

בניית מערכות ממוחשבות מבוססות WEB

חלק 3 – צד שרת

קישור ל-Repository הפרוייקט:

<https://github.com/VeredBarArbel/trempeMate.git>

הנחות כלליות

- בעת הרשמה לאתר נדרש להזין רק אימייל וסיסמא, ובהמשך ניתן להוסיף פרטים אישיים בפרופיל המתמש. ההנחה הזו נובעת מהרצון להקל על users חדשים ולא להעמיס עליהם בעת ההרשמה, וזה יכול להעלות את מספר המשתמשים בשירות.
- ניתן להיות בעל פרופיל נהג ולא להזין את פרטי הרכב בפרופיל (הנחה הנובעת מפרטיות).
- ניתן להירשם כנוסע פעם אחת בלבד לכל נסיעה.
- כדי לשנות את מספר המקומות יש למחוק את ההשתתפות בנסיעה ולהוסיפה מחדש.
- לא ניתן להירשם לנסיעה אם מספר המקומות הרצוי גדול ממספר המקומות שנותרו בנסיעה.
- לא קיימת הגבלה על מספר המקומות שנהג יכול להציע בנסיעה.
- נסיעות שלא נותר בהן מקום פנוי לא יופיעו ברשימת הנסיעות המוצעות למשתמשים, גם אם תאריך יציאתן מאוחר מתאריך החיפוש. כאשר יתפנה מקום הנסיעה תופיע ברשימת החיפוש.
- נהג יכול לשריין מקומות בנסיעה של עצמו.

שימוש בפרוייקט הבסיס הנתון באתר

בחלק זה בפרוייקט עסקנו בצד השרת באתר ובניית בסיס נתונים רלוונטי עבורו. במסגרת חלק זה, המרנו את הפרוייקט הראשוני שיצרנו לקובץ flask במבנה blueprint. לשם המרת הפרוייקט לקובץ flask השתמשנו בסביבת העבודה שקיבלנו במסגרת הקורס. בתוך תיקיית pages יצרנו תיקייה לכל עמוד html שבנינו בחלק הקודם (עם תוספות עדכניות לחלק זה):

- תיקיית static- מכילה תיקיות הקבצים הסטטיים (תמונות, JS או CSS) בתוך כל תיקייה נמצאים הקבצים הרלוונטים לעמוד בלבד.
 - תיקיית templates – מכילה את דפי html הרלוונטים.
 - קובץ python- בקובץ זה יצרנו Blueprint לכל עמוד שיתקשר עם app.py החיצוני. כמו כן, בקובץ זה מוצגים הנתונים הרלוונטיים לאותו עמוד.
- בתיקיית הפרוייקט קיימות תיקיות static ו-template המכילות קבצים בעלי תפקיד משותף למספר דפים ובעלות תפקוד זהה למוזכר מעלה.
- קובץ app.py של הפרוייקט מכיל קישור blueprint עבור כל הדפים הרלוונטים למימוש הפרוייקט.

יצירת בסיס נתונים

לאחר מכן יצרנו בסיס נתונים על פי סמכת ה-ERD שיצרנו בחלק א', בשילוב שינויים שערכנו בחלק ב'. שם בסיס הנתונים: web-project-g17. בסיס הנתונים מכיל את הטבלאות הבאות:

שם הטבלה	משמעות בבסיס נתונים	מפתח ראשי	מפתח זר	שדות מוגדרים
cities	רשימת הערים מהן ניתן ליצור/להזמין נסיעה	city	-	city – varchar(255)

Id – int Email - varchar(255) Description – text DT - timestamp	-	id	טבלה המכילה את פרטי המשתמשים שמילאו טופס יצירת קשר באתר	contacts
email - varchar(255) first_name- varchar(255) last_name- varchar(255) create_date - timestamp about_me - text phone_number- varchar(10) password- varchar(120) car_type- varchar(100) car_color- varchar(100)	-	email	מציגה את המשתמשים הרשומים לאתר	users
Trip_id - int driver - varchar(255) pick_up_date – date pick_up_time – time pick_up_city – varchar(255) drop_city – varchar(255) available_seats – int price – int passangers – int	<u>Driver</u> - users(email)	trip_id	טבלה השומרת את כלל הנסיעות שבוצעו באתר	trips
trip – int user - varchar(255) seats_amount – int	<u>user</u> - users(email) <u>trip</u> - trips(trip_id)	Trip,user	טבלה השומרת את רשימת העצירות עבור כל נסיעה	tripuser

לאחר יצירת טבלאות בבסיס הנתונים הזנו רשומות לכל אחת מהן כדי שבהמשך נוכל ליצור שאילתות שונות להצגת דגם של אתר מתפקד.

