Timebox1 – admin bruger

Oversigt

OpgaveNavn	Admin bruger		
Implementering af krav	WEB-2, WEB-11		
Udført af	Jan	**Dato**	24-09-2021
Timebox	1	**Område**	website

Contents

Introduktion

Denne rapport vil gå igennem hvordan admin brugerens side vil indeholde, hvordan det oprettes, samt funktionaliteterne dertil.

Analyse

For at ændringerne beskrevet i dette dokument kan fungere, skal ændringer til modellerne, managers og views fra deliverablen TB1-login være lavet først.

For at have en admin bruger, der vil kunne tilgå databaseelementerne, vil der skulle laves nogle ændringer dertil, samt skal der laves nogle custom ændringer til Djangos standard admin bruger.

For at kunne lave disse ændringer, er der diverse filer der skal ændres, samt nye filer der skal kreeres:

- *Forms.py*
    - Forms filen fortæller hvilke elementer der skal passere imellem databasen, og HTML siden for admins
- *Admin.py*
    - For at kunne ændre i Djangos standard template for admins, vil der kunne laves en *admin.py* fil, hvor de diverse ændringer kan defineres i.

Design

Der vil blive brugt Djangos standard design for admin siden.

Når Django genererer siden for admin's, vil den starte ud med at lede efter en admin.py fil og en forms.py side, for så at overskrive standardtemplaten, med ændringerne defineret i disse 2 filer.

Implementering

Forms.py:
I *forms.py* er der blevet defineret hvilke "fields" fra "users" modellen, der skal kunne passeres videre, til admin siden, når der ses / redigeres / oprettes nye brugere igennem admin siden, som kan ses på Figur 1.

```
class CustomUserCreationForm(UserCreationForm):
    class Meta:
        model = Users
        fields = ('email', 'name', 'phoneNumber', 'company', 'userType')


class CustomUserChangeForm(UserChangeForm):
    class Meta:
        model = Users
        fields = ('email', 'name', 'phoneNumber', 'company', 'userType')
```

Figur 1 forms.py

Admin.py:
For at kunne lave ledet mellem modellerne og admin siden, skal der fortælles hvilken form der skal bruges, i stedet for standardformen I Django, og hvilken model der skal bruges, som kan ses på Figur 2.

```
"""
Make the admin use the custom forms, and define admin site
"""
add_form = CustomUserCreationForm
form = CustomUserChangeForm
model = Users
```

Figur 2 forms import

Der skal defineres hvad der kan ses I listen af brugere, og hvilke felter der kan filtreres imellem, som kan ses på Figur 3.

```
# What is shown in the list of users
list_display = ('name', 'company', 'email', 'userType')

# What filters can be applied to the entries
list_filter = ('userType',)
```

Figur 3 felter + filter

Der skal defineres hvad der kan ses, når en bruger bliver redigeret, som vist på Figur 4.

```
# What is shown when editing user
fieldsets = (
    ('User info',
     {'fields': (
        'email',
        'name',
        'phoneNumber',
        'company',
        'userType',
        'password'
     )}),
    ('Permissions',
     {'fields': (
        'is_superuser',
        'is_staff',
     )}),
)
```

Figur 4 edit user

Der skal defineres hvad der kan ses, når en burger bliver oprettet, som vist på Figur 5.

```
# what is shown in create user
add_fieldsets = (
    (None,
     {
        'classes': ('wide',),
        'fields': (
            'email',
            'name',
            'phoneNumber',
            'company',
            'userType',
            'password', 'password2',
            'is_superuser',
            'is_staff',
        )}),
)
```

Figur 5 create user

Der skal defineres hvilke felter de kan søges på, I søge feltet, og hvilken rækkefølge elementerne bliver listet som standard, som vist på Figur 6.

```
# What field can be searched by
search_fields = ('email', 'name', 'company', 'phoneNumber')

# What are the fields ordered by as standard
ordering = ('name',)
```

Figur 6 search + order

Der skal defineres hvilke databasetabeller der kan tilgås igennem admin siden, samt skal der sættes hvilke standard udseende elementer, der skal overskrives på users siden, og hvilke databaseelementer der skal fjernes, som kan ses på Figur 7.

```
"""
Makes it possible to see the different tables on the admin site
"""
# The "users" section, applies a custom class
admin.site.register(Users, CustomUserAdmin)

# These use Djangos standard templates
admin.site.register(Logs)
admin.site.register(Facilities)
admin.site.register(JoinTableUser)
admin.site.register(JoinTableFacility)

# Remove group field on admin site
admin.site.unregister(Group)
```

Figur 7 register / unregister

Verifikation

Der skal testes at de forskellige ændringer, nævnt i implementeringsafsnittet, er taget til på admin siden.

