**משימה לתרגול והתנסות - פרויקט מסכם למודול**

**Node JS**

**להלן פרטי המשימה**

**מטרת הפרויקט**

בפרויקט זה נתנסה בבניית rest api בצד שרת.

תשתית זו תשמש אותנו בהמשך גם בפרויקט המסכם את מודול - react.

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!

**כלים**

פרויקט זה יהיה מבוסס על node js וכן הוא ישלב את המודולים הבאים:

* express
* mongoose
* bcryptjs
* joi
* jsonwebtoken

\* ניתן להשתמש במודולים ובספריות נוספות במידת הצורך

**בסיס נתונים**

יש להשתמש בפרויקט זה בבסיס הנתונים של MongoDB בלבד כדי להתנסות באופן שמירת נתונים לפרויקט שצפוי להתמודד עם Big Data .

חשוב מאוד לקבל נסיון פרקטי באחד מתוכנות לבסיסי הנתונים הפופולריים השייכים למשפחת ה NoSQL

**להלן דרישות הפתוח בפרויקט**

מקרא: נקודת גישה = endpoint שיטת = method

1. צרו נקודת גישה בשיטת POST שתאפשר הרשמה של לקוח חדש למערכת.

הלקוח יכול להיות לקוח רגיל או לקוח עסקי.

בתהליך ההרשמה יספק הלקוח שם, אימייל (ייחודי), סיסמה, ודגל המסמן האם הוא לקוח עסקי או רגיל.

במהלך ביצוע בקשה לנקודת גישה זו, יש לבצע ולידציה על הנתונים שסופקו במידה והנתונים נמצאו לא תקינים יש להחזיר תשובה המסמנת סטטוס 400 עם מלל המפרט את השגיאה הרלונטית.

במידה והנתונים תקינים יש לשמור את פרטי הלקוח החדש בבסיס הנתונים ולהחזיר תשובה המסמנת סטטוס 200 לצד json המייצג את ה id השם והאימיל של הלקוח החדש שנוצר בבסיס הנתונים.  
 דגש חשוב: יש להצפין את הסיסמה של הלקוח בבסיס הנתונים בעזרת המודול bcryptjs.

1. צרו נקודת גישה בשיטת POST שתאפשר התחברות של לקוח למערכת דרך שליחת אימייל וסיסמה.

בנקודת גישה זו יש לבצע בשלב ראשון ולידציה מקדימה המוודאת תבנית תקינה לאימייל וכמו כן שהסיסמה באורך של שישה תווים לפחות.

בשלב שני יש לבצע בדיקה נוספת שהאימייל והסיסמה אכן קיימים בבסיס הנתונים. במידה ולא יש להחזיר תגובה המסמנת סטטוס 400 לצד מלל המתאר את השגיאה הרלונטית.  
 במידה ושני שלבי הבדיקה עברו תקין יש להחזיר תגובה המסמנת סטטוס 200 זאת לצד token (שיווצר בעזרת המודול jsonwebtoken) שיכיל את פרטי המשתמש. ה- id שלו כפי שקיים בבסיס הנתונים וכמו כן את הערך המסמל האם הוא לקוח רגיל או עסקי.

1. צרו נקודת גישה בשיטת GET שתאפשר קבלת פרטים מלאים של המשתמש.

הפרטים כדלהלן:

שם,אימייל, האם לקוח עסקי, תאריך יצירה, מזהה רשומה.

לנקודה זו יסופק טוקן, אותו token שהתקבל בעת התחברות מוצלחת (מסעיף 2).

במידה ונמצא תקין, יש להחזיר תגובה המסמנת סטטוס 200, זאת לצד הפרטים האמורים של המשתמש.

במידה ולא סופק token או שנמצא לא תקין יש להחזיר תגובה המסמנת סטטוס 401 לצד מלל המתאר את השגיאה הרלונטית.

1. נקודת גישה בשיטת GET שתאפשר קבלת מערך של כל כרטיסי הביקור השמורים תחת המשתמש הנוכחי.  
    נקודת גישה זו תהייה פתוחה למחוברים בלבד ותדרוש את ה token שסופק לאחר התחברות (מסעיף 2)
2. צרו נקודת גישה בשיטת GET שתקבל id של כרטיס ביקור ותחזיר את הפרטים המלאים של אותו כרטיס ביקור במידה וכרטיס הביקור שייך למשתמש הנוכחי.

נקודת גישה זו תהייה פתוחה למשתמשים מחוברים בלבד ותדרוש את ה token שסופק לאחר תהליך התחברות (סעיף 2).

1. נקודת גישה בשיטת DELETE שתקבל id של כרטיס ביקור ותאפשר מחיקת כרטיס ביקור.  
    נקודת גישה זו תהייה פתוחה למחוברים בלבד ותדרוש את ה token שסופק לאחר תהליך התחברות (סעיף 2)

רק הלקוח ששייכת לו הכרטיסיה יוכל למחוק את הכרטיסיה

1. נקודת גישה בשיטת PUT שתקבל id של כרטיס ביקור ותאפשר עריכת פרטי כרטיס ביקור.  
    נקודת גישה זו תהייה פתוחה למחוברים בלבד ותדרוש את ה token שסופק לאחר תהליך ההתחברות (סעיף 2)

רק הלקוח ששייכת לו הכרטיסיה יוכל לערוך את הכרטיסיה

1. צרו נקודת גישה בשיטת POST שתאפשר פתיחת כרטיס ביקור למשתמשים שהוגדרו עסקיים.

על הכרטיס שיפתח לכלול את השדות הבאים:

שם העסק, תיאור העסק, כתובת העסק,טלפון העסק, תמונת העסק (קישור לתמונה) כמו כן בעת שמירת כרטיס הביקור יונפק לכרטיס מזהה רנדומלי וייחודי.

דגש חשוב: גישה זו תהיה פתוחה רק במקרה שצורפה לבקשה token אותו טוקן שמקבלים לאחר לאחר התחברות (כמופיע בסעיף 2).

יש לשמור את הכרטיס שהתקבל בבסיס הנתונים.

במידה והשמירה הצליחה יש להחזיר תגובה המסמנת סטטוס 200 זאת לצד פרטי כרטיס הביקור החדש שנשמר בבסיס הנתונים.  
במידה והשמירה לא הצליחה יש להחזיר תגובה המסמלת כישלון.

דגש חשוב: יש לשמור בכרטיס הביקור מזהה משתמש (user\_id) שמכוון לרשומה רלוונטית באוסף משתמשים, זאת כדי שנוכל לזהות איזה כרטיס שייך לאיזה משתמש.