

מדריך תיירות מקומי

מגישה: נעה ישראלי - 316392356  
מנחה: ד"ר איאד סולימאן

תוכן עניינים:

[תוכן עניינים: 1](#_Toc85477954)  
[מבוא: 2](#_Toc85477954)  
[הצגת הבעיה והמימוש: 3](#_Toc85477954)  
[מבנה האתר: 4-7](#_Toc85477954)  
[ארכיטקטורה: 8-9](#_Toc85477954)  
[דוגמאות: 10-12](#_Toc85477954)  
[טכנולוגיות: 13](#_Toc85477954)  
[סיכום ומסקנות: 14](#_Toc85477954)  
[ביבליוגרפיה: 15](#_Toc85477954)  
[קישורים: 16](#_Toc85477954)

מבוא

בפרוייקט זה מפורט תהליך הבנייה והשימוש של אתר המשמש לצריכת ואספקת תיירות על ידי אנשים פרטיים בארץ ובעולם.  
כולם מכירים את הסיטואציה בה אנחנו מגיעים לעיר שמעולם לא היינו בה בארץ או בחו"ל ולא יודעים מה הכי כדאי לראות. הרבה פעמים, לא משנה כמה נתכונן וכמה מפות או מסלולים יהיו לנו, אין כמו עצה של אדם מקומי על מנת להפוך טיול לבלתי נשכח.   
האתר המפורט בפרויקט זה מספק ממשק אלגנטי ופשוט למשתמש על מנת שכל אדם יוכל למצוא בלחיצת כפתור ועל ידי מילוי פרטים בסיסיים בלבד את המדריך המקומי שמתאים בדיוק בשבילו.  
בנוסף, ניתן מהצד השני להירשם לאתר גם כמדריך ולספק שירותי תיירות בשלל נושאים מגוונים בעיר בה גרים.

הצגת הבעיה

האתר נותן מענה על הבעיה של תיירים שאינם מכירים את המקום אליו הגיעו. האתר מתווך בין בני אדם שרוצים לצרוך תיירות מסוגים שונים לבין תושבים מקומיים שרוצים ויש להם את הידע להדריך ברחבי העיר שלהם בנושאים שונים.

מימוש הפתרון

האתר בנוי בצורה כזאת שאפשר להירשם כשני סוגי משתמשים – מדריך או תייר.

**משתמש מסוג תייר** צריך להזין בטופס ההרשמה פרטים כגון שמו, שפה שהוא מדבר, סוג הקבוצה שמחפשת הדרכה (משפחה, זוג, שותפים לעסקים וכו') ודרכי תקשורת.   
ברגע שהמשתמש מתחבר לאתר מופיע לו בדף הבית את רשימת המדריכים המדורגים ביותר באתר לעיונו, הוא יכול להיכנס לפרופיל של המדריכים דרך דף הבית שלו או ללחוץ על מציאת התאמה ולהפעיל את האלגוריתם התאמה של האתר.

**משתמש מסוג מדריך** צריך להזין בטופס ההרשמה פרטים כגון שמו, שפות אותן הוא מדבר, דרכי תקשורת כמו אי-מייל או מספר טלפון, גיל, האם יש ברשותו רכב, תעודת יושר , סוגי תיירות בהם הוא מתמחה (לדוגמה – קולינריה, חיי לילה, מוזיאונים, ספורט ...) ועוד.  
לאחר שנרשם כמדריך, הוא בעצם מחכה שמשתמש מסוג תייר ייצוא איתו קשר. ברגע שזה קורה וההדרכה מתבצעת בהצלחה, התייר רשאי לדרג את המדריך על הטיול ולכל מדריך יש דירוג משלו שמלווה בביקורות לעיונם של תיירים באתר.

מבנה האתר

האתר בנוי משלושה חלקים – חלק משותף, חלק למשתמש מסוג מדריך וחלק למשתמש מסוג תייר.

