

### סביבות פיתוח באינטרנט – עבודת בית 3

#### מטרת התרגיל:

בתרגיל זה תתכננו ותממשו צד שרת – server side בסביבת node.js, תוך שימוש בבסיס נתונים ב-Azure. חשבו על עצמכם כמתכנתים ומנהלי מוצר אשר מקבלים מפרט מלקוח ונדרשים לתכנן לממש את האתר.

#### יש לקרוא את כל ההוראות לפני תחילת העבודה!

#### כללי:

תרגיל זה מתמקד בצד שרת ומתחלק ל 2 חלקים:

1. הצעת API
2. מימוש API – יפורסם בהמשך

בתרגיל הנוכחי תבצעו כתיבה של הצעת API ואילו בתרגיל הבא תממשו אותו, לכן שימו לב שאתם כותבים כמה שיותר מסודר ומקיף על מנת לייעל את עבודתכם בהמשך. את בסיס הנתונים יש ליצור ב-Azure כפי שהוסבר בתרגול, מצורף script כהצעה ליצירת טבלה

#### המערכת אותה עליכם לבנות:

נושא הפרויקט הינו מערכת מידע על אתרי תיירות ועניין שונים בערים בעולם (כדוגמת Trip Advisor), נכנה זאת בשם: נקודות עניין (Point Of Interests), אשר תאפשר למשתמשים לחפש אחר מקומות פופולריים לביקור בעיר מסוימת וכן לשמור פריטי מידע אלה במרכז לצורך שימוש במהלך הטיול שלהם (חשבו על: שמירה למועדפים)

נק' עניין יכולות להיות: אתרים היסטוריים, אטרקציות, מוזיאונים, מסעדות, מועדוני לילה, מרכזי קניות גדולים וכד'.

#### עליכם לבחור עיר תיירות מרכזית בעולם ולממש את הפרויקט בהתאם לעיר אותה בחרתם.

המערכת תשמש ל:

- הצגת כלל אתרי התיירות והעניין בעיר מסוימת לפי קטגוריות
- שמירה והצגה מאוחרת של אתרים שנבחרו
- תרומת משתמשים לאתר: דירוג נקודות עניין והשארת ביקורת
- ניהול (מינימלי) של מאגר המשתמשים ופריטי המידע

בחלק הבא יפורט המפרט המבוקש עבור האתר, מהפירוט עליכם לגזור את ה API המבוקש ואופן המימוש של השרת האפליקציה עבור האתר המבוקש. שימו לב כי בנוסף למפרט, מפורטות לכם הדרישות של פריטי המידע הנדרשים בכל חלק.

## 1. מנשק משתמשים

### 1.1. כללי

- עליכם לעצב את האתר: צבעים, תמונות, אלמנטים וכד' על פי העיר עליה בחרתם לממש את הפרויקט.
- עליכם לבחור 4 קטגוריות עם 5 נק' עניין לפחות בכל קטגוריה
- את האפליקציה ילווה bar – סרגל ניווט אשר יאפשר לעבור בין הדפים השונים
- לאורך כל האפליקציה, נקודות עניין תופענה עם שם ותמונה מתאימה, לחיצה על השם או התמונה תפתח חלון חדש (new tab) של נקודת העניין, עם הפרטים הבאים:
  - כמה צפו בנקודת העניין
  - תיאור של נק' העניין
  - דירוג באחוזים
  - 2 ביקורות אחרונות עם תאריך

### 1.2. מסך כניסה

#### 1.2.1. לקוחות חדשים (או שלא רשומים או שלא מחוברים)

- בחלק העליון של המסך תופיע הודעת המבקשת מהמשתמש להתחבר עם מקום לעשות זאת, ובמידה ואינו מחובר מעודדת אותו, בהודעה מתאימה להירשם לאתר, עם כפתור אשר יעביר את המשתמשים לעמוד הרשמה - Register.
- המערכת תאפשר אחזור סיסמא ע"י הצגתה על גבי המסך ללקוח החפץ בכך, אחזור הססמא יהיה באמצעות מענה על שאלת אימות. ( למשל: מה כתובתך? יש לשאול על שאלות אלו בעת ההרשמה)
- חלקו התחתון של המסך יהיה אזור explore, אשר יציג 3 נקודות עניין פופולריות (מעל דירוג מסוים) ורנדומליות, כלומר בכל פעם שנטען מחדש את העמוד הנק' יתחלפו.
- בחלקו העליון של המסך יירשם hello guest

