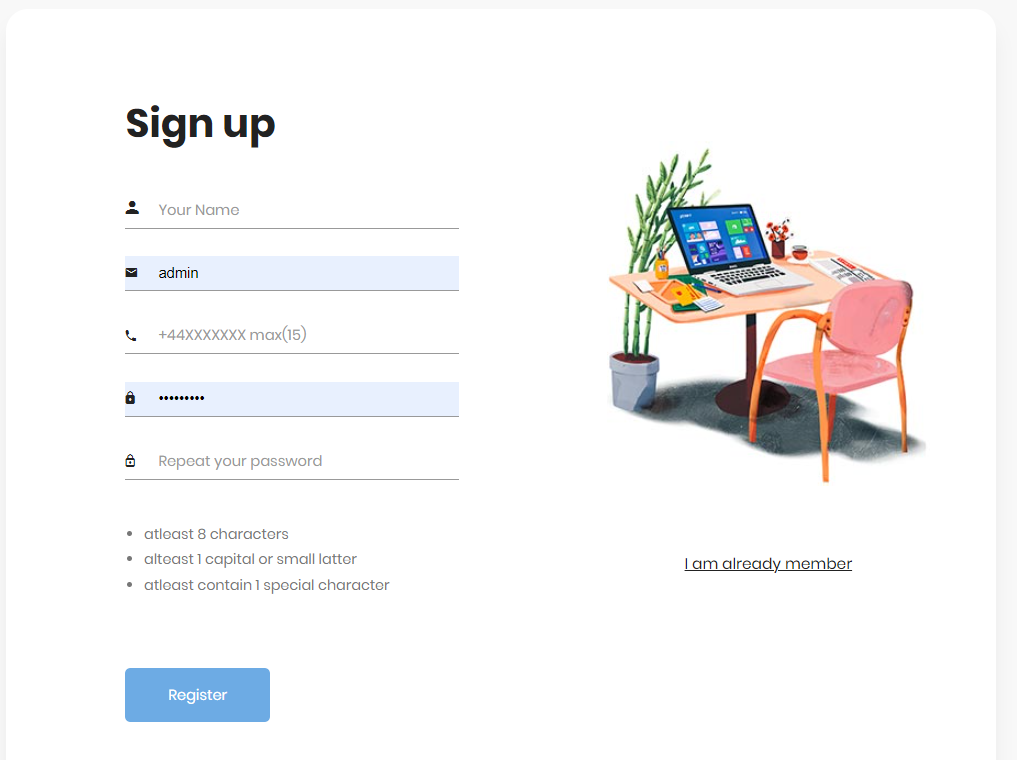
Application URL: <https://online-lovejoy.herokuapp.com/>

OneDrive link for code: <here>

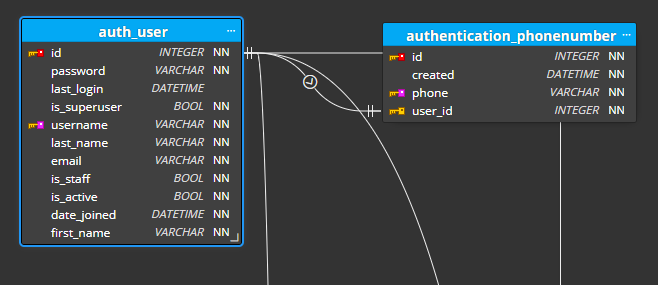
## Task 0 – Self-reflection

Task 1 – User registration

The user registration page look like this



It takes the username, email, password and telephone to store in the database for user registration as shown in the below figure



The above registration function is significantly secured due to the following reasons:

* Use the CSRF token to handle the request. CSRF token is token that generated by the server for each page and ensure that each request is coming on the server having a CSRF token. The use of CSRF token ensure from the malicious traffic on the server.
* Use the distinct validations to ensure that the data in the from is according to the format. Firstly, validate that the username is not already reserved.
* Validated that the username has only contain the alpha numeric characters with maximum length of 10.
* Validate that email is not fetched with another account.
* Validated that the phone number is according to the format of UK.
* Match the password and retyped password. Use the regex operator to confirm that the password must contain at least one capital, one small one digit and one special character with the length of 8 digits minimum.
* All these validations were validated on the server side. If the browser is not supported the JavaScript or user intentionally stop the script on the server, the registration will secure in that condition also.
* Store the received values on the server were store in DB table with built-in Django feature that ensure the prevention from SQL injection.
* Lastly, after storing the values, the registered user will still inactive, and the activation link is sent to user provided email. Through that email, user will be able to activate his account. The code snippet of all the validations is shown in below figure.

