Proiect Ia Ingineria Software

-CV Web App-

Studenta: Vasilache Maria Profesor coordinator:

Grupa: 30238 Luminita Marghescu

-2022/2023-

1.Despre Aplicatie:

Aplicatia este destinate utilizatorilor care isi doresc sa isi genereze un CV pentru urmatorul loc de munca.

E o aplicatie interactive si atractiva, punand in valoare noile tendinte in tehnologie.

Accentul este pus pe inovare, de aceea se foloseste pt a atreage mai multi utilizatori, cod QR pentru navigare usoate intre pagini, butoane cu denumiri intuitive.

Feedback-ul este important, de acee avem un formular, pentru utilizator, in care poate adauga atasamente

2.Functionalitate

a. Pagina pentru a crea un nou CV

b. Vizualizarea CV-ului create in format pdf

c. Scanare cod QR si redirectarea catre Europass.ro

d. Trimitere mail cu feedback si atasament

3. DESIGN PATTERNS: Observer

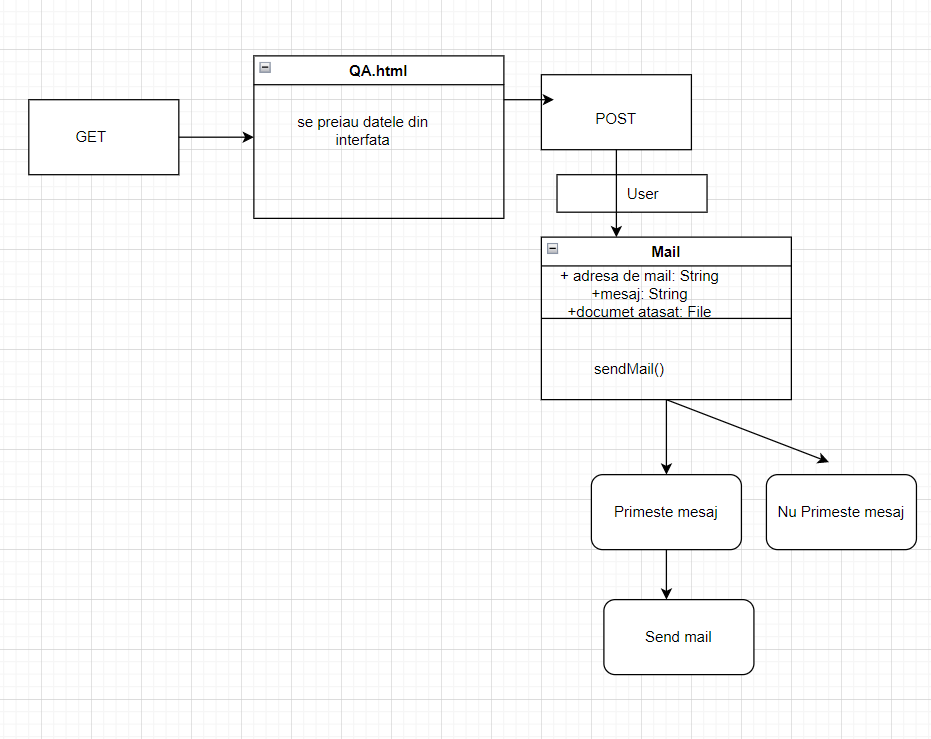
Observatorul este un model de design comportamental care vă permite să definiți un mecanism de abonament pentru a notifica mai multe obiecte despre orice evenimente care se întâmplă cu obiectul pe care îl observă.

In proiectul meu, am folosit acest pattern petru a trimite notificari atunci cand se trimite un email.

Astfel, la schimbarea clasei Email, prin modificarea atributului trimis, se va trimite o notificare paginii principale pentru a anunta trimiterea cu success a mail-ului.

Atunci cand se detecteaza mesajul de trimis, atunci se va trimite o notificare, si toate instantele vor trimite mai departe continutul, generand mesajul propriu-zis.

Implementarea propriu-zisa se gaseste in **observer.py,**  iar apelarea acesteia are loc in metpda qaa, din **qaForm.py**.



Diagram

Description automatically generated

4. Implementare

1. Clasa **CV**

- are implementata metoda SendMail

--foloseste clasa EmailForm

2. Fisierul **Files** – contine toate fisierele pe care le vom updata, noi

3. **Pages**- clasele necesare pentru a prelua date din html(tamplates)

🡪data.py = avem datele pe care le vom pune in cv(pentru CV-ul Demo)

🡪forms.py = preia datele din formularul de la QA

🡪urls.py = redirectare spre pagini, cu functii pentru post

🡪qrPage = clasa pentru redirectare caree pagina cu Europass, se genereaza codul QR

🡪 qaForm.py = clasa care prei date din pagina QA Form si trimite mesaj. In aceasta metoda preluandu-se si atasamentul care va fi trimis prin emai.