#### 1.2.2. לקוחות רשומים

- לקוחות מחוברים יזוהו בהרשמה באמצעות token אשר יישמר בדפדפן בזמן ההרשמה שלהם – כפי שיימדד בתרגולים.
- עבור לקוחות מחוברים, יש להראות שהלקוח זוהה ע"י hello [username] בחלק העליון של המסך , כמו כן לא יראה את אפשרות הניווט לחלון Register
- המערכת תציג למשתמש מסך המחולק לשני חלקים באופן ברור: נקודות עניין מומלצות וכן נקודות עניין אחרונות שנשמרו.
- המערכת תציג למשתמש 2 נק' עניין הפופולריות ביותר בשני תחומים שונים (נקודת עניין לתחום), על פי התחומים המועדפים אותם בחר המשתמש בהרשמה שלו (יש לדרוש זאת בהרשמה)
- בחלקו השני המערכת תציג למשתמש 2 נק' עניין אחרונות ששמר במערכת, במידה ולא שמר תוצג הודעה מתאימה

### 1.3. מסך הרשמה

- ברישום משתמש יש לבקש את הפרטים הבאים:
  - FirstName
  - LastName
  - City
  - Country
  - Email
  - קטגוריות קיימות שמעניינות את הלקוח, לפחות 2
  - שאלות אימות לשחזור ססמא
- ערכים אפשריים של ארץ הלקוח יש לקחת מXML - המצורף לתרגיל ולשלבם באתר.
- לכל לקוח יוגדר שם משתמש (בין 3 ל 8 תווים) וסיסמא (בין 5 ל 10 תווים)
- שם המשתמש יכול אך ורק אותיות והסיסמא תכיל שילוב של אותיות ומספרים בלבד. סלאחר הרשמה יש להעביר את הלקוח לדף ה כניסה על מנת שיתחבר למערכת.

### 1.4. מסך About

- יכלול תקציר שיווקי עם תמונות בבחרות על העיר אותה אתם משווקים בפרויקט.

## 2. ממשק נקודות עניין

- יש להציג את רשימת נקודות העניין לפי קטגוריות ולאפשר סינון ע"פ קטגוריה ספציפית
- יש לאפשר מיון של נק' העניין על פי דירוג

### 2.1. מסך תצוגה של נקודות עניין

#### 2.1.1. חיפוש נקודות עניין

- יש לאפשר חיפוש של נקודת עניין לפי שמה, תוצאות החיפוש יוצגו בממשק נקודות העניין.
- במידה ולא נמצאו תוצאות רלוונטיות יש להציג הודעה מתאימה
- גם כאן לחיצה על נקודת עניין שחזרה תאפשר פתיחת חלון חדש עם פרטים

#### 2.1.2. שמירת נקודות עניין

- אופציה זו תאפשר למשתמשים רשומים באתר בלבד, משתמשים שאינם רשומים לא יראו את האופציות הבאות:
- ליד כל נק' עניין יופי סימן מתחלף (לדוגמא כוכב של מועדף – ריק או צבוע)
- כאשר משתמש ירצה לשמור נק' עניין מועדפת, הוא ילחץ על הסימון, צורת הסימון תתחלף כך שתעיד שנק' העניין נשמרה, וכן הנק' תיוסוף לרשימת מועדפים
- לחיצה חוזרת תסיר את הנק' מרשימת המועדפים – סימון כמועדף יתאפשר גם בעמוד הנקודה עצמה וגם כאשר מופיע ברשימות
- בראש המסך יופיע למשתמשים הרשומים אייקון למעין "רשימת מועדפים", בחרו סימון מתאים לבחירתכם, בכל פעם שתווסף נק' עניין שנשמרה יופיע מונה המציין את מספר הפריטים ברשימת המועדפים.
- לחיצה על האייקון תעביר את המשתמשים למסך המועדפים

## 2.2. מסך נקודות עניין מועדפות

- במסך זה יצפה המשתמש בנקודות העניין אותן שמר
- יתאפשר מיון של נק' העניין לפי קטגוריה או דירוג
- למשתמש יתאפשר להסיר נקודות מהרשימה
- גם כאן יוכל המשתמש ללחוץ על נק' עניין ולקבל את עמוד הפרטים עליה
- למשתמש יתאפשר למיין בעצמו את רשימת נקודות העניין, כך שתופענה לו בסדר בו הוא מעדיף לצורך שימוש עתידי.

