

בניית מערכות ממוחשבות מבוססות WEB

Rabbit – חלק ג'

מגשים:

רוני מנדלסון 205664741, מאיה זקן 208308288

שגיא רוטשטיין 311242762, דור חן 311410484

מטרת חלק זה היא להשתמש בכלים שלמדנו עד כה בקורס כדי לעצב ולבנות את צד השרת של אתר Rabbit.

עדכונים ושינויים מחלק ב':

- הפרדת קבצי JS, CSS, HTML ותמונות וסידורם ב-Blueprint.
- מילוי שדות action ו-method בהתאם לפונקציות המוגדרות בטפסים השונים.

הנחות יסוד:

- בכדי לעשות שימוש ברוב הדפים באתר, יש תחילה להתחבר לאתר. בחרנו לממש זאת כך מכיוון שרצינו לשמור על אמינות המשתמשים והדירוגים באתר.
- כדי לדרג נהג או נוסע צרך קודם כל להיות חייבור ביניהם בנסיעה אחת לפחות
- בעמוד זה יוצגו רק נסיעות בסטטוס "Running" כלומר, נסיעות שעדיין רלוונטיות. עדכון הסטטוס יעשה ע"י הנהג (יפורט בהמשך).

טבלאות בבסיס נתונים:

- **users** : {ID, Name, Email, Password, DOB, FacebookLink, LicenseNumber, CarNumber, Company, CarModel, Phone, Profile}
- **rides**: {ID, **DriverID** (users), DepartureTime, Origin, Destination, Price, Seats, Mask, Pets, Smoking, Comment, Status}
- **rateus**: {ID, UserID, Rate, Message}
- **rating**: {ID, **ToID** (users), **FromID** (users), **OnID** (rides), Rate, Message}
- **joinride**: {ID, RideID, PassengerID, Pickup, Dropoff, Message}
- **contact**: {ID, Email, Message, Name, userID}

הסברים על הטבלאות:

users: מפתח ראשי- כל משתמש מקבל אוטומטית מזהה ראשי על ידי בסיס הנתונים.

הטבלה מכילה את כל פרטי הלקוחות כולל פרטי התחברות לאתר.

rides: מפתח ראשי- כל נסיעה מקבלת אוטומטית מזהה ראשי על ידי בסיס הנתונים.

הטבלה מכילה את כל פרטי הנסיעה כולל זהות הנהג (כמפתח זר מטבלת users), דרישות ספציפיות שיש לו מהנוסעים ואת סטטוס הנסיעה (פתוחה או הסתיימה).

rateus: מפתח ראשי- כל רשומה מקבלת אוטומטית מזהה ראשי על ידי בסיס הנתונים.

הטבלה מכילה את דירוג האתר והערות שנכתבו ע"י משתמשי האתר.

rating: מפתח ראשי- כל רשומה מקבלת אוטומטית מזהה ראשי על ידי בסיס הנתונים.

הטבלה מכילה 3 מפתחות זרים, שניים אשר מתקבלים מטבלת users ומצביעים על המשתמש המדרג והמדורג, ואחד שמתקבל מטבלת rides ומצביע על הנסיעה המשותפת דרכה מדרגים. בנוסף הטבלה מכילה את הדירוג הממוצע והערות.

joinride: מפתח ראשי- כל רשומה מקבלת אוטומטית מזהה ראשי על ידי בסיס הנתונים.

הטבלה מכילה את פרטי הנסיעה שממלא הנוסע בעת הצטרפותו לנסיעה קיימת ומחברת בין הנסיעה לנוסע.

contact: מפתח ראשי- כל רשומה מקבלת אוטומטית מזהה ראשי על ידי בסיס הנתונים.

הטבלה מכילה את ההודעות אשר משתמש שלח לאתר.

הבסיס:

מכיל את ה-Header בין דפי ה-html המוצגים באתר. עמודי ה- HTML עושים Extend לבסיס ודורסים את ה- Block הרלוונטי כך שכל עמוד מתעדכן עם התוכן הרלוונטי לו.