Text

Description automatically generated

## Task 2 - Develop a secure login feature.

Graphical user interface, application, Teams

Description automatically generated

The above registration function is significantly secured due to the following reasons:

* Use the CSRF token to handle the request. CSRF token is token that generated by the server for each page and ensure that each request is coming on the server having a CSRF token. The use of CSRF token ensure from the malicious traffic on the server.
* Like the registration function, login function also use the Django built-in function to authenticate the user and also prevent from the SQL injection.

Text

Description automatically generated

## Task 3 - Implement password strength and password recovery

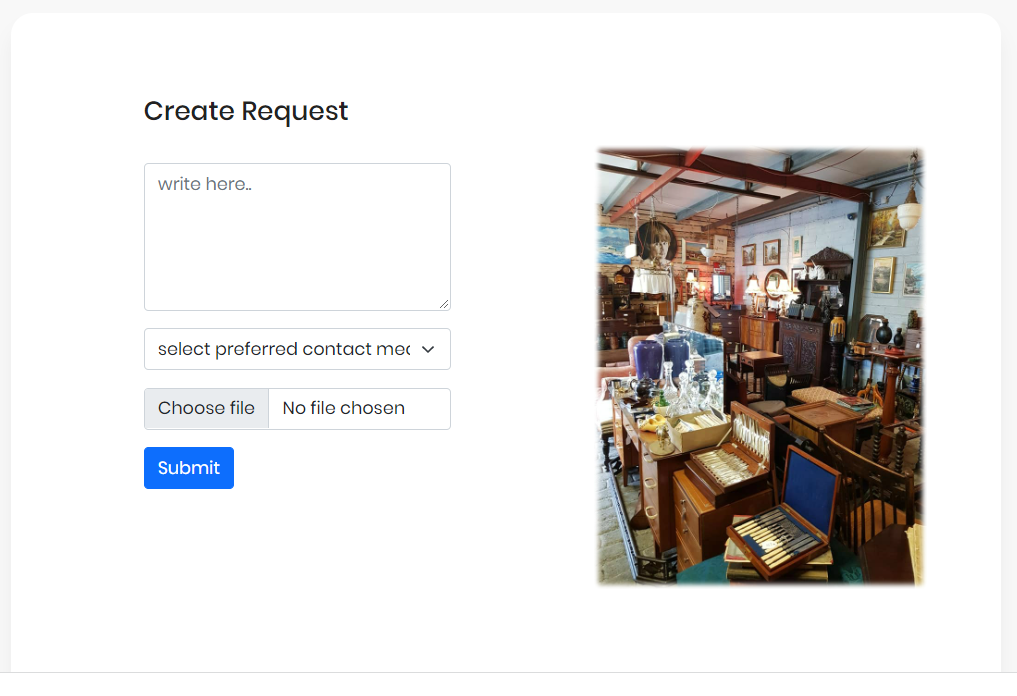
Text

Description automatically generated

The password policy was strengthened by the following techniques:

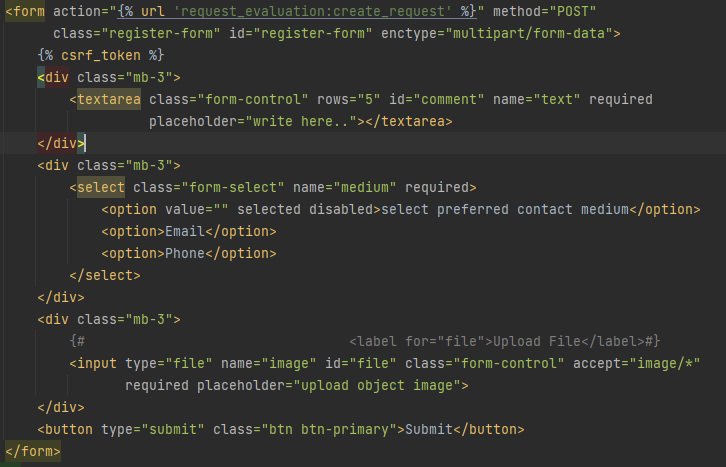
* Firstly, the password and retyped password were matched and ensured that both passwords are similar.
* Validate that the password is at least 8 digits long.
* Password must contain at least one capital letter.
* Password must contain at least one small letter.
* Password must contain at least one digit.
* Password must contain ta least one special character.
* All of these validation rules were implemented using the regex operation.