החלק המשותף:  
כולל בתוכו את דפים הבאים –

1. **Sign-in:** דף הכניסה
2. **Sign-up:** דף ההרשמה שבנוי בצורה אינטרקטיבית כך שלאחר מילוי של פרטים בסיסיים דף ההרשמה יותאם בזמן אמת להרשמה של תייר או מדריך לפי בחירת המשתמש.
3. **Header & Footer:** כותרת האתר ותחתית האתר אחידות לשני סוגי המשתמשים
4. **Navigation-Bar:** מופיע לשני סוגי המשתמשים ומותאם אישית.
5. **Edit-Profile:** דף לעריכת הפרטים – מותאם אישית לסוג המשתמש

החלק של המדריך:   
כולל בתוכו את הדפים הבאים –

1. **Guide Home Page:** דף בית בו כתוב לכל מדריך את הדירוג הנוכחי שלו, ביקורות אחרונות שנתנו לו, עריכת פרופיל וכו'.
2. **Reviews:** דף בו המדריך יכול לעיין ולהגיב לביקורות.

החלק של התייר:  
כולל בתוכו את הדפים הבאים –

1. **Tourist Home Page:** דף בית בו מוצגים המדריכים המדורגים ביותר, פרטי התייר, אופציה לדיווח על הדרכה שנגמרה וכפתור למציאת התאמה מידית למדריך קיים במערכת.
2. **Matches:** דף ההתאמות, ברגע שתייר לוחץ על כפתור ההתאמה מוצגת לו טבלה של המדריכים המתאימים לו לפי דירוג והוא יכול לבקר בפרופיל שלהם ע"י לחיצה על שמם בטבלה.

* התקשורת בין כל חלקי האפליקציה מתאפשרת בעזרת שירות Services של Angular שבעצם משתמש לתקשורת בין כל הדפים שממומשים ע"י קומפוננטות של אנגולר.

ה-Services:

1. **TouristService:**
   * אחראי על הבאת המידע של התייר המחובר כעת.
   * אחראי על הבאת רשימת המדריכים לכל תייר מה-DB.
   * אחראי על אלגוריתם ההתאמה
   * אחראי על בניית טבלת תוצאות ההתאמה.
   * מחזיק את המחלקה שמייצגת תייר.
2. **GuideService:**

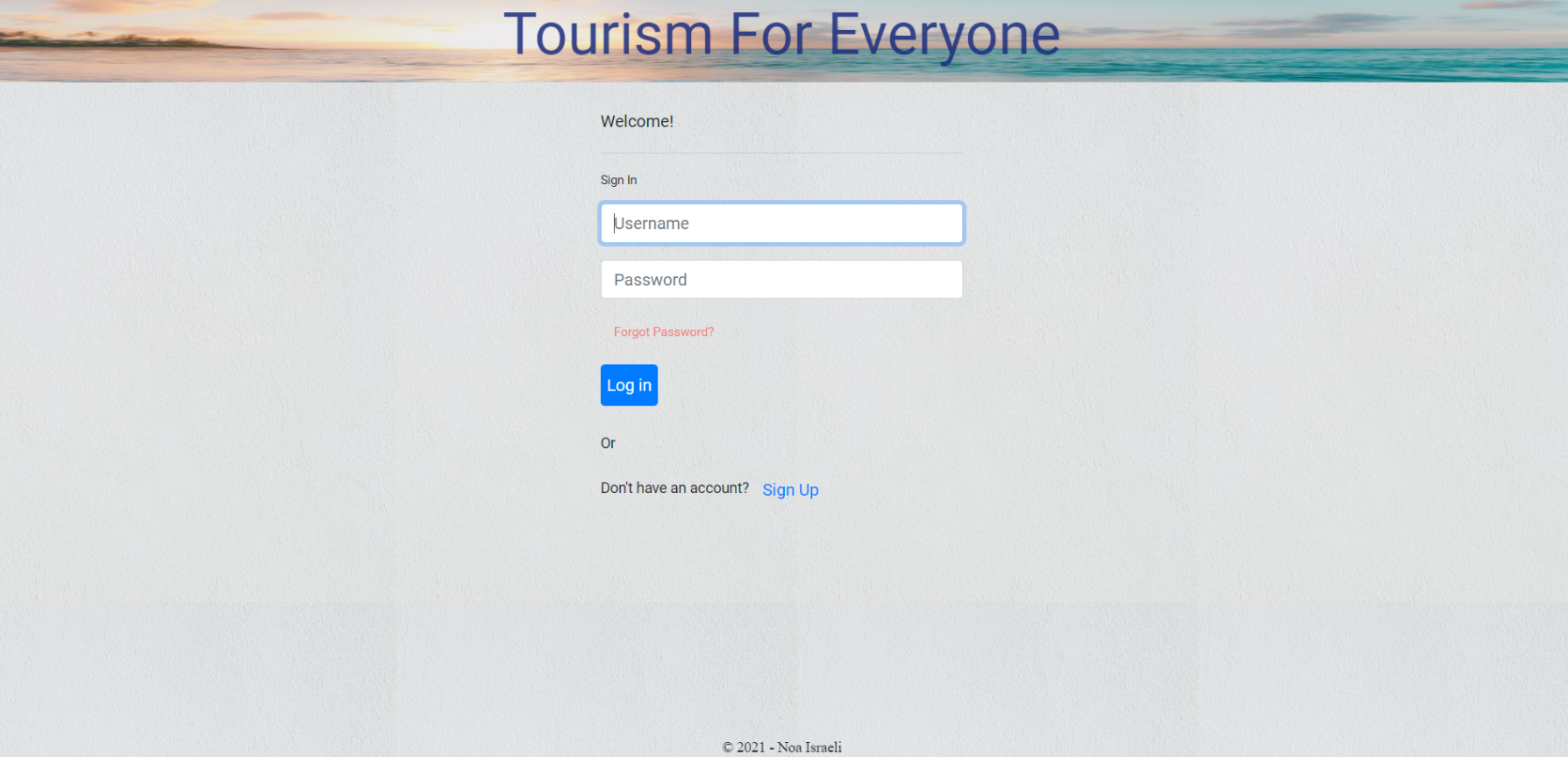
* אחראי על הבאת המידע של המדריך המחובר כעת.
* מחזיק את המחלקה שמייצגת מדריך.

