פרויקט קורס web – חלק 3

****

**קבוצה 3**  
דנה רונן 203070461  
איתמר גרובר 204282867  
אפרת דובי 315587378  
עמית גבאי 316088525

Carlos & Empanadas

**ארגון מחדש של הפרויקט**

בחלק השני של הפרויקט עבדנו על צד הלקוח של האתר. צד השרת של האתר היה מורכב מאוסף של קבצי html שמקושרים עם קבצי html אחרים ולקבצי css ו- js. לקראת תחילת העבודה על צד השרת של האתר ראינו לנכון תחילה לארגן את המבנה של הפרויקט מחדש כך שיאפשר הפרדה ברורה בין כל עמוד באתר ויאפשר לנו לעבוד בצורה מסודרת. בשלב זה, עדיין לא ערכנו את הקוד ולא השתמשנו בJinja.

**מבנה הפרויקט הישן:**

* html – תיקייה שהכילה באופן מרוכז את כל קבצי ה html.
* css – תיקייה שהכילה באופן מרוכז את כל קבצי ה - css.
* js – תיקייה שהכילה באופן מרוכז את כל קבצי ה - js.
* images – תיקייה שהכילה את כל קבצי המדיה שיוצגו באתר.

**מבנה הפרויקט החדש:**

* components – תיקייה שמכילה אלמנטים בהם יש שימוש במספר עמודים, זאת כדי להימנע מכפל קוד ולאפשר תחזוקה מהירה ונקייה של הקוד.
* pages – תיקייה אשר מכילה תיקייה עבור כל עמוד באתר. שמה של כל תיקייה הוא על פי שם העמוד אותו היא מכילה.
  + תיקייה של עמוד מכילה בתוכה את הקבצי html, css וה js ששייכים לאותו העמוד. בנוסף, כל תיקייה של עמוד מכילה בתוכה את המדיה הרלוונטית לעמוד וקובץ python לצד השרת שיטפל בפניות הרלוונטיות לעמוד.
* templates – תיקייה אשר מכילה את הקובץ תבנית html הבסיסית base.html שמסדר ומארגן את הlayout- של כל העמודים באתר.
* static – מכילה קבצי css, js ומדיה אשר רלוונטיים לכל דף באתר.
* utilities – מכילה את הלוגיקה של הקוד צד שרת, הכוללת מחלקות שעוטפות פניות למסד הנתונים.
* SQL queries – תיקייה שמכילה את השאילתות ליצירת הטבלאות והוספת נתונים כפי שנדרש לצרף.
* קבצי app.py, settings.py, .env – להרצה של השרת.

**יצירת צד שרת בסיסי**

תחילה, סידרנו וערכנו את קטעי הקוד של צד הלקוח. לאחר מכן, פנינו לבניית שרת flask שיקלוט את בקשות המשתמש על כתובת ip ו- port ויצליח להגיש לנו את הקבצי צד לקוח הרלוונטיים.

עברנו על כל קבצי ה-python של כל עמוד לפי הארכיטקטורה החדשה והוספנו endpoints בעזרת הספרייה flask על מנת להגיש את העמודים הרלוונטיים אליהם. בקובץ app.py אתחלנו את שרת הflask ורשמנו את כל הblueprints- שיצרנו.

**עבודה עם Jinja**

הקובץ base.html שניתן לנו היווה את הבסיס לתבנית של העמודים השונים באתר. כל עמוד השתמש ב-extend לקובץ התבנית base.html והרחיב בלוקים בתוכן העמוד במידת הצורך. השימוש בתבנית סייע לנו להימנע מכפילויות בקוד. לאחר השימוש בתבנית base.html בכל העמודים, ייצאנו את סרגלי הניווט לתיקיית components. במקום לכתוב בכל עמוד את סרגלי הניווט, כתבנו אותם פעם אחת ובכל פעם השתמשנו ב-Include בקובץ התבנית.

