

Python Full Stack Web Developer Project Part III



זהו החלק השלישי של הפרויקט המתגלגל.

בחלק זה עליכם להוסיף למערכת החופשות שלכם פרויקט עם frontend ו-backend המנגיש סטטיסטיקות עבור המערכת ומאפשר לצפות בהן ע"י משתמש Admin בלבד.
כמו כן על הפרויקט לרוץ בסביבת ענן יחד עם שימוש ב-Docker.

טכנולוגיות פיתוח:

- מסד הנתונים: PostgreSQL.
- צד השרת: Django או Flask.
- צד הלקוח: React.
- הרצה בסביבת Docker.
- העלאה ל-AWS Cloud.

סוגי הפרויקטים:

- צד השרת: פרויקט Django או פרויקט Flask.
- צד הלקוח: פרויקט React המשתמש ב-API שבצד השרת לצורך קבלת והצגת המידע.

מסד הנתונים

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

יש לטעון נתונים התחלתיים אל מסד הנתונים בעת בניית הפרויקט על ידי Docker Compose.

פרויקט ה-Backend

- יצירת פרויקט חדש (פרויקט נפרד מפרויקט אתר החופשות, אך מתחבר לאותו מסד נתונים!).
- שימוש ב-Framework לבניית API (השתמשו ב-Flask או ב-Django).
- תקשורת עם מסד הנתונים באחת משתי דרכים לבחירתכם:
 - o Code-First: בניית מודלים דרכם אפשר לקרוא את המידע ממסד הנתונים.
 - o בניית DAL + BLL המאפשרים לקרוא את המידע ממסד הנתונים ע"י שאילתות SQL.
- על ה-API להכיל את ה-Routes הבאים:
 - o /login לביצוע Login ע"י שם משתמש וסיסמה. רק משתמש המוגדר כאדמין יכול להיכנס למערכת.
 - o /logout לביצוע Logout מהמערכת.
 - o /vacations/stats להחזרת אובייקט המכיל שלושה מאפיינים: מספר החופשות שכבר הסתיימו, מספר החופשות הפעילות כרגע ומספר החופשות העתידיות.
דוגמה לאובייקט כזה:
{ "pastVacations": 12, "ongoingVacations": 7, "futureVacations": 15 }
 - o /users/total להחזרת אובייקט המכיל את מספר המשתמשים הקיימים במערכת.
דוגמה לאובייקט כזה:
{ "totalUsers": 37 }
 - o /likes/total להחזרת אובייקט המכיל את מספר הלייקים שיש כרגע במערכת.
דוגמה לאובייקט כזה:
{ "totalLikes": 84 }
 - o /likes/distribution 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 כותרת, תמונה הקשורה לסטטיסטיקות, הסבר על המערכת ותפריט ניווט עם לינקים לשאר הדפים.
 - דף Login המאפשר להיכנס למערכת ע"י שם משתמש וסיסמה.
 - משתמש שלא מוגדר כאדמין ומנסה להתחבר יקבל הודעת שגיאה.
 - 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 מחוברים.
 - בונסים: הוספת סטטיסטיקות נוספות, הצגה של גרפים המבוססים על הסטטיסטיקות, הוספת widgets לסינון הנתונים המוצגים.
 - דף אודות המציג את הפרטים שלכם – מתכנתי המערכת.
- יש לבנות מנגנון Routing ותפריט המאפשר לנווט בין הדפים הנ"ל. על מנגנון ה-Routing להכיל גם Default Route וגם Route עבור page-not-found.
- יש ליצור עיצוב אסתטי עבור האתר.
- בונס: יצירת טסטים לבדיקה של דף הבית, דף הסטטיסטיקות, ודף ההתחברות.

Docker

- על כל הרכיבים לרוץ ב-Docker Containers (לכל אתר יש לבנות Dockerfile משלו).
- יש לבנות Docker Compose אחד עבור כלל המערכת (database, vacations, stats).

Cloud

- יש להעלות את 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

בדומה לחלקים הקודמים של הפרויקט:

- יש להשתמש ב-Repository ב-GitHub עבור הפרויקט ולתרגל עבודה עם Git.
- שימו לב: בשונה מהפרויקטים הקודמים, יש להגיש Repository המכילה מספר פרויקטים. להלן המבנה המומלץ:

```
projectRoot/
├── vacations_website/ ← הפרויקט של חלק 2
│   ├── Dockerfile
│   ├── initial_data.json
│   └── ...
├── stats_website/
│   ├── backend/ ← צד שרת עבור אתר הסטטיסטיקות
│   ├── frontend/ ← צד לקוח עבור אתר הסטטיסטיקות
│   ├── Dockerfile
│   └── ...
└── docker-compose.yml
```

- כמו מקודם, יש להגיש את הקישור ל-Repository באתר הקורס.
- כמו מקודם, על תיקיית הפרויקט (project root) להכיל קובץ README.md עם הפרטים הבאים:
 - שם מלא של התלמיד/ה.
 - הוראות ב-README להרצת כלל הפרויקט באמצעות docker compose.
 - אם יש לכם הערות נוספות, יש להוסיף אותן בקובץ זה.
- יש לוודא שקבצי ה-requirements.txt של Flask ו-Django מכילים את כלל ההתקנות (pip freeze).

הערות:

- יש להקפיד על כתיבת קוד נקי ומסודר: שדות מתחילים באות קטנה, מחלקות באות גדולה וכו'.
- יש להפריד מחלקות לקבצים נפרדים.
- יש להוסיף שורת תיעוד (docstring) לכל פונקציה ומחלקה ו-type hints.
- יש לוודא שכל הטסטים עוברים בהצלחה.
- אין להעלות ל-GitHub תיקיות dependencies בסגנון venv או node_modules.

- מועד הגשה: 01/09/2025.

בהצלחה!