עמוד success.html מרחיב את base.html ומכיל Block בו נשתמש בדפים השונים עם הודעה למשתמש האם הצליח בפעולתו וקישור לעמוד הבית.

תיאור מחלקות:

UserModel

המחלקה כוללת את הפונקציות הבאות:

- Register - מאפשרת הרשמה של משתמש חדש לאתר.
- Login - מאפשרת התחברות לאתר לאחר ביצוע הרשמה.
- Update - מאפשרת עדכון של פרטי המשתמש.
- GetUser - מחזירה את פרטי המשתמש המבוקש מתוך בסיס הנתונים.

RideModel

המחלקה כוללת את הפונקציות הבאות:

- AddRide - מאפשרת הוספת נסיעה לבסיס הנתונים.
- ViewAll - מחזירה את פרטי הנסיעות המפורסמות באתר.
- ViewMyRides - מחזירה את פרטי הנסיעות בהן הנהג הספציפי השתתף.
- MarkAsEnd - מעדכנת את סטטוס הנסיעה הרלוונטית מ"פתוחה" ל"הסתיימה".
- ViewRides - מחזירה את פרטי הנסיעות הרלוונטיות לאחר ביצוע חיפוש מסוים.

RateModel

המחלקה כוללת את הפונקציה הבאה:

- AddRate - מוסיפה דירוג משתמש לבסיס הנתונים.

RankUsModel

המחלקה כוללת את הפונקציה הבאה:

- AddRank - מוסיפה דירוג אתר לבסיס הנתונים.

JoinRide

המחלקה כוללת את הפונקציות הבאות:

- JoinRide - מוסיפה לבסיס הנתונים רשומה המהווה הצטרפות של נוסע לנסיעה.
- ViewMyRides - מחזירה את פרטי הנסיעות בהן הנוסע הספציפי השתתף.
- ViewMyPassengers - מחזירה את פרטי הנוסעים שנסעו עם הנהג בנסיעה.

ContactModel

המחלקה כוללת את הפונקציות הבאות:

- AddContact - מוסיפה לבסיס הנתונים הודעת יצירת קשר ע"י המשתמש.
- ViewMessage – מחזירה את כל ההודעות שהמשתמש הספציפי שלח.
- deleteMessage – מאפשרת למשתמש למחוק הודעה ששלח.

תיאור דפים:

AboutUs: דף אודות החברה.

Contacts: דף יצירת קשר עם החברה.

Homepage: עמוד הבית של האתר.

JoinTheRide: דף הצטרפות לנסיעה קיימת. מגיעים לדף זה כאשר לוחצים על כפתור הצטרפות לנסיעה מתוך דף חיפוש הנסיעות. דף זה מכיל טופס בו הנוסע מעדכן את נקודת האיסוף וההורדה שלו ויכול להשאיר הערות לנהג.

MyRides: מציג את הנסיעות של המשתמש בתור נוסע ובתור נהג. בעמוד זה יוכל נהג לעדכן את סטטוס נסיעתו מנסיעה רלוונטית (Running) ללא רלוונטית (END) ובצורה זו להסיר אותה מרשימת הנסיעות המוצגות לשאר המשתמשים (עמוד SearchResults).

MyPassenger: מציג לנהג את הנוסעים שנסעו איתו. מעמוד זה יש קישור לדירוג הנוסעים.

Page_not_found: דף שגיאה המכיל קישור לדף הבית.

RankDriver: עמוד בו נוסע יכול לדרג את הנהג איתו הוא נסע. הציון אותו הנהג מקבל בעת הדירוג מהווה ממוצע של שלושת הקריטריונים לדירוג, והדירוג הסופי שיוצג בעמוד המשתמש מהווה ממוצע של כל הדירוגים שמשתמש זה קיבל.

RankPassenger: עמוד בו נהג יכול לדרג את הנוסעים שנסעו איתו. הציון אותו הנוסע מקבל בעת הדירוג מהווה ממוצע של שלושת הקריטריונים לדירוג, והדירוג הסופי שיוצג בעמוד המשתמש מהווה ממוצע של כל הדירוגים שמשתמש זה קיבל.

