**Python Full Stack Project - Part I**

**הילה ניר, 204719140**

הפרויקט שלי מחולק לשני apps:orders, manager. החלטתי לעשות לapps האלה, מתוך הרעיון שיש פעולות שהלקוח יכול לעשות, ויש פעולות של בק אופיס (פאנל ניהול) שרק המנהל יכול לעשות.

הלקוח יכול להירשם. לאחר הרשמה עם פרטים אישיים והתחברות עם שם משתמש וסיסמא הלקוחות יוכלו לבחור מתוך רשימה של מנות ולהזמין משלוח.

במקביל מסופק אתר בק אופיס (פאנל ניהול) אליו יתחברו חשבונות מנהל. הפאנל יאפשר צפייה במשלוחים פעילים וסימון משלוחים שבוצעו. בנוסף מנהלים יוכלו לנהל את המנות והקטגוריות שמוצעות באתר.

בכל הנוגע לטפסים (forms), השתמשתי בטפסים חכמים, ModelForms, של Django ולא input ידני.

CategoryForm ו-DishForm הם משמשים ליצירה ולעריכה של מופעים של אובייקטים מסוג קטגוריה ומנה.

NewUserCreationForm ו-EditUserForm הם טפסים מובנים המשמשים ליצירה ולעריכה של מופעים של מודל היוזר.

NewUserCreationForm מוסיף שדות נוספים ל-UserCreationForm (יורש ממנו) המוגדר כברירת מחדל, כולל שם פרטי, שם משפחה ודואר אלקטרוני, בעוד EditUserForm מאפשר למשתמש לערוך את פרטי פרופיל המשתמש הקיים שלו מבלי לשנות את הסיסמה שלו.

**קישור לgithub:**

https://github.com/HillaNir/PROJECT-RESTAURANT\_29.4-final.git

**סיסמא למנהל:**

שם משתמש: omri\_i

סיסמא: 12345678ab

**ה-templates:**

ב-Manager app:

* **Add\_categoty.html**: - הוספת קטגוריה חדשה לתפריט.
* **Edit\_categoty.html**: - עריכת קטגוריות קיימות בתפריט.
* **Add\_dish.html**: - הוספת מנה חדשה לתפריט.
* **Edit\_dish.html**: - עריכת מנות קיימות בתפריט.
* **All\_deliveries.html**: - הצגת רשימת כל המשלוחים במערכת.
* **All\_users.html**: - הצגת רשימת כל המשתמשים במערכת.

בOrders app-:

* **all\_categories.html**: - הצגת כל הקטגוריות הקיימות בתפריט.
* **user\_delivery\_confirmation.html**: - אישור ללקוח על המשלוח.
* **delivery\_details.html**: - הלקוח מזין את פרטי המשלוח.
* **index.html**: - הדף הראשי של האתר.
* **login.html**: - דף התחברות למשתמש.
* **order\_history.html**: - הצגת היסטוריית הזמנות קודמות של הלקוח.
* **show\_cart.html**: - הצגת סל הקניות של הלקוח.
* **show\_category.html**: - הצגת כל המנות בתוך קטגוריה מסוימת בתפריט.
* **show\_dish.html**: - הצגת פרטי מנה מסוימת בתפריט.
* **signup.html**: - דף הרשמה למשתמש חדש.
* **update\_user.html**: - עדכון פרטי המשתמש.

templates הנמצאים בתיקייה הראשית של הפרויקט:

Base.html: תבנית שמגדירה לנו את המראה הכללי של כל עמוד, מחולק ל-block title, בעזרתו נוכל להציג בכל עמוד את כותרתו, ו block content בעזרתו נוכל להציג בכל עמוד את התוכן הרלוונטי לנו. מה שאפשר דינמיות ושליטה מכל מקום, והתבנית הכללית תישאר זהה.

Nav.html: עמוד התפריט העליון אשר מצורף ל Base.html, ומוצג תמיד בחלק העליון של כל עמוד.

**routes:**

ב-Manager app:

| **נתיב** | **מתודה** | **הסבר** |
| --- | --- | --- |
| /all\_users/ | GET | רשימת כל המשתמשים במערכת |
| /all\_deliveries/ | GET | רשימת כל המשלוחים במערכת |
| /all\_deliveries/ | POST | אישור משלוח |
| /add\_dish/[int:id](javascript:void(0)) | GET | הצגת טופס הוספת מנה חדשה לקטגוריה ספציפית |
| /edit\_dish/[int:id](javascript:void(0))/ | GET | הצגת טופס עריכת מנה קיימת |
| /delete\_dish/int[:id](javascript:void(0))/ | GET | עמוד מחיקת מנה קיימת |
| /add\_category/ | GET | הצגת טופס הוספת קטגוריה חדשה למערכת |
| /edit\_category/[int:id](javascript:void(0))/ | GET | הצגת טופס עריכת קטגוריה קיימת |
| /delete\_category/[int:id](javascript:void(0))/ | GET | עמוד מחיקת קטגוריה קיימת |
| /add\_dish/[int:id](javascript:void(0)) | POST | ביצוע הוספת מנה חדשה לקטגוריה ספציפית |
| /edit\_dish/[int:id](javascript:void(0))/ | POST | ביצוע עריכת מנה קיימת |
| /delete\_dish/[int:id](javascript:void(0))/ | POST | ביצוע מחיקת מנה קיימת |
| /add\_category/ | POST | ביצוע הוספת קטגוריה חדשה למערכת |
| /edit\_category/[int:id](javascript:void(0))/ | POST | ביצוע עריכת קטגוריה קיימת |
| /delete\_category/[int:id](javascript:void(0))/ | POST | ביצוע מחיקת קטגוריה קיימת |

