

UFR-Sciences Université d'Angers

Master 2 - Informatique ACDI

Projet Django

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, centered on a dark green rectangular background.

Étudiants :

Choueib Alexandre
Loughran Christopher
Maignan Quentin

Enseignant :

Teletchea Stéphane

Sommaire

Présentation général du projet	3
L'objectif du site	3
Ses fonctionnalités	4
L'architecture du projet	4
Les applications django	4
Les modèles	5
Les vues	6
Les templates	6
Un exemple de requête de A à Z	7
Les fonctionnalités	7
Le déploiement docker	10
Le dockerfile	10
Le docker-compose	10
Lancer l'application sur un conteneur	11
Annexes	11

1. Présentation général du projet

a. L'objectif du site

Dans le cadre de notre **UE Culture GNU** nous avons dû créer un site web utilisant le framework Django.

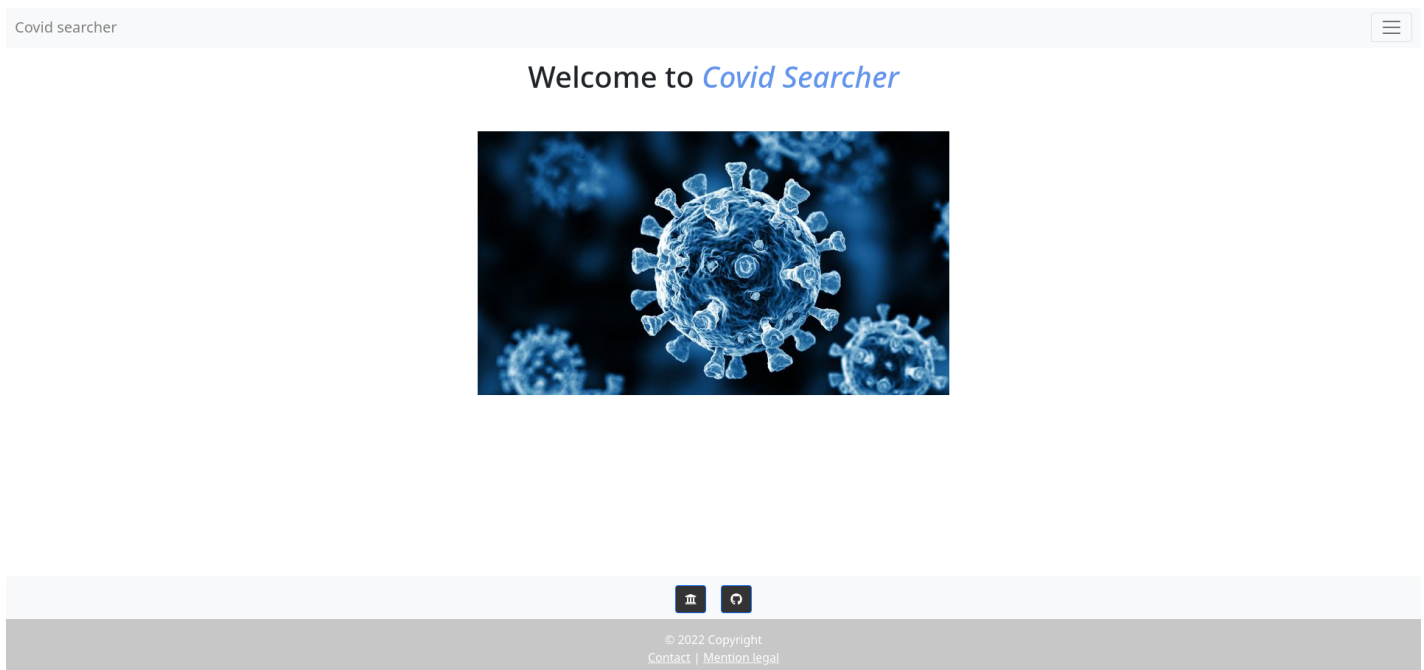
Les critères du site web nous ont été donnés par notre professeur et sont les suivantes :

- Le site doit être capable de récupérer sur le site <https://www.ncbi.nlm.nih.gov/> des articles scientifiques.
- Il doit aussi être capable de **parser** certaines données comme par exemple :
 - l'âge
 - mots clés liés au covid (Ivermectine, chloroquine, pfizer, etc)
- Afficher des statistiques (fréquence d'apparition des mots clés, etc)

Tous ces critères ont été respectés dans la conception et la réalisation du site django que nous vous transmettons.