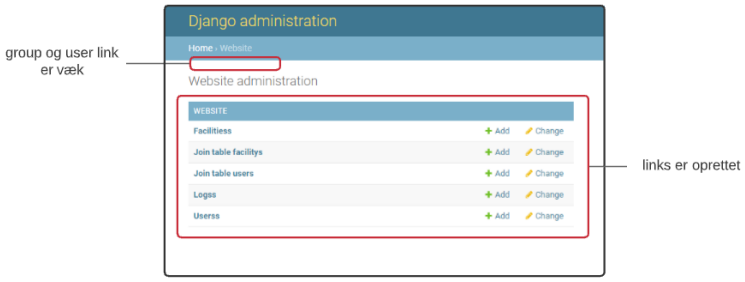
- Tabeller der kan ses på admin siden er ændret
- Udseendet for users tabel siden er ændret
- Add users siden er ændret
- See/edit users siden er ændret

Tabel 1: Tests til verifikation af opgave

Test	Test Steps	Pre-requisites	Pass-betingelser	Resultat
Admin tabeller	N/A	• Serveren er startet op. • Der er logget ind som admin.	1. group link er væk. 2. user link er væk. 3. users link er oprettet. 4. facilites link er oprettet. 5. logs link er oprettet. 6. join table users link er oprettet. 7. join table facility link er oprettet.	Bestået
Users tabel side	1. der trykkes på users linket	• Serveren er startet op. • Der er logget ind som admin.	• Siden passer overens med det definerede i *admin.py* filen, beskrevet i implementations afsnittet	Bestået
Add user	1. der trykkes på users linket 2. der trykkes på add user linket	• Serveren er startet op. • Der er logget ind som admin.	Siden passer overens med det definerede i *admin.py* filen, beskrevet i implementations afsnittet	Bestået
Edit/see user	1. der trykkes på users linket 2. der trykkes på en bruger	• Serveren er startet op. • Der er logget ind som admin.	Siden passer overens med det definerede i *admin.py* filen, beskrevet i implementations afsnittet	Bestået

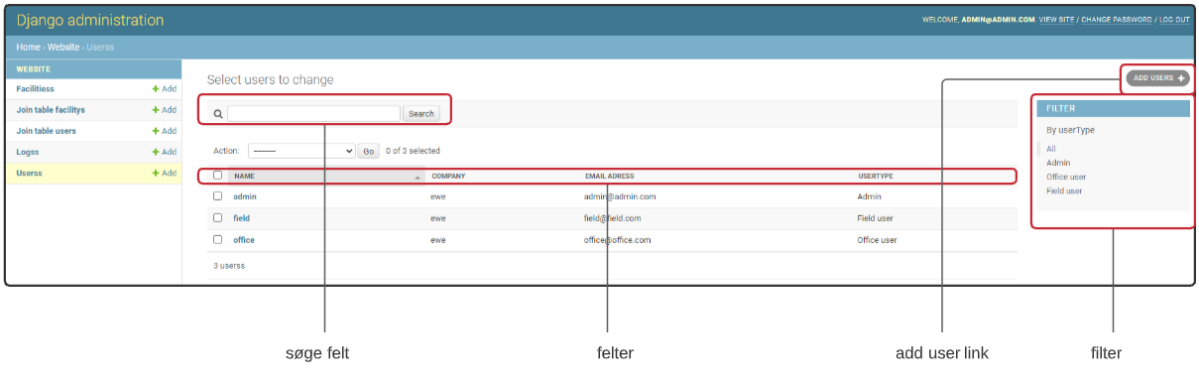
Testresultat

Resultat for ”admin tabeller” test, kan ses på Figur 8.



