

פרויקט מסכם מודול Django

מסמך הנחיות ודגשים לפרויקט

מפרט אפיון הפרויקט

מבוא ומטרות

בפרויקט זה נתנסה בבניית **rest API** בצד שרת. ידע ויכולת בניית **rest API** נחשב למשימה מבוקשת ביותר בעולם הפיתוח בהייטק ולכן יש ליחס חשיבות גדולה מאוד לפרויקט זה ומומלץ להשקיע בקריאה ובהבנה של הקוד, המודולים, הקבצים, והתיקיות המרכיבים פרויקט זה!

תיאור כללי:

פיתוח צד שרת לאפליקציה אינטרנטית הכוללת מערכת ניהול אתר המאפשרת לעורך בלוג לפרסם תוכן לעורך ולמחוק אותו. ולמשתמשים רשומים לכתוב תגובות.

דגשים מרכזיים:

- יש להשתמש במאגר המידע **postgressSQL** בפרויקט זה • יש להשתמש **Django-Rest-Framework** •
- יש להשתמש בספריות **django-filter**, **django-cors-headers**, **djangorestframework-simplejwt** או **python-dotenv** ו **python-decouple** לצורך בניית פרויקט זה.

אופן הגשת הפרויקט:

את הפרויקט יש להעלות ל **git** – ללא תיקיות ה **venv**–ולשלוח לינק אליו במשימה שתפתח בקמפוס. על הפרויקט לכלול קובץ **ReadMe.md** שמסביר על הפרויקט, תכולתו, הפונקציונאליות ודרך ההתממשקות עמו.

דרישות כלליות

- יש לשמור על קוד נקי ומסודר: לנקות הדפסות **print** וקטעי קוד שהפכו להערות • יש לבנות קוד שמספר סיפור ולתת לפונקציות ולמשתנים שמות משמעותיים • יש לחלק את הפרויקט למודולים לפי נושאים: לשים לב שלא יופיע בו קובץ ובו למעלה מ 200 שורות קוד • יש להקפיד על המוסכמות לכתובת קוד.

חשוב לזכור שפרויקט זה יהיה חלק מתיק העבודות שלכם וייצג אותך בכבוד מול מעסיקים פוטנציאליים ולכן יש לשמור על אסתטיקה של קוד ועיצוב.

אפיון פרויקט בלוג עם משתמשים, כתבות ותגובות

מבנה הנתונים (Models)

- משתמשים (**User**)
- לכל משתמש תהיה הרשאה רגילה או הרשאת מנהל
- משתמשים רשומים יכולים לכתוב תגובות

מנהלים יכולים ליצור, לערוך ולמחוק כתבות.

כתבות (Article)

- כל כתבה כוללת כותרת, תוכן, תאריך פרסום, שם המחבר
לכל כתבה יכולה להיות רשימת תגיות (לסינון כתבות לפי קטגוריה)
הכתבות זמינות לכולם לצפייה.

תגובות (Comment)

- לכל כתבה יכולים להיות תגובות שנכתבו על ידי משתמשים רשומים
- כל תגובה כוללת תוכן, שם המשתמש, תאריך תגובה
המנהל יכול למחוק תגובות לא רצויות.

נקודות קצה (API Endpoints)

הרשמה והתחברות (Authentication)

תיאור	נתיב	שיטה
הרשמת משתמש חדש	/api/register/	POST
קבלת jwt token להתחברות	/api/token/	POST
חידוש ה Token	/api/token/refresh/	POST

ניהול כתבות (Articles API)

תיאור	נתיב	שיטה
קבלת רשימת כל הכתבות	/api/articles/	GET
חיפוש כתבות לפי כותרת, תוכן או תגיות	/api/articles/?search=<query>	GET
קבלת כתבה מסוימת	/api/articles/<id>/	GET
בלבד למנהל (חדשה כתבה יצירת)	/api/articles/	POST
עדכון כתבה (למנהל בלבד)	/api/articles/<id>/	PUT
מחיקת כתבה למנהל בלבד()	/api/articles/<id>/	DELETE

ניהול תגובות (Comments API)

תיאור	נתיב	שיטה
קבלת כל התגובות לכתבה מסוימת	/api/articles/<id>/comments/	GET
הוספת תגובה לכתבה (למשתמשים רשומים בלבד)	/api/articles/<id>/comments/	POST
מחיקת תגובה (למנהל בלבד)	/api/comments/<id>/	DELETE

בקרת גישה

משתמשים רגילים:

יכולים להירשם ולהתחבר

יכולים לצפות בכתבות

יכולים להוסיף תגובות

מנהלים - יכולים ליצור, לערוך ולמחוק כתבות

. יכולים למחוק תגובות של משתמשים

חיפוש

: חיפוש כתבות לפי

כותרת

תוכן

תגיות

שם מחבר

זריעה ראשונית של מסד הנתונים

יש ליישם זריעה ראשונית של מסד הנתונים: יש להזין לפחות 2 משתמשים, 2 כתבות, 2 תגובות לכל כתבה.

הפרויקט יכול תיעוד מלא לכל נקודות הקצה בAPI

תוך שימוש בBrowsable API של Django-Rest-Framework

שמירה של סודות במשתני סביבה:

יכש לשמור סיסמאות וכתובות כמשתני סביבה מבלי לחשוף אותן.

לשם כך השתמשו בספריה מתאימה לבחירתכם מבין הספריות המוצעות במבוא לפרוייקט.

קבוצות:

קבוצת עורכים, קבוצת משתמשים, קבוצת ניהול

צד לקוח:

יש לממש צד לקוח בראקט:

הפרויקט יכול סרגל ניווט עם אפשרויות הרשמה/התחברות/התנתקות

יש להציג בעמוד הראשי 3 כתבות אחרונות עם אפשרות להציג כתבות ישנות יותר וחיפוש

בלחיצה על כתבה יש להציג עמוד פרטי כתבה ותגובות. כולל אפשרות עריכה של תגובה ומחיקה של תגובה למשתמש שיצר אותה בלבד.