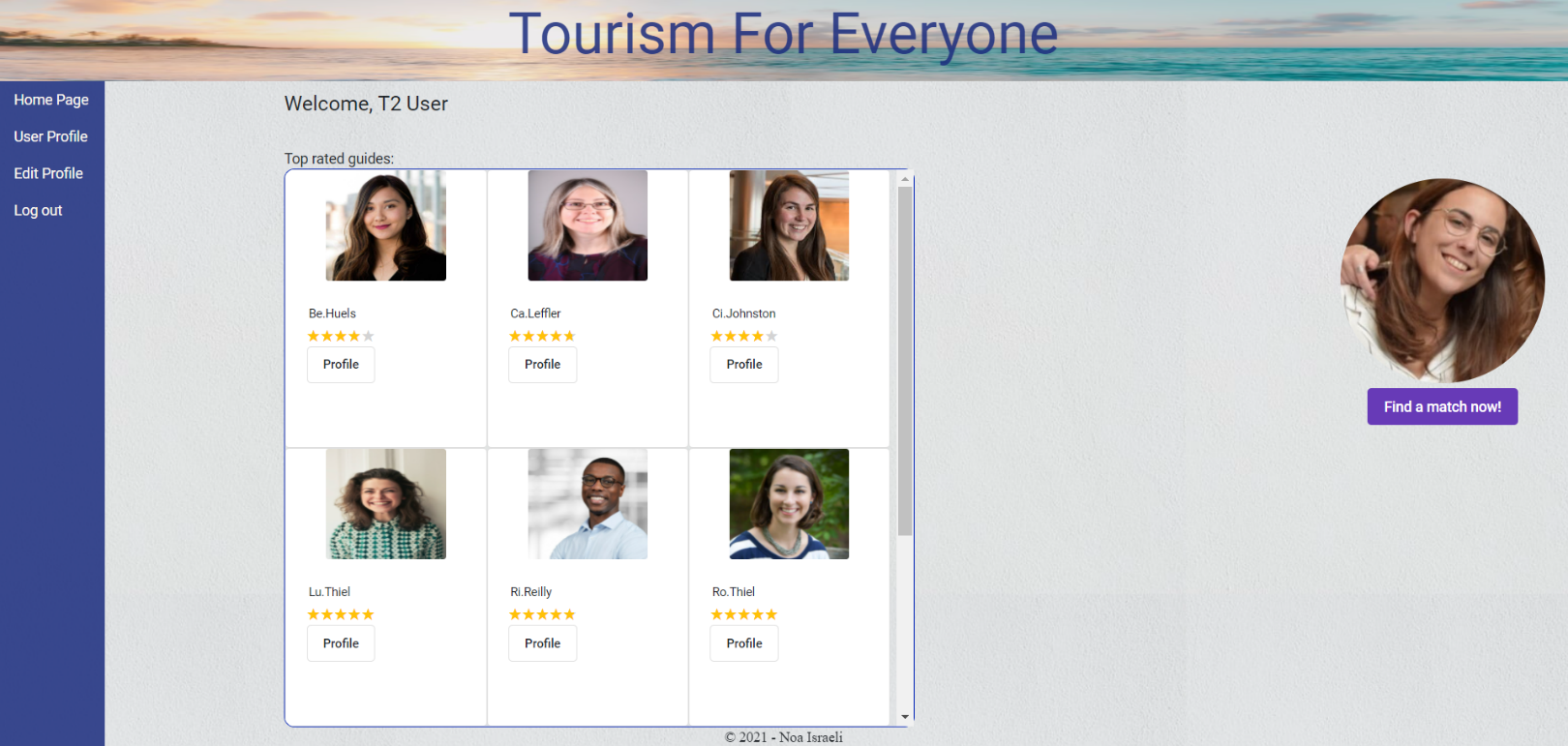
1. **AuthService:**

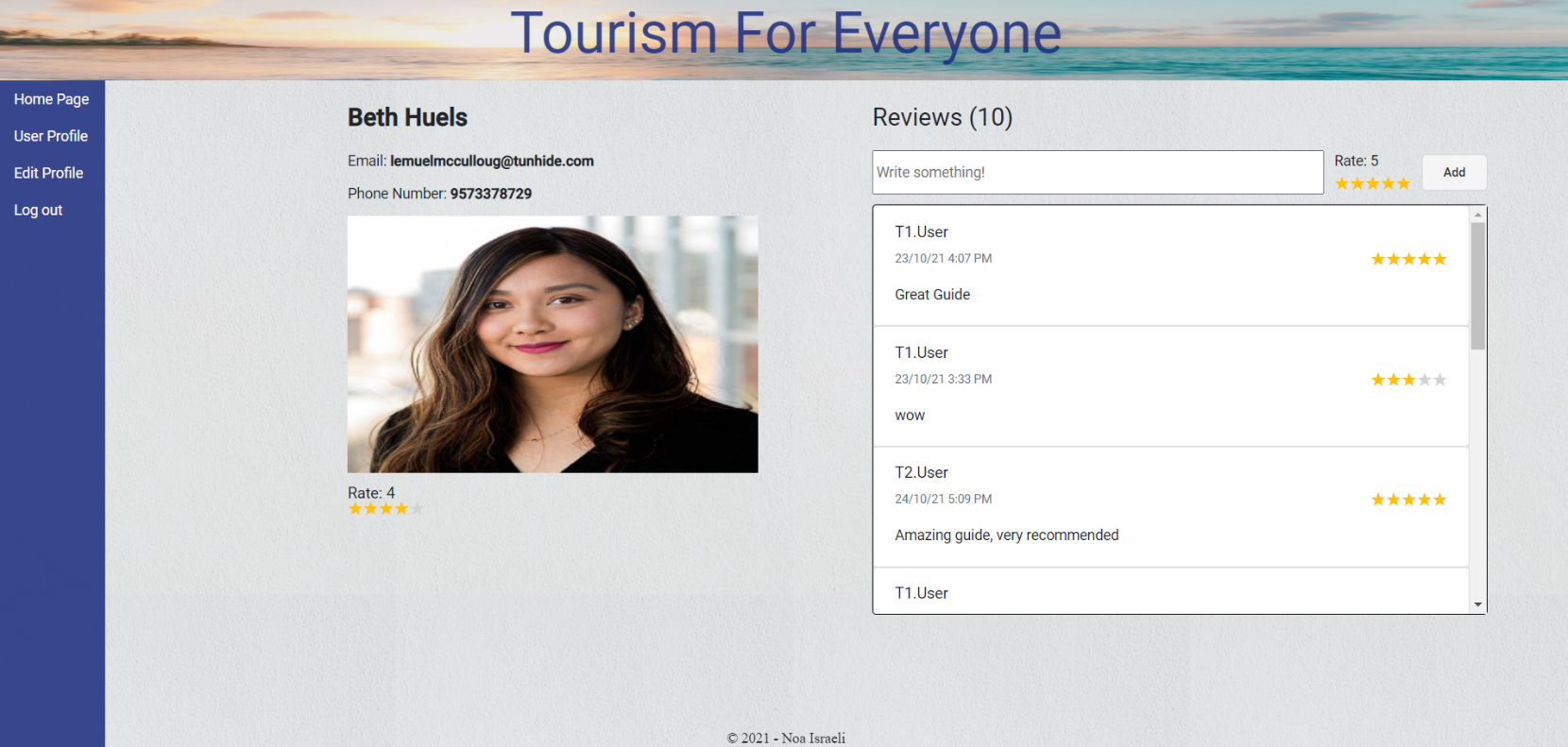
* אחראי על הבאת המידע על המשתמש המחובר כעת.
* אחראי על תקשורת עם שירות ה-authentication של firebase ועל פעולות שקשורות בכך כגון sign-in, sign-out, sign-up, וכו'.
* מחזיק את המחלקה שמייצגת user.

