**פרוייקט אפליקציית מזג אויר**

**מטרת הפרוייקט:**

**שימוש נכון במבנה mvc**

**שימוש בפונקציות middleware**

**שימוש בjwt**

**שימוש בmailer**

**שימוש בrequest**

**שימוש נכון במבנה סכמות ובקשרים בינהן**

**תרגול פונקציות mongoose**

**דרישות:**

**עליכן לפתח אפליקציית צד שרת כאשר כל הקריאות ייבדקו עם postman ויתועדו בpostman collection.**

**האפליקצייה תכלול רישום וכניסה של users למערכת על בסיס שם משתמש וסיסמה.**

**כל משתמש שנוסף למערכת ייקבל מייל ״ברוכים הבאים״ מהכתובת noreply.**

**בסוף האוטנטיקצייה, ייתקבל jwt שישלח עם כל קריאת api.**

**כל user יוכל לבצע קריאה (request) לapi הבא:**

<https://openweathermap.org/api>

**על מנת למצוא את נתוני מזג האויר בעיר אותה הוא מבקש.**

**כל היסטוריית הקריאות של כל user נשמרת בבסיס הנתונים שלו וניתן להגיע אליה על ידי שימוש בid של אותו משתמש.**

**משתמש יכול למחוק קריאות מההיסטוריה.**

**שמ**

**בנוסף,**

**קיים admin שמנהל את כל הusers. לאדמין יש גישה לכל הusers ובאפשרותו למחוק אותם.**

**במחיקת user יש למחוק גם את כל נתוני מזג האויר שלו.**

**דגשים:**

**יש ליצור סכמה של user, סכמה של admin וסכמה של weather.**

**הסכמות אמורות להיות קשורות בצורה נכונה בקשרים הרלוונטים (יחיד ליחיד, יחיד לרבים או רבים לרבים).**

**יש להשתמש ב pre/post בתוך הסכמה.**

**יש להשתמש בפונקציות mongoose המיועדות לקשרים בין סכמות (populate, push, set, pullלמחיקה וכו)**

**יש להחזיר סטטוס קוד נכון ותקין לכל שגיאה או הצלחה. מספיק לאחד**

**בהצלחה!**

הפונקציות:

USER

* רישום (הוספת משתמש) + שליחת מייל, token, ושלא יהיה במידלוור של בדיקת תוקן
* לוגין
* קבלת כל החיפושים לפי userId

WEATHER

* הוספת חיפוש
* מחיקת חיפוש מההסטוריה. (כמובן למחוק גם מה user)

ADMIN

* רישום (הוספת מנהל)+ שליחת מייל, token, middleware-pre,post
* לוגין אצל User
* מחיקת משתמש (כמובן גם מה admin)
* קבלת כל המשתמשים של מנהל מסוים.

בהצלחה רבה!

Populate for the get funcs