Nous vous détaillerons dans la prochaine partie les fonctionnalités qui ont été implémentées.

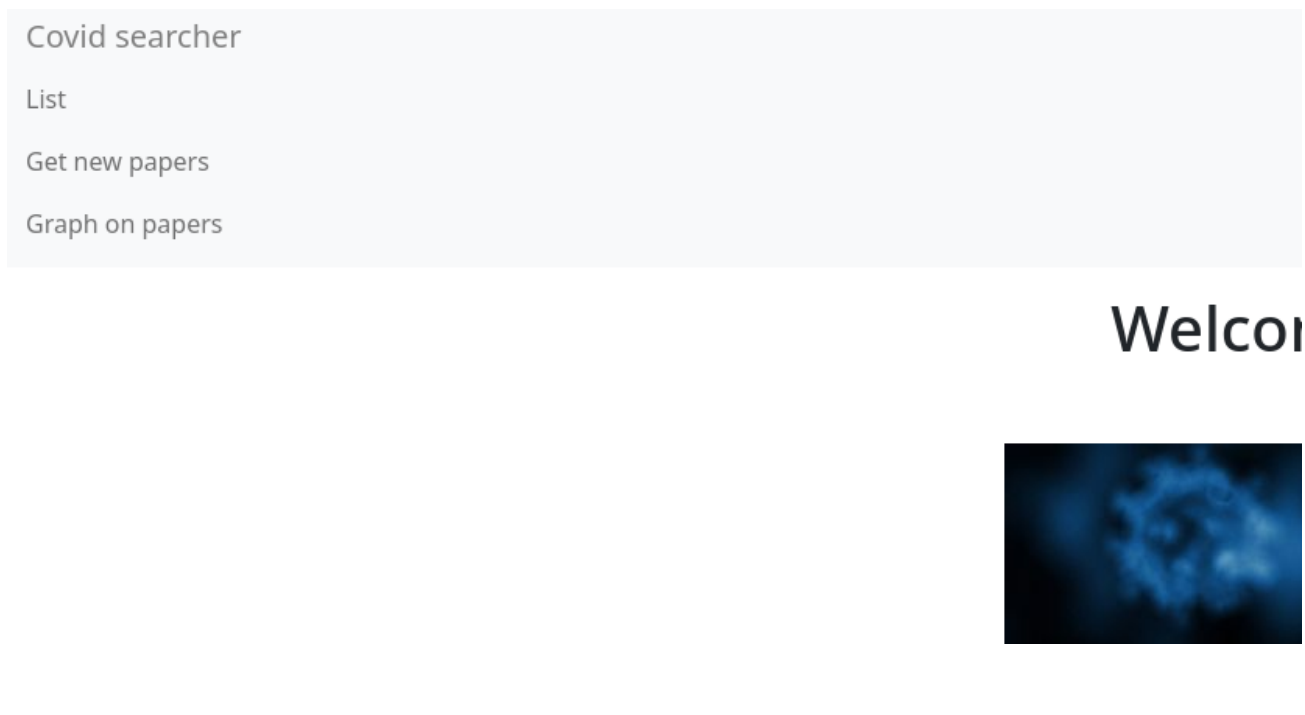
Voici la vue principale de la page d'accueil de notre site.



b. Ses fonctionnalités

Notre site "Covid searcher" respecte toutes les demandes stipulées plus haut.

Comme vous pouvez le voir sur la capture d'écran ci-dessous, nous avons un bouton burger qui permet d'afficher un menu avec quelques différentes options.



Chaque item nous redirige vers une page avec une fonctionnalité bien définie.

- **List** : Cette page nous affiche tous les articles stockés dans la base de données. On peut aussi filtrer ces articles avec des mots clés.
- **Get New Paper** : Cette page nous permet d'ajouter des nouveaux articles dans la base de données en entrant un sujet particulier dans un champ texte. Le site récupère les articles avec des occurrences du mot clé tapé sur le site ncbi .
- **Graph on papers** : Cette page récupère les données qui ont été récoltés et les affiche dans des graphes.