1. **ReviewsService:**

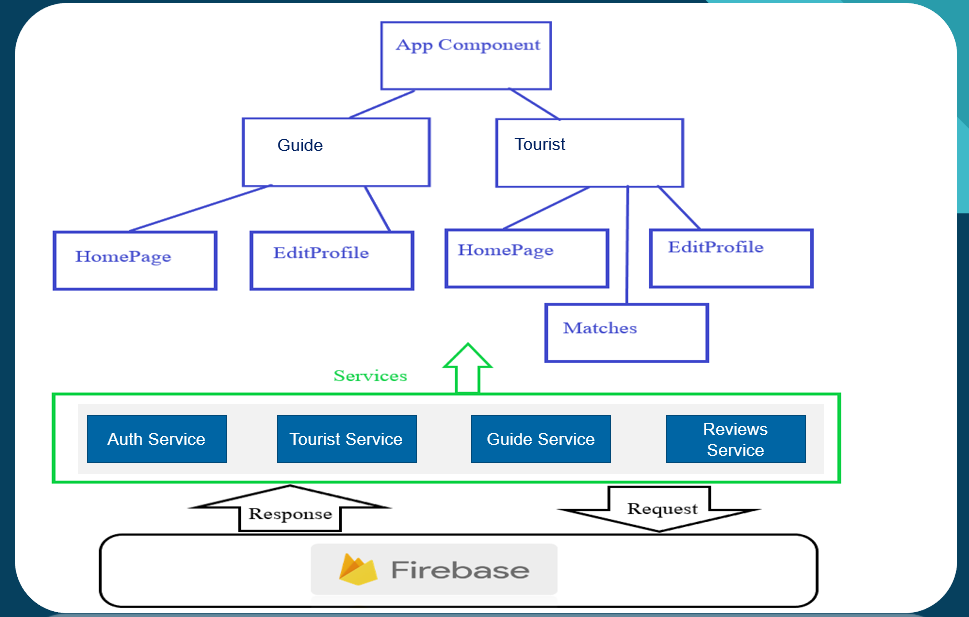
* אחראי על הבאת כל הביקורות שקיימות ב-DB למטרת מיון לכל מדריך באופן ספציפי.
* אחראי על פעולת הפרסום של ביקורת כלומר, הכנסתה ל-DB .
* מחזיק את המחלקה שמייצגת Review.

