

# תרגיל 4 להגשה – Node.js

בל הזכויות שמורות לקרן יעקב 2024

# הגשת תרגיל 4 – זמנים ודגשים

- תחילת התרגיל – 24.6.24
- תאריך אחרון להגשה – 10.7.24 – 00:01
- בתוך תיקיית dev\_XX (שכבר פתחתם) תהיה תיקיה בשם ex4 ושם יהיו הקבצים של הפרוייקט
- במודל יש לפרסם קישור לריפוזיטורי בגיטהאב ולנתיב הראשי של השרת (ב-Render)

## התרגיל לא יבדק במקרים הבאים:

- אין לינקים במודל
- הגשה באיחור ללא אישור
- חשד להעתקה

## הוראות תרגיל 4 (1)

התרגיל הזה יהיה קצת שונה, תצטרכו להבין מה לתכנת מתוך הדרישות בפועל.

בתרגיל הזה לא יהיה צד לקוח מתוכנת, נשתמש בפוסטמן.

השירות צריך להיות מקושר לדטבייס כדי לשמור, לערוך ולקבל את ההעדפות ולתת את התוצאה של החופשה המבוקשת.

בשדרושות רשימות לבחור מהן נביא מקובץ JSON במקום לפתוח טבלה חדשה בדטבייס.

## הוראות תרגיל 4 (2)

אתם רוצים לטוס עם עוד 4 חברים (סה"כ תהיו 5) ליעד כלשהו לחופשת הסמסטר. פתחתם קבוצת ווטסאפ אבל קשה לעקוב אחרי הבקשות והאילוצים של כולם ולכן החלטתם לתכנת שירות Node.js שינהל ויחשב את היעד והתאריכים לחופשה.

כל אחד מהחברים יצטרך להכניס את שם משתמש וסיסמה לפני השימוש במערכת החופשות ולקבל קוד גישה. אפשר להירשם עם אותו שם המשתמש רק פעם אחת וגם קוד הגישה יהיה ייחודי. כדי להכניס העדפות חדשות לחופשה של משתמש או לערוך אותן חייבים להזין את קוד הגישה הנכון. אי אפשר למחוק העדפות חופשה, רק לערוך אותן. כל אחד יוכל לקבל את ההעדפות של כל אחד מהחברים, כולל את שלו (כלומר לא צריך קוד גישה)

כל אחד מהחברים יוכל להכניס את הפרטים המבוקשים לחופשה: תאריך התחלה, תאריך סיום, יעד מבוקש (מתוך רשימה של 10), סוג חופשה (מתוך רשימה של 5: מנוחה/ים/פארקים וכו'). אורך החופשה לא יוכל להיות יותר משבוע (בין תאריך ההתחלה ותאריך הסיום)

## הוראות תרגיל 4 (3)

כל אחד יוכל לחשב את תוצאות החופשה המבוקשת, רק אם כל ה-5 חברים מילאו את ההעדפות. אם לא – תהיה תשובה שצריך לחכות לכל ההעדפות של כולם.

תוצאות החופשה המבוקשת יחושבו לפי:

יעד – לפי הרוב

סוג חופשה – לפי הרוב

תאריכים – חפיפה בין תאריכים שכולם הכניסו

לדוגמה: 4 חברים העדיפו בין 1.8 ל-6.8 וחבר אחד העדיף בין 3.8 ל-5.8 – יבחרו התאריכים בין 3.8 ל-5.8

אם אין רוב ואין החלטה ליעד או לסוג חופשה – תיבחר האפשרות שנבחרה ראשונה (הכי מוקדמת) ע"י אחד החברים

# חלוקת הניקוד

- 40 נק' - פונקציונליות תקינה ועובדת, כולל מקרי קצה וניהול שגיאות
  - 20 נק' - קוד מסודר ונקי, clean code, שמות משתנים
  - 15 נק' - השירות מומש ע"י עקרונות RESTful API עם ראוטים ומתודות נכונות
  - 10 נק' - קומיטים משמעותיים במהלך העבודה (לפחות 15) מהתחלת הפרוייקט
  - 15 נק' - יש לייצר API - איך מבצעים קריאות לשירות ואיזו תשובות מתקבלות
- בנוסף 5 נק' - שימוש ב-API חיצוני שמציג את מזג האוויר ביעד הנבחר (לאחר שמתקבלת התשובה)
- התרגיל כולו הוא 10% מהציון הסופי בקורס.

# הנחיות נוספות

---

מצגות נוספות שיועלו בהמשך:

- להעלות את קוד צד השרת ל-Render
- פיבלוש API ע"י פוסטמן