RankUs: עמוד בו המשתמש יכול לדרג את האתר.

SearchResults: עמוד המציג את הנסיעות הקיימות, ניתן גם לבצע בו חיפוש נסיעה לפי שדות מסוימים. בעמוד זה נהג שפרסם נסיעה כלשהיא לא יוכל לראות את נסיעתו אלא רק נסיעות של נהגים אחרים אליהן יוכל להצטרף. בנוסף, בעמוד זה יוצגו רק נסיעות בסטטוס "Running" כלומר, נסיעות שעדיין רלוונטיות.

SignIn: עמוד הרשמה והתחברות לאתר.

update_profile: עמוד ובו טופס המאפשר למשתמש לעדכן את פרטיו.

StartARide: עמוד בו נהג יכול לפרסם נסיעה חדשה.

UserProfile: עמוד המציג את פרטי המשתמש. בנוסף, קיים קישור לעמוד הפייסבוק של המשתמש, עמוד עדכון פרופיל (update_profile) ועמוד הנסיעות שלו (MyRides).

MyMessages: עמוד המציג את כל ההודעות שמשתמש ספציפי שלח לאתר, ונותן אפשרות למחוק את ההודעות.

שאלות:

לאורך הפרויקט כתבנו מספר שאלות SQL. כעת נפרט על חלקן.

• SELECT – דוגמא לשאלתה:

התחברות לאתר- בעת חיבור משתמש לאתר ישנו צורך לזהות האם יש התאמה בין הנתונים המוזנים בטופס הכניסה לבין בסיס הנתונים.

```
def Login(self, email, password):
    query = "SELECT ID FROM users WHERE email=%s AND password=MD5(%s)"
    return dbManager.fetch(query, (email, password,))
```

• INSERT - דוגמא לשאלתה:

הוספת נסיעה- נהג יכול להוסיף נסיעה, הפרטים אותם הוא ממלא נשמרים בבסיס הנתונים בטבלת rides.

```
def AddRide(self, driver, date, origin, destination, price, seats, animals, masks, smoke, msg):
    query = "INSERT INTO rides(DriverID, DepartureTime, Origin, Destination, Price, Seats, Mask, Pets, Smoking, Comment, Status) \
    VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, 'RUNNING')"
    return dbManager.commit(query, (driver, date, origin, destination, price, seats, masks, animals, smoke, msg,))
```

- **UPDATE – דוגמה לשאילתה:**

שינוי סטטוס נסיעה- נהג יכול לשנות את שדה "status" מטבלת rides מ-Running ל-END, כלומר להודיע שהנסיעה כבר אינה רלוונטית.

```
def MarkAsEnd(self, rideID):  
    query = "UPDATE rides SET Status='END' WHERE ID=%s"  
    return dbManager.commit(query, (rideID,))
```

- **UPDATE – דוגמה לשאילתה:**

עדכון פרופיל משתמש- משתמש יכול לשנות את פרטיו. המשתמש מחויב להכניס את סיסמתו על מנת לעדכן את פרטיו.

```
def Update(self, ID, name, email, dob, password, fb, license, num, comp, model, phone, profile):  
    query = "UPDATE users SET Name=%s, Email=%s, DOB=%s, password=MD5(%s), FacebookLink=%s, \" \\  
            \" LicenseNumber=%s, CarNumber=%s, Company=%s, CarModel=%s, Phone=%s, Profile=%s WHERE ID=%s"  
    return dbManager.commit(query, (name, email, dob, password, fb, license, num, comp, model, phone, profile, ID))
```

- **DELETE – דוגמה לשאילתה:**

מחיקת הודעה שנשלחה – המשתמש יוכל לצפות בהודעות ששלח להנהלת האתר ובמידה והוא מתחרט ורוצה לבטל את אחת ההודעות הוא יכול ללחוץ על DELETE וההודעה תימחק.

```
def deleteMessage( ID):  
    query = "DELETE FROM contact WHERE contact.ID = %s"  
    return dbManager.commit(query, (ID,))
```