תמונה של עמוד כניסה:

תמונה של דף בית של משתמש מסוג תייר:

תמונה של דף בית של מדריך בכניסה ע"י פרופיל מסוג תייר:

ארכיטקטורה



הסבר על התרשים:

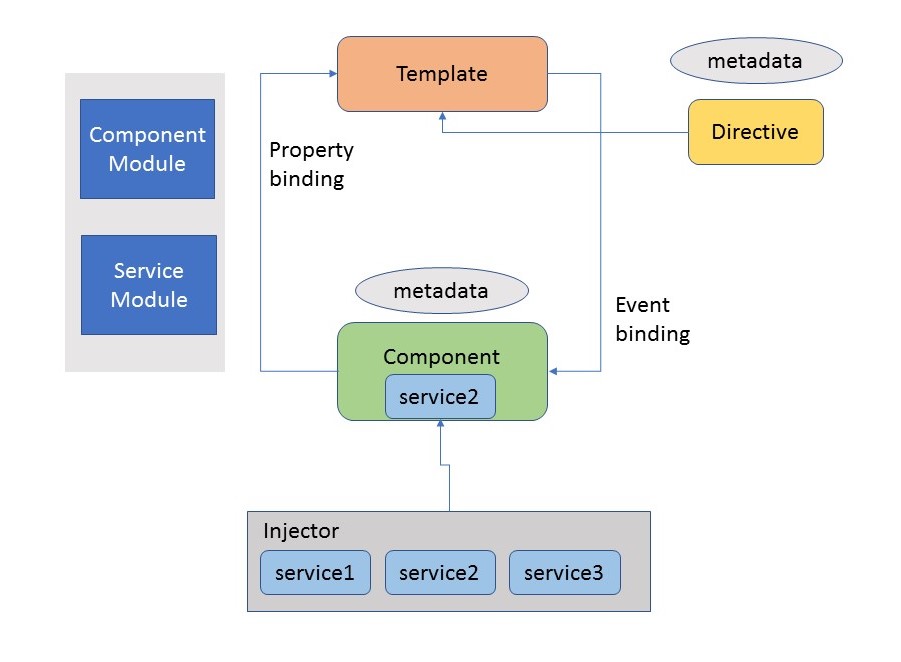
בכחול, ניתן לראות את היררכיית הקומפוננטות מהן מורכבת האפליקציה -   
מתחת לקומפוננטה הראשית App, ישנם שני מודולים עיקריים – תייר ומדריך. לכל אחד מהם יש קומפוננטות מותאמות אליו.

כל רכיב באפליקציה יכול להיות קומפוננטה, זה תלוי ב-Design שרוצים. באפליקציה שלי, החלטתי שכל דף יהיה קומפוננטה בפני עצמו, ובנוסף גם תפריט הניווט יהווה קומפוננטה. זאת לאחר שראיתי כי הוא לוקח המון קוד וכדאי להפריד.

החלוקה לקומפוננטות מאפשרת מודולריות נוחה מאוד של האפליקציה וגם כתיבת טסטים בצורה נוחה וספציפית.

בירוק, ניתן לראות את החלק של ה-Services. חלק זה בעצם מכיל שלושה שירותים שמתקשרים עם ה-DB ומביאים את המידע. השירות של התייר מביא מידע שנחוץ לקומפוננטות במודול של התייר, כנ"ל לגבי שירות המדריך.   
השירות Authentication מתקשר לא רק עם ה-DB אלא גם עם שירות ה-Authentication של Firebase ובעצם דואג להרשמה, כניסה והבאת המידע הרלוונטי לגבי המשתמש שמחובר כרגע לאתר בזמן אמת.

הסבר על Angular Component:



כל קומפוננטה באנגולר מורכבת משלושה חלקים – Class, Template ו- Metadata.

