**אפיון פרויקט – מערכת להעלאת מודעות לחיפוש שותפים לדירה.**

**עמית מוזס**

[**https://dirot-shutafim.onrender.com**](https://dirot-shutafim.onrender.com/) **,** [**https://github.com/amit-moses/dirut\_shutafim**](https://github.com/amit-moses/dirut_shutafim)

|  |  |
| --- | --- |
| **מודל Apartment**  תהיה דירה בודדת בה יאוחסן כל הפרטים שהמפרסם הכניס על הדירה | |
| פעולה שתחזיר כתובת url של המודעה. | get\_url |
| פעולה שתחזיר את כותרת המודעה עד 22 תווים ולאחר מכן תוסיף ... לצורך תצוגה בדף החיפוש | short\_title |
| פעולה שתחזיר את תוכן המודעה עד 25 תווים ולאחר מכן תוסיף ... לצורך תצוגה בדף החיפוש | short\_details |
| פעולה שתחזיר את תאריך הכניסה לדירה בפורמט dd/mm/yyyy לצורך תצוגה | get\_date\_format |
| פעולה שתחזיר את כל הערכים לש האובייקט כJSON | toJSON |
| פעולה שתקבל תמונה מסוג imagedata ותמחק אותה ובנוסך תמחק אותה גם באחסון התמונות בfirebase | delete\_1\_image |
| פעולה שמקבלת רשימה של כתובות url ומוחקת את כל התמונות מלבדן (תמונות לשימור) | updateImages\_1 |
| פעולה שמחזירה מחרוזת עם 6 תווים רנדומלים לצורך קביעת שם הקובץ שיעלה לfirebase ויהיה חלק מכתובת הurl של התמונה | make\_random\_name |
| פעולה שמקבלת רשימה של אובייקטים מסוג תמונה שנקלטה על ידי המשתמש ותעלה את התמונות החדשות | uploadImages |
| פעולה שתקבל תמונות לשימור ותמונות חדשות, ותפעיל את הפעולה updateImages\_1ולאחר מכן updateImages על התמונות החדשות. | updateImages |
| פעולה שתמחק את כל התמונות של הדירה גם בענן של firebase | deleteAllImages |

|  |  |
| --- | --- |
| **המודל ImageData**  תכיל את כתובת הurl של התמונה מהfirebase ובנוסף תשמור את המזהה של הדירה | |
| פעולה שתחזיר את המספר שמציג את הסדר בו התמונה הועלתה אם הועלתה ראשונה – יתקבל 0. אם שניה יתקבל 1 .... | index |
| פעולה שתחזיר את כתובת הurl של התמונה | \_\_str\_\_ |

Templates

1. **addmanage.html** – מסך העריכה של מודעה, ממנו ניתן לעלות מודעה או לערוך מודעה, הכנסת פרטים על המודעה ובנוסף קליטה של תמונות ועריכה שלהן אם המודעה קיימת.
2. **index.html** – מסך חיפוש הדירות – בו יופיעו 12 דירות ולאחר גלילת הדף עד סוף המודעות שהוצגו יוצגו עוד 12 לצורך יעילות של העלאת מודעות למקרה שיהיה הרבה מודעות. יופיע כפתור חיפוש שיפתח פופ-אפ וממנו יקלטו מהמשתמש ערכי פרטי דירות שאותתן ירצה לחפש. בנוסף יופיע כפתור מיין שבו המשתמש יוכל לבחור את הסדר בו יופיעו המודעות בהתאם לבחירה, וכפתור סדר עולה/יורד.
3. **login.html** – עמוד הכניסה למשתמשים מפרסמי דירות, יופיע לשונית התחברות (פשוטה) והרשמה **–** בה המשתמש יכניס את כתובת המייל שלו וישלח לו מייל לאישור כתובת המייל, רק לאחר לחיצה על הקישור החשבון יהפוך לפעיל. אם ינסה להתחבר ללא לחיצה על הקישור יראה הודעת שגיאה.בנוסף יופי כפתור שכחת סיסמא – מותאם מהadmin
4. **myads.html** – מסך בו המשתמש ייראה את המודעות שפרסם עם כפתורי עריכה ומחיקה.
5. **singlepage.html** – מסך בו כל הגולשים באתר יוכלו להגיע מהעמוד הראשי של חיפוש הדירה. במסך זה יופיע כל הפרטים אותם מפרסם המודעה הכניס ובנוסף אם המפרסם בחר שיוכלו לשלחו לו מייל ישירות מהאתר יהיה כפתור צור קשר בו יפתח פופ-אפ של יצירת קשר עם פרטים, תוכן המודעה וכותרת וכפתור אימות RECAPacha של גוגל על מנת למנוע שליחת ספאם למפרסם. ההודעות יגיעו ישירות למייל של מפרסם הדירה באמצעות מערכת SMTP.