2. L'architecture du projet

a. Les applications django

Django divise son architecture en apps. Nous avons 3 apps dans notre projet.

La première est l'application par défaut de Django, elle se trouve dans le dossier djangoProject. Cette application contient le mapping de toutes nos routes. Elle contient également toutes les configurations nécessaires au bon fonctionnement de l'application, qui se trouve dans le fichier settings.py.

La deuxième application se trouve dans le dossier paper. Cette application gère les routes pour afficher un article, la liste de tous les articles et l'affichage de notre page de graphique.

La troisième, et dernière application, est dans le dossier d'information. Cette application nous permet de gérer les routes qui vont rajouter en base de données les articles scientifiques ainsi que les différentes informations le concernant.

b. Les modèles

Pour représenter nos objets en base de données, Django utilise un ORM (object relational mapping). Cet ORM nous a permis de créer des modèles de données pour effectuer les opérations de CRUD (create, read, update, delete) élémentaire très simplement.

Dans notre architecture nous avons deux modèles. Le premier représente nos articles scientifique :

```
class Paper(models.Model):
    paper_title = models.CharField(max_length=30)
    paper_year = models.CharField(max_length=30)
    paper_subject = models.CharField(max_length=30)
    paper_text = models.TextField()
    paper_most_used_words = models.TextField()
```

Nous avons plusieurs champs :

- Un id généré automatiquement par django
- L'année de publication de l'article
- Le sujet avec lequel l'utilisateur a cherché cette article
- Le contenu text de l'article
- La liste des 10 mots les plus utilisés dans cette article

Le deuxième modèle, Information, représente les mots clés trouvés ou non sur nos articles :

```
class Information(models.Model):
    paper_id = models.CharField(max_length=30)
    info_patient = models.BooleanField()
    info_molecule = models.BooleanField()
    info_ronapreve = models.BooleanField()
    info_molnupiravir = models.BooleanField()
    info_remdesivir = models.BooleanField()
    info_hydroxychloroquine = models.BooleanField()
    info_colchicine = models.BooleanField()
    info_azithromycine = models.BooleanField()
    info_avigan = models.BooleanField()
    info_anakinra = models.BooleanField()
    info_pfizer = models.BooleanField()
    info_moderna = models.BooleanField()
    info_astrazeneca = models.BooleanField()
```

Nous avons plusieurs champs :

- Un id généré automatiquement par django
- Un identifiant vers un article scientifique
- Quelques booléens qui enregistrent si un mot clé est présent dans l'article ou pas

c. Les vues

En django les vues sont le cerveau de notre application, c'est dans les fichiers `views.py` que le code métier se trouve. Toutes les vues fonctionnent sur le même principe. Elle dispose d'un ensemble d'attribut d'entrée, implémente un algorithme métier et retourne des données.

Dans notre cas nous avons choisi une architecture MVC (model view controller) donc nos vues (qui sont nos controller, **je sais c'est tordu**) nous retournent toujours un template Django. Par exemple :

```
def index(request):
    paper_list = Paper.objects.all()
    context = {'paper_list': paper_list}
    return render(request, '../templates/paperList.html', context)
```

Cette vue **index** prend en entrée une requête http pour retourner un template html ainsi qu'un context, qui sera utilisé dans le code HTML, dans lequel nous avons tous nos articles scientifiques.

d. Les templates

Les templates sont des fichiers HTML représentant nos pages web. Dans les templates nous pouvons utiliser du code python pour y faire des traitements.

Si dans la vues précédente nous avons envoyé un context comme celui-ci

```
context = {'paper_list': paper_list}
```

Nous pouvons ensuite dans le template HTML utiliser les doubles accolades pour accéder aux données de l'objet context, comme ceci :

```
{{ paper_list }}
```

Nous avons organisé nos templates pour ne pas avoir à faire du copier coller pour reproduire nos menus et styles entre chaque page. Nous avons donc un template qui sera la base de toutes nos pages, le template `base.html`. Tous les autres templates pourront alors utiliser ce template comme base. De plus, Django nous permet de définir des blocs dans nos templates. Dans le template de base nous avons donc plusieurs blocs. Un bloc *content* pour ajouter du contenu à nos pages. Un bloc *css* et *js* pour ajouter les fichiers css et javascript dans la partie head de notre base.

e. Un exemple de requête de A à Z

Prenons l'exemple de la page qui affiche la liste de tous nos articles disponibles dans notre base de données.