יצירת מחלקות

יצירת המחלקות נעשתה בתיקיית utilities שם נבנו שלוש מחלקות המפורטות מטה. השימוש במחלקות אפשר לנו להציג את שלושת האובייקטים, להעניק מאפיינים ושיטות כך שנוכל להשתמש בהם בשלב מאוחר יותר, ולאפשר אינטראקציה בין האובייקטים השונים והמידע המוזן להם.

בנוסף לשדות יעודיים עבור כל מחלקה, יצרנו בנאי, ופונקציות CRUD.

מחלקת User:

מחלקה זו יוצרת את האובייקט משתמש רשום ושדות המחלקה הם עמודות טבלת users. למחלקה זו יצרנו בנאי היוצר ישות של משתמש רשום, ובנוסף לכך יצרנו שיטת insert בשם register_user, שיטת update בשם update_user, שיטת check_password, ושיטת user_session, על מנת לשמור על חיבור המשתמש באופן גלובלי בכל הדפים באתר.

נשתמש בשיטת רישום המשתמש בעת יצירת משתמש חדש (פונקציית signup בקובץ Welcome.py) ובשיטת עדכון המשתמש בעת עדכון משתמש (פונקציית updateProfile בקובץ UserProfile.py).

מחלקת Trip:

מחלקה זו יוצרת את האובייקט נסיעה, המכילה את עמודות טבלת trips. למחלקה זו בנינו בנאי שיצור ישות של נסיעה, get לשדה מזהה הנסיעה (trip_id), ושיטת insert בשם create_trip. בשיטה האחרונה נשתמש בעת יצירת נסיעה (פונקציית submitARide בקובץ AddARide.py).

המחלקה ContactDetails:

מחלקה זו יוצרת את האובייקט טופס פניית משתמש באתר. ביצענו הנחה שרק משתמשים רשומים יכולים ליצור קשר באמצעות טופס זה על מנת להבטיח את אמינות הנהגים והנסיעות. שדות המחלקה הם עמודות טבלת contacts. במחלקה זו יצרנו בנאי ושיטת insert בשם create_request. נשתמש בשיטה זו בעת יצירת טופס פנייה (פונקציית submitContact בקובץ contactUs.py).

קובץ general.py

בנוסף למחלקות, יצרנו את עמוד הפיתוח general תחת תיקיית utilities. עמוד זה מכיל את הפונקציות הכלליות שנשתמש בהן במסגרת פעילות האתר. פירוט הפונקציות מופיע בטבלה הבאה:

שם הפונקציה	מטרה ושימוש
get_user	פונקציה זו מחזירה לנו פרטי היוזר המחובר על מנת להזין את פרטיו לפונקציית ה-session.
get_trip	יוזר מחפש נסיעה ובוחר בנסיעה מסוימת. פונקציה זו מחזירה את פרטי הנסיעה לפי ה-trip_id.
save_spot	פונקציית זו מקבלת את מזהה הנסיעה, מזהה המשתמש ומספר מקומות רצוי, ומוסיפה את המשתמש לנסיעה אם יש מקום בטרמפ ואם הוא לא נרשם אליה לפני כן.
driver_history	פונקציה זו מציגה את היסטוריית הנסיעות של המשתמש בתור נהג הנסיעה.
trempp_history	פונקציה זו מציגה את היסטוריית הנסיעות של המשתמש בתור טרמפיסט.
remove_reg	פונקציית זו מאפשרת למשתמש לבטל הרשמתו לנסיעה, ומעדכנת את טבלת הנסיעות והעצירות בהתאם.
search_ride	פונקציית זו מציגה לטרמפיסט את רשימת הנסיעות הרלוונטיות עבורו בעת חיפוש עיר איסוף, עיר הורדה ותאריך ספציפי. אם אין נסיעה מתאימה עבורו הפונקצייה מתריאה על כך.
city	פונקציה זו מציגה את רשימת הערים המשתתפות בביצוע הנסיעות ב-TremppMate.