🡪 getDetaliinewCV.py: clasa in care se vor prelua toate datele primate de la utilizator😊

🡪 newCV.py: redirectarea catre pagin in care evem template-ul pentru CV

🡪 downloadCV.py: se va descarca cv-ul model

🡪pdf.py: genereaza un pdf, in urma preluarii datelor din interfata.

4. Folder-ul **templates,** toate paginile html pe vare le vom folosi:

Graphical user interface, application

Description automatically generated

5, Folder-ul Files/page/static = contine css, js si imaginile pe care le vom pune in cv

6. Folder comfig-> aici avem toate link-urile, in **urls.py**

5. Diagrama Use-Case

Diagram

Description automatically generated

6. Functionare

1. Pagina de start:

Se porneste aplicatia din linia de comanda cu urmatoarele comenzi:

🡪python -m pipenv shell

🡪python manage.py runserver

🡪In browser: <http://127.0.0.1:8000/>

Actiuni pagina start:

Graphical user interface

Description automatically generated

1. Numele utilizatorului conectat
2. Buton de Log OUT
3. Butoane: NewCV(**creez propriul CV**), MyCV(**un exemplu de CV**), Europass QR(**cv-ul Europas, accesay cu codul QR)**, Download( **descarcarea CV-ului)**
4. Butoane pentru miscarea imaginilor de prezentare
5. NewCV

Graphical user interface, text, application, email

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated



1. Butonul pentru revenire la pagina de start
2. Campuri de completat -Field
3. Attachment
4. Drop Down
5. Selectare data( dintr-un calendar)
6. Buton pentru a trimite formularul

2.1 Dupa apasarea butonului Submit

Graphical user interface

Description automatically generated

1. Buton pentru revenire la pagina principala
2. Redirectionare catre paginile sociale
3. Redirectionarea catre GitHub- proiect personal
4. Generare de pdf, cu datele preluate din formular

2.2 Create PDF

Dupa apasarea butonului de Create PDF

Graphical user interface, text, application

Description automatically generated

1. NewCV

Graphical user interface, application

Description automatically generated(1) Revenire la pagina de start

(2) Redirectionarea la pagina de Git,

cu codul sursa

1. Europass QR

Graphical user interface

Description automatically generated

1. Buton reveire la pagina de start
2. Scanare cod QR
3. Download Button:

A picture containing text, computer, screenshot, desk

Description automatically generated

Cand se apasa butonul, se descarca CV-ul

1. Observatii:

Pornire aplicatie:

D:\An3\IS\P1>python -m pipenv shell

(P1-qv8LNru9) D:\An3\IS\P1>python manage.py runserver

Preluare fisiere media:

\pages\static\img – de aici se va prelua poza

\Files – de aici se va prelua cv-ul

Conturi:

user: [dranga@yahoo.com](http://127.0.0.1:8000/admin/account/emailaddress/2/change/) ->Maria.12\*ab

admin: [vasilachemaria20@yahoo.com](mailto:vasilachemaria20@yahoo.com) ->12

Bibliografie

1. <https://getbootstrap.com/docs/5.0/forms/overview/>
2. <https://learndjango.com/tutorials/django-email-contact-form>
3. <https://www.geeksforgeeks.org/generate-qr-code-using-qrcode-in-python/>
4. <https://stackoverflow.com/questions/20033712/html-img-src-wont-load-my-images>
5. <https://stackoverflow.com/questions/70669746/python-how-to-call-a-function-with-django>
6. <https://www.bing.com/search?FORM=ALBN01&PC=ATAL&PTAG=ATAL00000028&q=create%20a%20pdf%20from%20html%20python>
7. <https://pythonguides.com/get-data-from-get-request-in-django/>
8. <https://www.youtube.com/watch?v=AQrsjt4yyrw&ab_channel=GreatAdib>
9. <https://www.youtube.com/watch?v=wzZiONbtwiA&ab_channel=MaxGoodridge>
10. <https://pythonguides.com/get-data-from-get-request-in-django/>
11. <https://pythonguides.com/django-get-all-data-from-post-request/>
12. <https://pythonguides.com/create-model-in-django/>
13. <https://data-flair.training/blogs/django-database/>
14. <http://jsfiddle.net/mannejkumar/cjpS2/>
15. <https://refactoring.guru/design-patterns/observer/python/example>