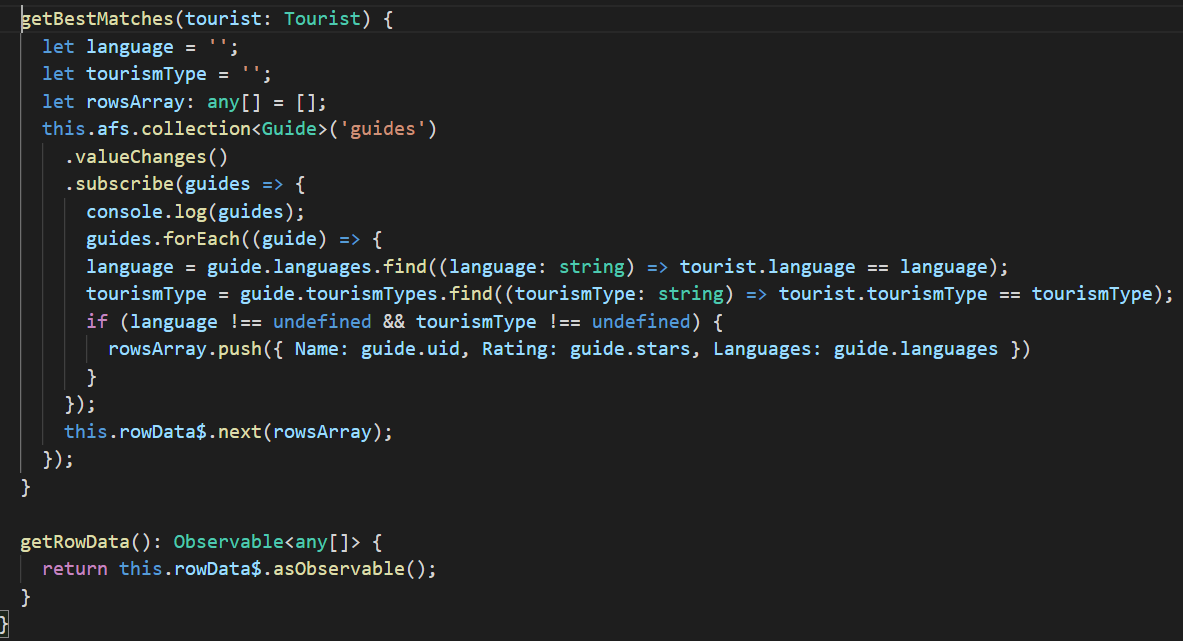
ה-Class הוא קובץ שכתוב בשפת TypeScript ושם נמצאת הלוגיקה, הפונקציות והפעולות שרוצים לעשות. ה-Template הוא בעצם קובץ ה-Html שאחראי על עיצוב האפליקציה ועל ה-UI. ב-Angular, שני החלקים יכולים לתקשר בינם לבין עצמם בצורה נוחה בעזרת **Data Binding** כמו שמצוירק בתרשים.   
ה-Metadata אלו בעצם כל התוספות וה-imports שיש לכל קומפוננטה.

דוגמאות

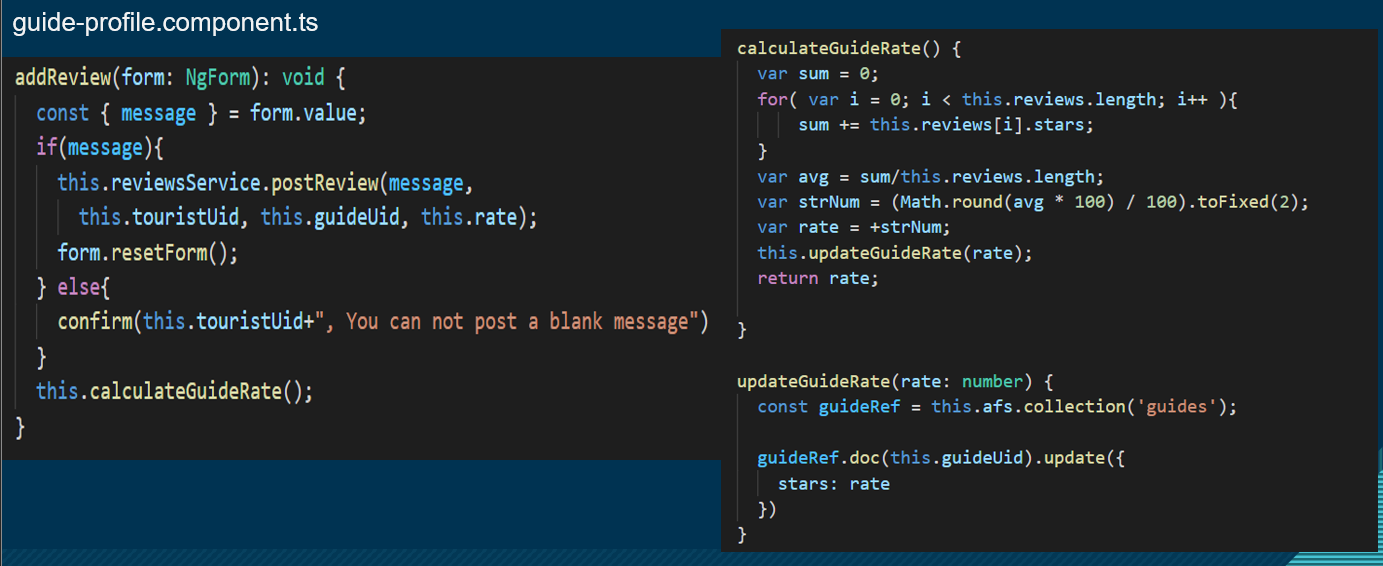
דוגמה לקומפוננטה של Matches:

כאן הקומפוננטה בעצם מביאה מה-Service את כל הפרטים שהיא צריכה לגבי התייר הנוכחי כדי למצוא לו התאמה – היא לוקחת את פרטי ה-user של התייר ואת פרטי ה-tourist שלו ע"י ה-Tourist Service ואחר כך קוראת לאלגוריתם ההתאמה (הקריאה מוקפת בעיגול אדום) שמפורט למטה.

דוגמה לקוד אלגוריתם התאמה:



הסבר:  
האלגוריתם בעצם מקבל אובייקט מסוג Tourist, מוציא את רשימת כל המדריכים שיש ב-DB ובודק לפי סוג התייר והשפות האם יש התאמה.   
במידה וכן, מוציא מאוביקט המדריך את שמו, הדירוג שלו והשפות אותן מדבר על מנת להציג אותם לתייר בתוצאות החיפוש.

דוגמה לקוד הוספת ביקורת על מדריך ע"י שימוש ב-reviews service ועדכון מידי ב-DB:

הסבר על התמונה:  
ברגע שתייר רוצה להוסיף ביקורת על מדריך מסוים, הפונקציה AddReview נקראית עם הטופס עצמו כפרמטר, הפונקציה קוראת ל-reviewsService שמעדכן את הביקורת ב-DB של הביקורת ב-Firestore, עושה ריסט לטופס וקורא לפונקציה לחישוב הדירוג החדש של המדריך (הפונקציה מצד ימין למעלה שנקראית CalculateGuideRate).  
בפונקציית החישוב הפונקציה מחשבת מחדש את הממוצע לפי כל הביקורות (מחשבת מחדש למקרה שמשתמש אחר הוסיף ביקורת במקביל, צריך לשמור על סנכרון) ואז קוראת לפונקציה updateGuideRate עם פרמטר של הדירוג החדר והפונקציה הזו (ימין למטה) מעדכנת את ה-DB של המדריך הרלוונטי עם הדירוג החדש שחושב.

טכנולוגיות

צד לקוח:

1. Angular
2. TypeScript
3. HTML
4. CSS
5. Bootstrap

צד שרת:

מסד נתונים – Data Base :

1. FireBase Authentication – אחסון וניהול משתמשים
2. Firestore Data Base – אחסון נתוני המדריכים והתיירים לאלגוריתם ההתאמה

ניהול גרסאות:

1. Git
2. GitHub

אחסון אתר: firebase

סיכום

כאשר התחלתי לבנות את האפליקציה, לא ידעתי בכלל מה זה אנגולר. נכנסתי לעולם הזה ולמדתי את הכל מאפס לגמרי בעצמי. האתר הוא בעצם העבודה הראשונה שלי באנגולר מתחילתה ועד סופה. נהניתי מאוד לכתוב, לחקור את שפת ה-Typescript ואת ה-Framework הייחודי של Angular.

כיום, אני עובדת כמפתחת אפליקציות Web ומתעסקת בכל יום בדיוק בדברים ובטכנולוגיות שקיימות בפרויקט. אני מרגישה שהפרויקט הזה נתן לי את הזריקה הראשונית שהייתי צריכה למים כדי לעלות ולהצליח בתחום.

ביבליוגרפיה

1. <https://angular.io/>
2. <https://firebase.google.com/docs>
3. <https://getbootstrap.com/>

קישורים

קישור ל-GitHub - <https://github.com/noaisr1/AngularFinalProject.git>

קישור לאתר - <https://angular-my-project-6583d.firebaseapp.com/>