|  |  |
| --- | --- |
| **פירוט views** | |
| מחלקה שבונה את כפתור ה- RECAPacha בטופס יצירת קשר | Recaptcha |
| פעולה שמקבלת מייל, נושא הודעה, ותוכן הודעה ושולחת למייל זה מייל בהתאים למה שנקלט באמצעות SMTP | send\_email |
| פעולה שתשלח מייל אימות כתובת מייל לאחר הרשמה – תשלח קישור שלאחר שילחצו עליו החשבון יהפוך לפעיל | send\_vaildation\_email |
| תקלוט את כל נתוני החיפוש של מחפש הדירה ותשלח אותם לtemplate בדף הראשי לא ייקלטו נתונים לכן זה גם יהיה הדף הראשי ללא נתוני חיפוש:  <https://dirot-shutafim.onrender.com/> | index |
| פעולה שתקבל נתונים לאחר בקשת POST ובנוסף תקבל id אם הid קיים תבדוק אם המשתמש ששלח את הבקשה הוא המפרסם של הדירה ואם כן תעדכן את הנתונים אותם הכניס. אם לא נקלט id זהו מודעה חדשה. | api  @login\_required |
| פעולה שתקבל id של דירה, אם קיימת והנוסף המשתמש ששלח את הבקשה הוא המפרסם של הדירה. הפעולה תמחק את הדירה אם כל התמונות | delete\_apr  @login\_required |
| פעולה שתציג את מסך עריכת הדירה או הוספת דירה חדשה, תקלוט id אם קיים והמשתמש הוא המפרסם תציג את מסך העריכה, אם לא קיים id ייפתח טופס יצירת מודעה חדשה | newadd  @login\_required |
| פעולה שמחזירה את העמוד בו מופיעות כל המודעות אותם פרסם המשתמש בנוסף לכפתור עריכה ומחיקה | myads  @login\_required |
| פעולה שמקבלת בקשת GET ובו כל נתוני החיפוש אותם הכניס מחפש הדירה, הפעולה יוצרת query של דירות בהתאם לנותני החיפוש ובנוסף את הספרה שמייצגת את המיון של אותו הסדר שבחר המשתמש בו יופיע. הפעולה תשלח רק את 12 המודעות הראשונות כאובייקט מסוג JSON ותבחר את הדירה האחרונה שתוצג כדירה שלא תופיע ב12 הבאים. | search |
| פעולה שתקבל את כל ערכי הנתונים אותם הכניס הנרשם ואם אין מייל ומשתמש כאלה קיימים תיצור משתמש חדש וtokenc – string עם 12 תווים רנדומלים שיופיעו בקישור שישלח לאימות כתובת המייל באמצעות הפעולה send\_vaildation\_email | register\_page |
| פעולה שתקלוט שם משתמש וסיסמא ותבצע התחברות. | login\_page |
| פעולה שתבצע התנתקות מהמערכת | logout\_page |
| פעולה שתקלוט בקשת POST ממחפש הדירה ואם מילא את כל הפרטים ומילא את אימות RECAPacha תשלח מייל למפרסם עם הנתונים שנקלטו באמצעות SMTP והפעולה send\_email | send\_email\_to\_publisher |
| פעולה שתקלוט את ה-token מהמייל לאימות כתובת המייל שנשלח בעת ההרשמה ותבדוק אם זה token שרשום, אם כן תפעיל את החשבון. אם היוזר התחבר כבר או שנרשם הרגע – לא יצטרך להתחבר מחדש אם הגיע לשם ממחשב/דפדפן אחר יצטרך להתחבר שוב. | vaild\_account |
| פעולה שתחזיר דף 404 מותאם לעיצוב האתר | custom\_404 |

שמירת התמונות בfirebase: בהתחלה יפיע בשם הקובץ מזהה המודעה ולאחר מכן 6 תווים רנדומלים שנוצרו בעת העלאת התמונה

תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי

Dirot-shutafim

Models:

1.  User (default Django)

2.  Apartment

•   id

•   Publisher (fk)

•   city

•   street

•   rent\_price

•   floor

•   partners

•   gender

•   entry\_date

•   details

•   title

•   kosher

•   agree\_mail

•   type

-   get\_url – return the url of singlepage view of apartment.

-   short\_title / short\_ details – return short data for preview page.

-   get\_date\_format – return the entrey date as 'dd/mm/yyyy' format.

-   toJSON – return json object of this apartment.

-   delete\_1\_image – delete the image which entered.

-   updateImages\_1 – delete every image which deleted by the user.

-   make\_random\_name – select random key for file name in firebase

-   uploadImages – upload every image in the images list to imagedata model and upload to firebase

-   updateImages (updateImages\_1 + uploadImages)

-   deleteAllImages – deletes all the images of this apartment from imagedata model and from firebase

3.  ImageData

•   Id

•   apartment (fk)

•   myurl

-   index – get the index of this image according to the order of uploading.