- L'utilisateur fait une requête get vers la route `/paper/list`
- Django fait le mapping entre cette route et la fonction `index` dans le fichier `views.py`
- La fonction `index` charge toutes les données de notre modèle `Paper` dans le contexte.
- La fonction `index` retourne un template HTML accompagné du contexte
- Le navigateur charge la page HTML avec le code python compilé
- L'utilisateur voit le résultat sur son navigateur, c'est à dire la liste de tous les articles scientifiques

3. Les fonctionnalités

Voici la page qui nous permet d'ajouter des articles dans la base de données. En entrant un terme de recherche et un le nombre d'articles `x` que l'on veut ajouter, on récupère sur le site `ncbi`, `x` articles qui ont pour sujet le terme de recherche.

The screenshot shows a web application interface. At the top, there's a header with 'Covid searcher' text and a hamburger menu icon. Below the header is a search bar with a dropdown menu. The dropdown menu is open, showing a list of search results. The first result is 'Risks of SARS-CoV-2 Breakthrough Infection and Hospitalization in Fully Vaccinated Patients With Multiple Myeloma'. To the right of the search bar is a blue button labeled 'Sélectionner'. Below the search bar is a filter input field with the placeholder text 'Filter papers containing a specific tag'. At the bottom of the page, there's a footer with two icons (a library icon and a refresh icon), the text '© 2022 Copyright', and links for 'Contact' and 'Mention legal'.

Voici la page pour voir la liste de tous les articles dans la base de données.

On peut filtrer les articles selon un mot grâce au champ texte, et visualiser les titres des articles via le `select`.

Reactivation of IgA vasculitis following COVID-19 vaccination

Sélectionner

- Risks of SARS-CoV-2 Breakthrough Infection and Hospitalization in Fully Vaccinated Patients With Multiple Myeloma
- Hemoglobin glycation index, calculated from a single fasting glucose value, as a prediction tool for severe hypoglycemia and major adverse cardiovascular events in DEVOTE
- Community acceptance and willingness to pay for hypothetical COVID-19 vaccines in a developing country; a web-based nationwide study in Nigeria
- Low peripheral blood derived neutrophil-to-lymphocyte ratio (dNLR) is associated with increased tumor T-cell infiltration and favorable outcomes to first-line pembrolizumab in non-small cell lung cancer
- One-year Review of COVID-19 in the Arab World
- Reactivation of IgA vasculitis following COVID-19 vaccination
- Undetectable SARS-CoV-2 active adaptive immunity—post-vaccination or post-COVID-19 severe disease—after immunosuppressants use
- Vacuna frente al COVID-19
- Vacunación anti-COVID-19: La realidad tras los ensayos clínicos
- New variants of SARS-CoV-2
- Myocarditis and COVID-19 mRNA vaccines: a mechanistic hypothesis involving dsRNA
- Mixing mRNA, adenoviral, and spike-adjuvant vaccines for protection against COVID-19
- Response to "Bilateral Retinal Detachments in a Healthy 22-year-old Female Following Moderna SARS-CoV-2 Vaccination"
- Vaccinating adolescents against SARS-CoV-2 in England: a risk-benefit analysis
- Return of Benefit to Society of Publicly Funded Innovations to Combat COVID-19
- SARS-CoV-2 breakthrough infections in vaccinated individuals: measurement, causes and impact
- No Common Ground: A Spatial-Relational Analysis of EU-China Relations
- Bamlanivimab Efficacy in Older and High-BMI Outpatients With COVID-19 Selected for Treatment in a Lottery-Based Allocation Process
- Immunogenicity and safety of a third dose of CoronaVac, and immune persistence of a two-dose schedule, in healthy adults: interim results from two single-centre, double-blind, randomised, placebo-controlled phase 2 clinical trials
- Safety of COVID-19 vaccination in patients with previous cerebral venous sinus thrombosis



Sélectionner un article nous emmène sur la page de cet article:

Title : Risks of SARS-CoV-2 Breakthrough Infection and Hospitalization in Fully Vaccinated Patients With Multiple Myeloma

Sujet : moderna

Date : 2021 Nov 23

Text :
Click to display text

Mots les plus utilisés : Introduction Data early COVID pandemic vaccines available showed patients multiple



Ici on peut visualiser le titre de l'article, le sujet utilisé pour rechercher cet article, ainsi que sa date de publication, et quelques mots clés présents dans l'article, si on clique sur les mots clé cela nous affiche tous les articles contenant ce mot clé (les 10 mots les plus utilisés dans l'article).