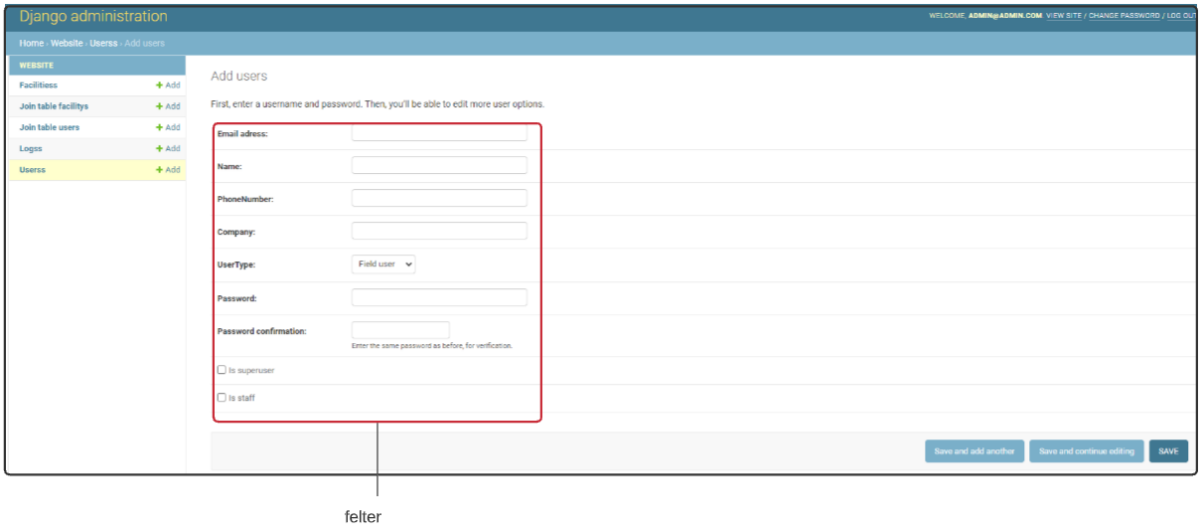
Figur 8 admin homepage

Resultatet for ”users tabel side” test, kan ses på Figur 9.



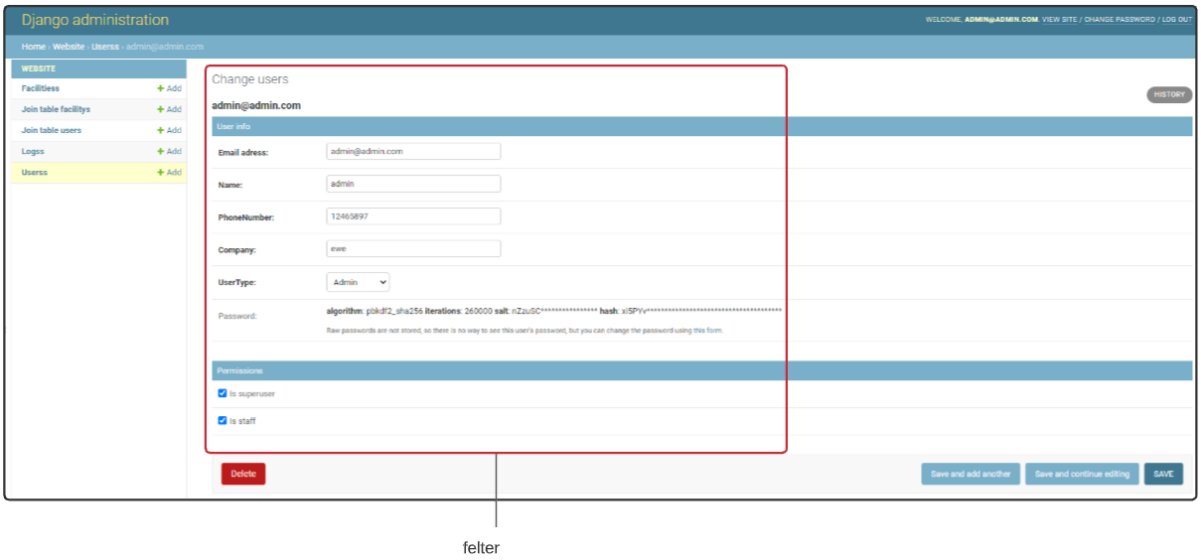
Figur 9 users page

Resultat for ”add user” test, kan ses på Figur 10.



Figur 10 add user

Resultat for “Edit/see user” test, kan ses på Figur 11.



Figur 11 edit/see user

Konklusion

Udfærdigelse for admin side, er blevet gjort som der blevet regnet med, ved at overskrive Djangos standard template for Djangos admin side.