

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

### עמית מוזס

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

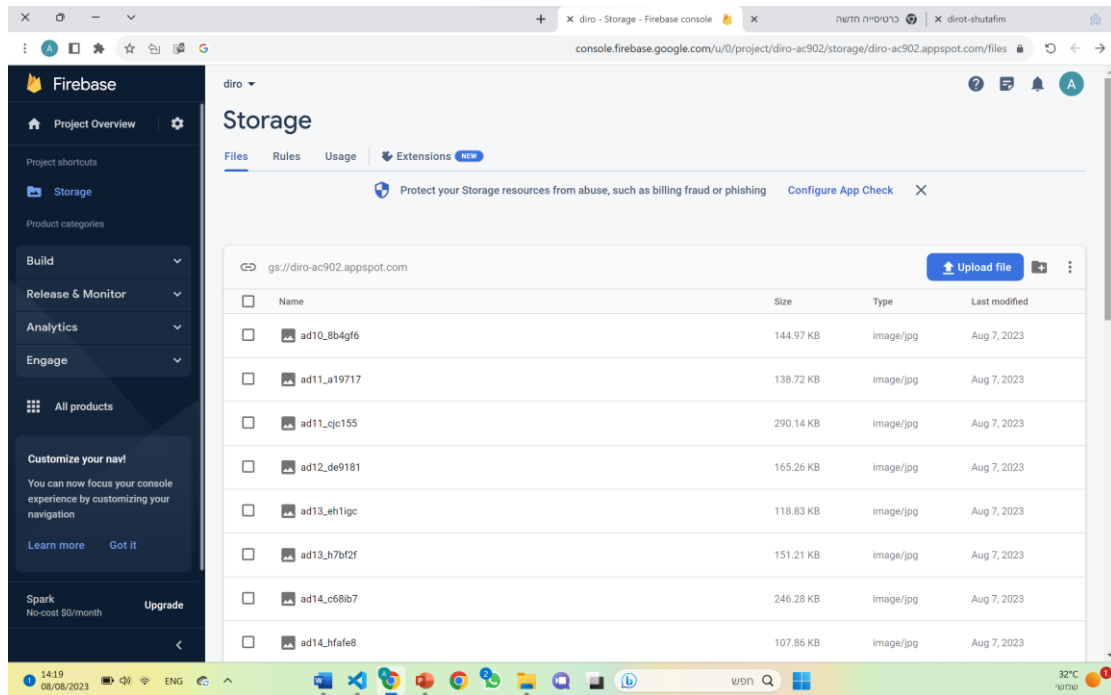
מודל Apartment	
תהיה דירה בודדת בה יאוחסן כל הפרטים שהמפרסם הכניס על הדירה	
get_url	פעולה שתחזיר כתובת url של המודעה.
short_title	פעולה שתחזיר את כותרת המודעה עד 22 תווים ולאחר מכן תוסיף ... לצורך תצוגה בדף החיפוש
short_details	פעולה שתחזיר את תוכן המודעה עד 25 תווים ולאחר מכן תוסיף ... לצורך תצוגה בדף החיפוש
get_date_format	פעולה שתחזיר את תאריך הכניסה לדירה בפורמט dd/mm/yyyy לצורך תצוגה
toJSON	פעולה שתחזיר את כל הערכים לש האובייקט כJSON
delete_1_image	פעולה שתקבל תמונה מסוג imagedata ותמחק אותה ובנוסף תמחק אותה גם באחסון התמונות בfirebase
updateImages_1	פעולה שמקבלת רשימה של כתובות url ומוחקת את כל התמונות מלבדן (תמונות לשימור)
make_random_name	פעולה שמחזירה מחרוזת עם 6 תווים רנדומלים לצורך קביעת שם הקובץ שיעלה לfirebase ויהיה חלק מכתובת url של התמונה
uploadImages	פעולה שמקבלת רשימה של אובייקטים מסוג תמונה שנקלטה על ידי המשתמש ותעלה את התמונות החדשות
updateImages	פעולה שתקבל תמונות לשימור ותמונות חדשות, ותפעיל את הפעולה updateImages_1 ולאחר מכן updateImages על התמונות החדשות.
deleteAllImages	פעולה שתמחק את כל התמונות של הדירה גם בענן של firebase
המודל ImageData	
תכיל את כתובת url של התמונה מהfirebase ובנוסף תשמור את המזהה של הדירה	
index	פעולה שתחזיר את המספר שמציג את הסדר בו התמונה הועלתה אם הועלתה ראשונה – יתקבל 0. אם שניה יתקבל 1 ....
__str__	פעולה שתחזיר את כתובת url של התמונה

### 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 ערכי פרטי דירות שאותן ירצה לחפש. בנוסף יופיע כפתור מיין שבו המשתמש יוכל לבחור את הסדר בו יופיעו המודעות בהתאם לבחירה, וכפתור סדר עולה/יורד.
3. **search\_by\_map.html** – מסך חיפוש דירות על גבי מפה באמצעות Bing maps.
4. **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
5. **myads.html** – מסך בו המשתמש יראה את המודעות שפרסם עם כפתורי עריכה ומחיקה.
6. **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	
Recaptcha	מחלקה שבונה את כפתור ה-RECAPcha בטופס יצירת קשר
send_email	פעולה שמקבלת מייל, נושא הודעה, ותוכן הודעה ושולחת למייל זה מייל בהתאים למה שנקלט באמצעות SMTP
send_vaildation_email	פעולה שתשלח מייל אימות כתובת מייל לאחר הרשמה – תשלח קישור שלאחר שילחצו עליו החשבון יהפוך לפעיל
index	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: <a href="https://dirot-shutafim.onrender.com">/https://dirot-shutafim.onrender.com</a>
api @login_required	פעולה שתקבל נתונים לאחר בקשת POST ובנוסף תקבל id אם id 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.
delete_apr @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
newadd @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 יפתח טופס יצירת מודעה חדשה
myads @login_required	פעולה שמחזירה את העמוד בו מופיעות כל המודעות אותם פרסם המשתמש בנוסף לכפתור עריכה ומחיקה
search	פעולה שמקבלת בקשת GET ובו כל נתוני החיפוש אותם הכניס מחפש הדירה, הפעולה יוצרת query של דירות בהתאם לנתוני 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.
register_page	פעולה שתקבל את כל ערכי הנתונים אותם הכניס הנרשם ואם אין מייל ומשתמש כאלה קיימים תיצור משתמש חדש ו token – string עם 12 תווים רנדומלים שיופיעו בקישור שישלח לאימות כתובת המייל באמצעות הפעולה send_vaildation_email
login_page	פעולה שתקלוט שם משתמש וסיסמא ותבצע התחברות.
logout_page	פעולה שתבצע התנתקות מהמערכת
send_email_to_publisher	פעולה שתקלוט בקשת POST ממחפש הדירה ואם מילא את כל הפרטים ומילא את אימות RECAPcha תשלח מייל למפרסם עם הנתונים שנקלטו באמצעות SMTP והפעולה send_email
vaild_account	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.
custom_404	פעולה שתחזיר דף 404 מותאם לעיצוב האתר
map	פעולה שמחזירה את דף חיפוש הדירות על גבי מפה כמו ב-index אך תצוגה על גבי מפה. אותם ערכי חיפוש.

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



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 entry 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.