אפליקציית אנשי קשר:

היום:

1. CRUD עם משתמשים (סשנים)
2. מימוש ב flask ו Django
3. להכניס כמה שיותר best practices שלמדנו
4. שילוב יותר עשיר של javascript לביצוע פעולות נפוצות

בשיעור הבא:

נוסיף את ה API ונהפוך את האפליקציה ל SPA

תהליך עבודה:

* הנתונים – איך נשמור וננהל אותם
  1. טבלאות מתאימות בDB
     1. פלאסק (SQL)
     2. ד'גאנגו (Models)

טבלאות:

Contacts (id, name, email, phone, address, user\_id)

Users (id, username, password)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| address | User\_id | phone | email | name | id |
|  | 1 |  |  | ofir | 1 |
|  | 2 |  |  | nick | 2 |
|  | 1 |  |  | nick | 3 |

|  |  |  |
| --- | --- | --- |
| password | Username | id |
| 1111 | tal@tal.com | 1 |
| 2222 | San@san.com | 2 |

מבני נתונים (העברה לטמפלייט):

1. אובייקטים
2. מילונים
3. tuples

* UI
  1. דף אחד עם טבלה
  2. כל שורה תהיה איש קשר + כפתורים לפעולות CRUD
  3. כפתור create להוספה
* ולידציות (אימות של הקלט) עם JS
  1. אימות של פרטי איש קשר שנוצר עם create
     1. למשל אימייל יכלול @
     2. קידומת של טלפון
     3. לבדוק שכל השדות מלאים (שדות חובה)
     4. כתובת – נניח שהעיר קיימת

תרגיל כיתה:

להוסיף ולידציות ל create:

1. לבדוק שהמספר טלפון מתחיל בקידומות של נייד (052, 054, 053) אחרת כותבים שגיאה: "מספר נייד לא כשר"
   1. "text".slice(0,3)
2. אימייל לבדוק שיש שטרודל אם לא לכתוב שגיאה שמייל לא תקין
3. לבדוק שכל השדות מלאים

תרגיל בית:

לתקן את הלוגיקה של הולידציה כך שמספיק שבדיקה אחת לא עוברת והטופס לא נשלח.