## Task 4 - Implement a “Evaluation Request” web page.

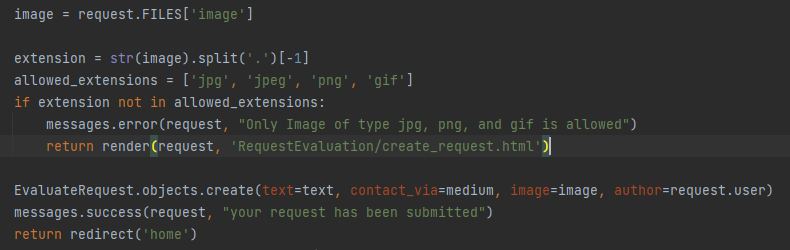


The following features were implemented in evaluation request function to ensure the security of the module:

* The evaluation request module also used the CSRF token in the html form that validated by the server to prevent from malicious traffic.
* Ensure that dropdown only contain the phone, and email options.



## Task 5 - Develop a feature that will allow customers to submit photographs

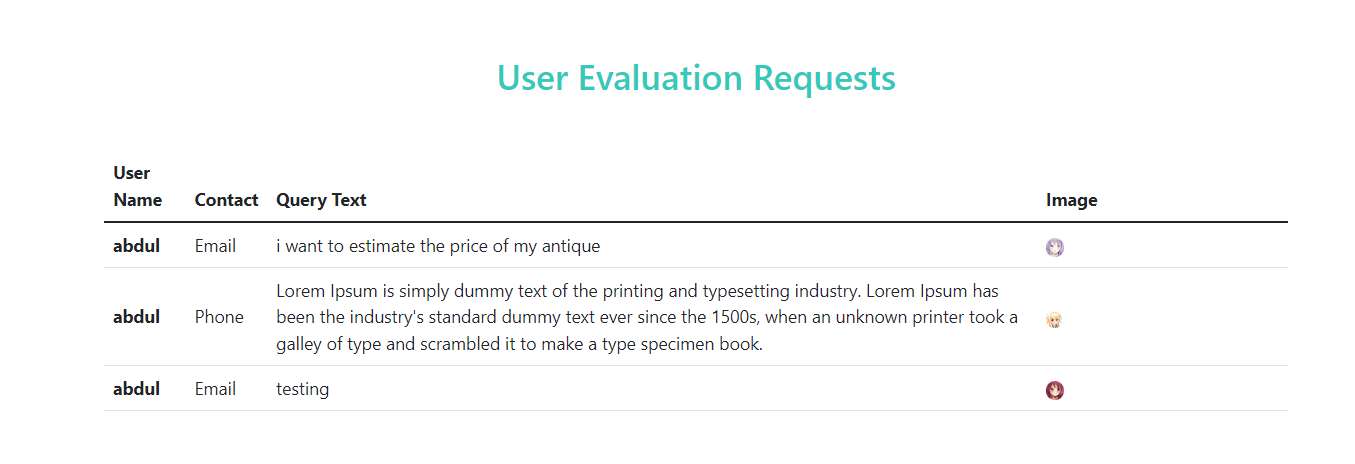


To ensure the security of this features, the following counter measures were completed:

* Validate that the uploading file contain at least one object
* Secondly, validate that uploaded file must have the image of type PNG, jpg, jpeg, or gif.
* Store the address in DB table and image in static file repository.
* Django built-in file manager store the uploaded file to prevent from the same name redundancy issue.

## Task 6 – Request Listing Page

The below shows the list of all evaluation requests that are only accessible with admin role.



To ensure that the request list is only accessible for admin user, the following steps were taken:

* Firstly, on the server side, it is validated that the user is authenticated and belong to the group of admin/superuser.
* The evaluation request list was fetched from DB after the user authentication. Otherwise, blank page with only navbar was return to the normal user.
* Although, the server authentication policy of validating admin user is very secure, but the admin role was again validated before displaying data on the client side as show in second image.

Text

Description automatically generated

Text

Description automatically generated