De plus, on peut afficher le texte de l'article (qui est caché par défaut), on voit ici, que les mots clés sont coloriés en rouge.

Covid searcher

Title : Risks of SARS-CoV-2 Breakthrough Infection and Hospitalization in Fully Vaccinated Patients With Multiple Myeloma

Sujet : moderna

Date : 2021 Nov 23

Text :

Introduction Data from early in the COVID- pandemic when vaccines were not available showed that patients with multiple myeloma (MM) were at increased risk for COVID- infection and severe outcomes., Recent studies showed a low rate of seroconversion after messenger RNA (mRNA) anti-SARS-CoV- vaccination in patients with MM and other hematological malignant neoplasms., However, the risk and outcomes of SARS-CoV- breakthrough infection in vaccinated patients with MM remains unknown. Methods This cohort study used the cloud based TriNetX Analytics network platform to access deidentified patient electronic health records (EHRs) from health care organizations in the United States. The study population comprised e patients who fulfilled the following inclusion criteria: had recent medical encounter(s) with health care organizations since December , ; had documented evidence of full vaccination in the EHRs (Pfizer-BioNTech, Moderna, or Johnson & Johnson vaccine) between December , , and October , ; and had no prior COVID- infection. EHR data are deidentified, and this study was exempt from institutional review board approval and informed consent per the US Federal Policy for the Protection of Human Subjects. We tested whether fully vaccinated patients with MM had higher risk for breakthrough infections than individuals without cancer after propensity score matching for demographics, adverse socioeconomic determinants of health, transplant procedures, comorbidities, vaccine types, and medications. Kaplan-Meier analysis was used to estimate probability of breakthrough infections starting days after full vaccination. Comparisons between cohorts were made using Co proportional hazards model and hazard ratio (HR). We tested whether hospitalization rates differed between patients with MM with breakthrough infections and propensity scorematched patients with MM without breakthrough infections. Statistical tests were either conducted within the TriNetX Analytics Platform or using R statistical software (version .) with significance set at $P < .05$ (-sided). Details of TriNetX, study population, and statistical analysis were described in the eMethods in the Supplement. This study followed the Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) reporting guidelines. Results The characteristics of the fully vaccinated population with MM and the population of patients without cancer are shown in the Table. Among vaccinated patients with MM, .% had monoclonal gammopathy of undetermined significance (MGUS), .% were in relapse, .% had never achieved remission, .% had chemotherapy, .% had targeted therapy, .% had radiation therapy, and .% had stem cell transplant; mean (SD) blood lymphocyte count was . ece/L (.ece/L). Among patients with MM with SARS-CoV- breakthrough infections, .% had MGUS, .% were in relapse, .% had never achieved remission, .% had chemotherapy, .% had targeted therapy, .% had radiation therapy, and .% had stem cell transplant; mean (SD) blood lymphocytes count was .ece/L (.ece/L). The overall risk of SARS-CoV- breakthrough infections was .% in the MM population and .% in the noncancer population. After propensity score matching for demographics, adverse socioeconomic determinants of health, transplant procedures, comorbidities, vaccine types, and medications, patients with MM remained at significantly increased risk for breakthrough infections compared with matched patients without cancer (HR, .; 95% CI, .-). The estimated probability of hospitalization at the end of the time window (October ,) was .% for patients with MM with breakthroughs, compared with .% for matched patients without breakthroughs (HR, .; 95% CI, .-) (Figure). Table. Characteristics of the Vaccinated Populations With MM and Without Cancer (as of October ,) in the TriNetX Database