**יצירת מסד נתונים**

לאחר שסיימנו לסדר את צד הלקוח, התחלנו ביצירת תשתית לצד השרת אמצעות תוכנת mySQL. mySQL הינו מסד נתונים רלציוני מבוסס SQL. ראשית כל, יצרנו טבלאות:

* users - טבלת משתמשים:
  + id – מפתח ראשי
  + email – דואר אלקטרוני של המשתמש
  + full\_name – השם המלא של המשתמש
  + password – הסיסמה של המשתמש
* contact\_us – טבלת פניות לקוחות:
  + id – מפתח ראשי
  + message – ההודעה שהלקוח השאיר לקרלוס
* cooking\_class – טבלת הרשמה לסדנאות אמפנדס עם קרלוס:
  + id – מפתח ראשי
  + email – דואר אלקטרוני של הנרשם לסדנא
  + first\_name – שם פרטי של הנרשם לסדנא
  + last\_name – שם משפחה של הנרשם לסדנא
  + phone\_number – מספר טלפון של הנרשם לסדנא
  + group\_size – גודל הקבוצה של הנרשם לסדנא
* products – טבלת המוצרים של קרלוס:
  + id – מפתח ראשי
  + product\_name – שם המוצר
  + price – מחיר המוצר
  + img – נתיב לתמונה של המוצר לתצוגה באתר
* orders – טבלת הזמנות של משתמשים:
  + id – מפתח ראשי
  + payer\_id – מפתח זר לשדה id שבטבלת users. המשתמש שביצע את ההזמנה ושילם.
  + order\_cost – הסכום לתשלום על ההזמנה
  + date – תאריך ביצוע ההזמנה
* shopping\_cart – טבלה שמכילה את כל המוצרים שהוזמנו בכל הזמנה:
  + id – מפתח ראשי
  + product\_id – מפתח זר לשדה id שבטבלת products. מצביע לתוצר ספציפי שהוזמן בהזמנה ספציפית (קשר של רבים לרבים)
  + order\_id – מפתח זר לשדה id שבטבלת orders. מצביע להזמנה שבה המוצר הוזמן.
  + quantity – הכמות שבה המוצר הוזמן

לכל טבלה הוספנו נתונים כדי שבסיס הנתונים לא יהיה ריק כאשר השאילתות של הטבלאות מצורפות בתיקיית SQL queries. לאחר הוספת הטבלאות, חיברנו את הפרויקט למסד הנתונים בקובץ .env

**לוגיקת צד שרת והצגת תוכן דינאמי**

אחד החלקים החשובים בעבודה עם Jinja ובפרויקט בכלל היה להפוך את התוכן הסטטי של האתר להיות דינאמי. במקום להציג נתונים תוך כתיבה ישירה שלהם אל קבצי המקור של צד הלקוח, בעזרת jinja אנחנו יכולים להפוך את צד הלקוח לדינאמי, לקבל נתונים מהשרת ו"לשתול" אותם בעמודים.

ברוב בקשות GET של הלקוח היינו צריכים לייבא נתונים מהמסד נתונים, ולכן יצרנו תחת התקייה utilities מחלקה עבור כל טבלה. כל מחלקה מכילה בתוכה פונקציות סטטיות אשר עושות שימוש באובייקט dbManager שניתן לנו. המחלקות הנ"ל היו אלו שעטפו את שאילתות הSQL-, והאובייקט dbManager קישר בין המחלקות למסד הנתונים לצורך שליפת המידע ממנו. לדוגמא, פונקציית Products.get\_all\_products() שמחזירה את כל השדות של כל המוצרים בטבלת products, או פונקציית Products.get\_products\_by\_ids(product\_ids) שמקבלת מערך של מזהי מוצרים ומחזירה את המוצרים בטבלת products שהמזהים שלהם נמצאו.

במקרים אחרים, בהם הלקוח שולח בקשת POST, לרוב היה מדובר בפנייה לשרת לצורך הוספה או עדכון של טבלה במסד הנתונים. פעולה מסוג זה דרשה קודם לאסוף את הנתונים שהוזנו בטופס הרלוונטי בדף ולשלוח אותם בבקשת POST לשרת לנתיב הרלוונטי. לדוגמא, כאשר לקוח רוצה להירשם לאתר, היינו צריכים קודם לאסוף את המייל, השם של המשתמש והסיסמה מהטופס בעמוד הhtml-, לשלוח את הנתונים לנתיב המתאים בשרת ולהמתין לתשובה. השרת בדק האם המייל כבר רשום במערכת בטבלת users. במידה ולא, יש להוסיף את המשתמש כמשתמש חדש לאתר.

הנתונים הדינאמיים שהושגו מהשרת בעזרת המסד נתונים הועברו ללקוח והושתלו בתוך העמודים בעזרת jinja.

**הנחות**

החלטנו שרק משתמשים רשומים יוכלו לבצע הזמנות ולכן כאשר משתמש אורח נכנס לעמוד ההזמנה, אנו מעבירים אותו אוטומטית לעמוד ההתחברות\הרשמה כדי שיוכל להמשיך בהזמנה.