes הבאים:
  + Route לביצוע Login ע"י אימייל וסיסמה. רק משתמש המוגדר כאדמין יכול להיכנס למערכת.
  + Route לביצוע Logout מהמערכת.
  + Route להחזרת אובייקט המכיל שלושה מאפיינים: מספר החופשות שכבר הסתיימו, מספר החופשות הפעילות כרגע ומספר החופשות העתידיות.

דוגמה לאובייקט כזה:

{ "past\_vacations": 12, "on\_going\_vacations": 7, "future\_vacations": 15 }

* + Route להחזרת אובייקט המכיל את מספר המשתמשים הקיימים במערכת.

דוגמה לאובייקט כזה:

{ "total\_users": 37 }

* + Route להחזרת אובייקט המכיל את מספר הלייקים שיש כרגע במערכת.

דוגמה לאובייקט כזה:

{ "total\_likes": 84 }

* + Route להחזרת מערך המכיל אובייקטים של התפלגות הלייקים לפי יעדי החופשות.

דוגמה למערך כזה:

[ { "destination": "Rome", "likes": 3 }, { "destination": "Rhodes", "likes": 8 }, … ]

# פרויקט ה-Frontend

* פרויקט ה-Frontend יכיל את הדפים הבאים:
  + דף בית המציג הסבר על המערכת ותמונה הקשורה לסטטיסטיקות.
  + דף Login המאפשר להיכנס למערכת ע"י אימייל וסיסמה.
  + דף סטטיסטיקות המציג את כלל הסטטיסטיקות הנ"ל ע"י גלישה ל-Routes שבניתם, קבלת המידע והצגתו בצורה אסתטית וברורה. דף זה נגיש אך ורק למשתמשים שביצעו Login.
  + דף אודות המציג את הפרטים שלכם – מתכנתי המערכת.
* בנוסף, יש לבנות מנגנון Routing ותפריט המאפשר לנווט בין הדפים הנ"ל. על מנגנון ה-Routing להכיל גם Default Route וכן Route עבור Page-Not-Found.
* עיצוב אסתטי עבור האתר.

# Docker

* על כלל הפרויקטים לרוץ על Docker Containers.
* לכל פרויקט יש לבנות Dockerfile משלו.
* יש לבנות Docker Compose אחד עבור כלל המערכת.

# Cloud

* יש להעלות את כלל ה-Docker Images המוגדרים ב-Docker Compose ל-Docker Hub.
* יש להריץ את המערכת על AWS Cloud ע"י ביצוע Pull ל-Docker Images והרצתם ע"י ה-Docker Compose.

# הגשת הפרויקט

* יש להגיש תיקייה ראשית המכילה את התיקיות הבאות:
  + Database: מכילה את קובץ ה-Export של מסד הנתונים.
  + Flask: מכילה את פרויקט ה-Flask.
  + Django: מכילה את פרויקט ה-Django.
  + React: מכילה את פרויקט ה-React (אם קיים).
* יש לוודא שקבצי ה-requirements.txt של Flask ו-Django מכילים את כלל ההתקנות (ביצוע pip freeze).
* יש למחוק את תיקיות הסביבה הווירטואלית מפרויקט ה-Flask ומפרויקט ה-Django.
* יש לבצע קובץ zip אחד ויחיד מהתיקייה הראשית ולהעלות ללשונית הגשת הפרויקט באתר הקורס.
* מועד הגשה: היום האחרון של המסלול – 04/06/2024, בו תתקיים הגנה על הפרויקט.

# בהצלחה!