### 2.2.1. שמירה של רשימת המועדפים

- יש לשמור את נקודות העניין של המשתמש בסשן בזיכרון המקומי בצד לקוח
- המשתמש יוכל לשמור את נק' העניין בבסיס הנתונים בשרת כך שתתאפשר לו גישה אליו מכל מחשב בכל עת, עליו לעשות זאת בלחיצה על כפתור מתאים. (לדוגמא: "שמור נק' מועדפות")

### 2.2.2. מתן ביקורת על נק' עניין

- לצד כל נק' עניין יהיה כפתור יתאפשר למשתמשים לתת ביקורת לנק' העניין
- בלחיצה על הכפתור ייפתח חלון modal אשר יאפשר למשתמשים:
- מתן דירוג בין 1-5
- כתיבת ביקורת
- במתן דירוג המערכת תשקלל את הציון של נק' העניין באחוזים, על פי כלל הדירוגים שזו קיבלה במערכת
- במידה וניתנה ביקורת מילולית, יש לעדכן אותה בסיס הנתונים (שימו לב שביקורת מעודכנת זו צריכה להופיע שעמוד המידע כפי שהוסבר)

### הוראות לתרגיל 3.1:

לעבודה זו 3 חלקים, שני החלקים הנותרים מימוש צד שרת ומימוש צד לקוח יפורסמו בהמשך:

**תאריך הגשה 1.5.18 –** יבוצע בזוגות: כל זוג יתחלק ל2 בעלי תפקידים: מתכנת ולקוח. תפקיד הלקוח הוא להגדיר את דרישות של המערכת ותפקיד המתכנת הוא לאפיין ולתרגם את הדרישות ל API שמוסכם על 2 הצדדים

#### מה מגישים:

1. יש להגיש מסמך ובו דף API בו אתם מפרטים את כל השיטות שה server שלכם יטפל בהן. הפירוט יעשה בטבלה הבאה (ראו דוגמא).. ניתן לכתוב את ה JSON או לכתוב מבנה נתונים כללי (למשל בדוגמא רשמנו החזרת מערך PRODUCTS , אך אפשר גם לרשום ממש את מבנה ה JSON).

אנו ממליצים בחום להתחיל לחשוב כבר עכשיו כיצד ייראה מבנה ה JSON שלכם על מנת שתוכלו לחסוך זמן יקר בהמשך כאשר התקופה תהיה יותר לחוצה):

| <u>ID</u> | <u>Method Name</u> | <u>HTTP Method</u> | <u>Parameters</u>  | <u>Returns</u>                        | <u>Explanation</u>  |
|-----------|--------------------|--------------------|--------------------|---------------------------------------|---|
| 1         | Login              | POST               | Username, Password | Token                                 | Login cannot pass in GET<br><br>True= success<br><br>False=fail   |
| 2         | getCities          | GET                | -                  | Returns all cities :<br><br>cities [] | Using GET because there is no information to deliver to the server or any wish to change any information. |
| ..        | ..                 | PUT                | ...                | ...                                   | ..  |

יש לבחור METHOD לפי הצורך, גם מעבר ל GET או POST ולנמק. שימו לב שה API תלוי מימוש ולכן קיימת עמודת הסבר בו עליכם להסביר למה דרושה השיטה ולמה בחרתם להשתמש בשיטה הספציפית של פרוטוקול http.

דגשים לכתיבת הAPI :

- כפי שנלמד בכיתה, עליכם לתת שמות משמעותיים לשיטות על מנת לאפשר API קריא וקל לשימוש
- חשבו היטב, היכן אתם שומרים איזה מידע, מתי כדאי לשמור בזיכרון המקומי של הדפדפן ומתי כדאי בDB

2. בסוף המסמך יש לשים תמונה שלכם המראה אתכם "עובדים" במרץ על קביעת ה API עם התאריך, כאשר כל אחד מחזיק שלט "מתכנת" או "לקוח".

#### יש להגיש את העבודה בשני אופנים:

1. הגשה לftp תחת תיקייה EX3.1 עם מספרי תעודות הזהות של שני המגישים
2. הגשה מודפסת לתא המתרגלים