בOrders app-:

| **נתיב** | **מתודה** | **הסבר** |
| --- | --- | --- |
| '' | GET | הדף הראשי של האתר. |
| 'signup/' | GET, POST | עמוד הרשמה למשתמש חדש.  GET- הצגת הטופס  POST- שליחת הטופס |
| 'login/' | GET, POST | עמוד כניסה למשתמש רשום.  GET- הצגת הטופס  POST- שליחת הטופס |
| 'all\_categories/' | GET | רשימת כל הקטגוריות הקיימות במערכת. |
| 'show\_category/[int:id](javascript:void(0))/' | GET | עמוד של קטגוריה ספציפית, שבו מוצגים כל המנות הקשורות לקטגוריה זו. |
| 'show\_dish/[int:id](javascript:void(0))/' | GET | עמוד של מנה ספציפית, שמציג את הפרטים שלה. |
| 'show\_cart/' | GET | עמוד הסל של הלקוח, עם כל המנות שהוסיפו לו. |
| 'add\_to\_cart/[int:id](javascript:void(0))/' | POST | מתודה שמוסיפה מנה לסל הקניות. |
| 'delete\_from\_cart/[int:id](javascript:void(0))' | POST | מתודה שמוחקת מנה מהסל הקניות. |
| 'change\_amount/[int:id](javascript:void(0))' | POST | מתודה שמשנה את כמות המנות שבסל הקניות. |
| 'create\_delivery/' | GET, POST | עמוד הזמנת משלוח חדשה. |
| 'delivery\_details/' | GET, POST | עמוד הזנת פרטי המשלוח.  GET- הצגת הטופס  POST- שליחת הטופס |
| 'user\_delivery\_confirmation/' | POST | עמוד אישור ללקוח על המשלוח. |
| 'order\_history/' | GET | עמוד היסטוריית הזמנות הלקוח. |
| 'update\_user/[int:id](javascript:void(0))/' | GET, POST | עמוד עדכון פרטי משתמש רשום.  GET- הצגת הטופס  POST- שליחת הטופס |

**models:**

ב-Manager app:

אין.

בOrders app-:

* User - מייצג משתמש, ומכיל מידע כמו שם משתמש וסיסמא.

חלק מהגדרות המובנות של Django ומייצג משתמש במערכת הפרויקט. הוא מוגדר כחלק ממודל ההרחבה AbstractUser של Django, והוא יורש מ-AbstractUser, ומכיל שדות של: שם משתמש, שם-פרטי, שם-משפחה, אימייל, סיסמא, האם המשתמש הוא חבר צוות, האם המשתמש מחובר. המודל מאפשר פעולות כמו יצירת משתמש חדש, התחברות, התנתקות.

לקוח יוצר עגלה- ללקוח אחד יכולות להיות הרבה עגלות- מערכת יחסים של onetomany.

* Cart - מייצג עגלת קניות, ומכיל מידע כמו משתמש המחזיק בעגלה.

בין מודל מנה למודל עגלה יש מערכת יחסים של manytomany אשר יוצרים **ישות חדשה** בשם items. כל מנה יכולה להופיע בהרבה בעגלות, וכל עגלה יכולה להכיל הרבה מנות.

* Category - מייצג קטגוריית מנות, ומכיל מידע כמו שם הקטגוריה ותמונה להצגה.

בין מודל קטגוריה למודל מנה יש מערכת יחסים של onetomany- תחת קטגוריה אחת יכולות להיות הרבה מנות.

* Delivery - מייצג משלוח, ומכיל מידע כמו לאיזה עגלה הוא שייך, האם המשלוח בוצע, כתובת המשלוח, הערות ותאריך היצירה.

בין המודל משלוח לעגלה יש מערכת יחסים של onetoone, לכל עגלה יש משלוח אחד וההפך.

* Dish - מייצג מנה, ומכיל מידע כמו שם המנה, מחיר, תיאור, תמונה להצגה, האם המנה ללא גלוטן והאם המנה צמחונית, וקטגוריית המנה.

בין מודל מנה למודל עגלה יש מערכת יחסים של manytomany אשר יוצרים **ישות חדשה** בשם items. כל מנה יכולה להופיע בהרבה בעגלות, וכל עגלה יכולה להכיל הרבה מנות.

בנוסף, בין מודל קטגוריה למודל מנה יש מערכת יחסים של onetomany- תחת קטגוריה אחת יכולות להיות הרבה מנות.

* Item - מייצג פריט בעגלת קניות, ומכיל מידע כמו המנה המצורפת לפריט, העגלה המצורפת לפריט וכמות המנות שנבחרו.
* בין מודל מנה למודל עגלה יש מערכת יחסים של manytomany אשר יוצרים **ישות חדשה** בשם items- אשר לה יש שדה נוסף של כמות-amount.

תודה.