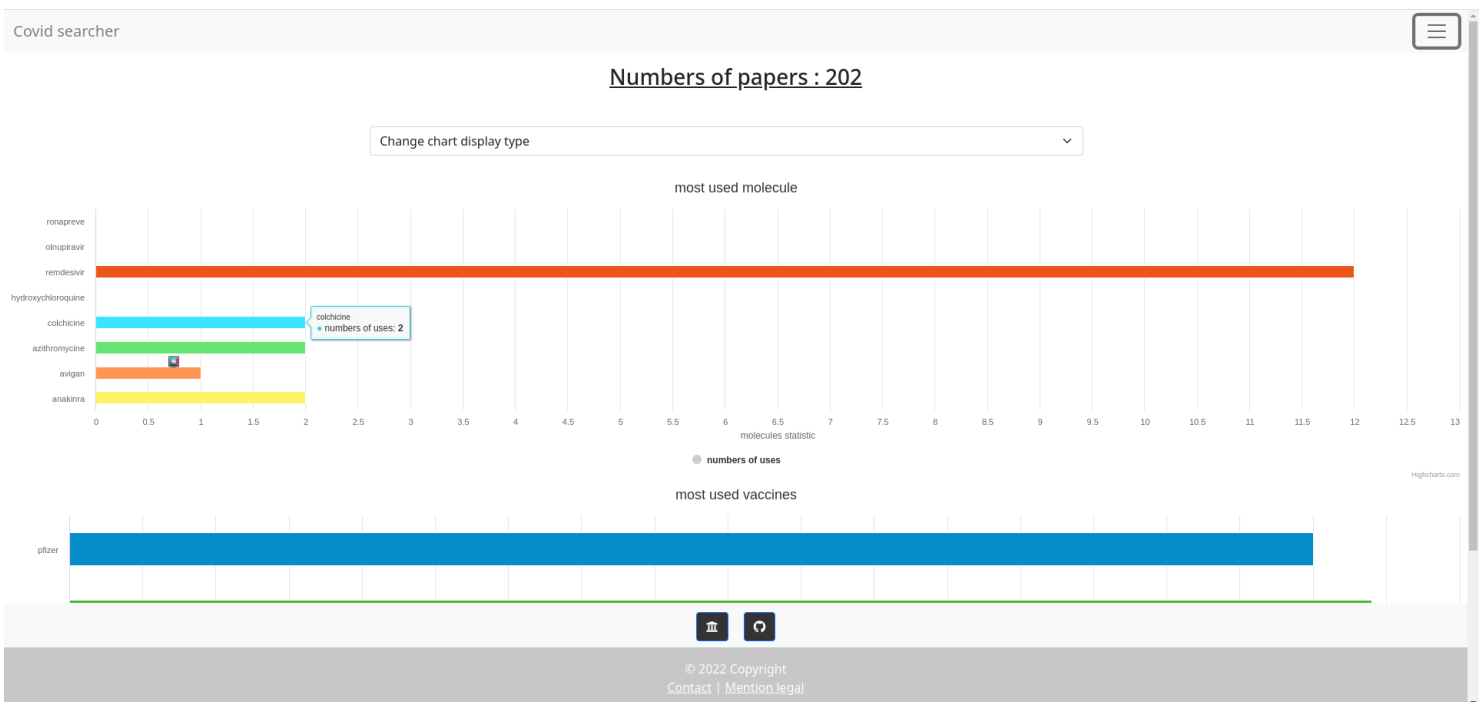
Patient characteristics (fully vaccinated)	Multiple myeloma	Without cancer	P value
Total No. of patients	100	100	
Age, mean (SD), y (I)	68 (10)	68 (10)	
Gender			
Female	50	50	
Male	50	50	
Race and ethnicity			
White	70	70	
Black or African American	10	10	
Hispanic or Latino	5	5	
Asian	5	5	
Unknown	5	5	
Comorbidity			
Hypertension	30	30	
Heart diseases	20	20	
Type diabetes	10	10	
Overweight or obesity	40	40	
Cerebrovascular diseases	15	15	
Chronic lung diseases	10	10	
Chronic kidney disease	5	5	
Liver diseases	5	5	
HIV	5	5	
Alcohol use	5	5	
Tobacco use	5	5	
Substance use disorders	5	5	
Organ or tissue transplant	5	5	
Adverse socioeconomic determinants of health			
Full vaccination type			
Pfizer-BioNTech	50	50	
Moderna	50	50	
Johnson & Johnson	5	5	
COVID-related medications			
Deamethasone	5	5	
Hydrocortisone	5	5	
Ibuprofen	5	5	
Methylprednisolone	5	5	
Prednisone	5	5	
Naproxen	5	5	
Remdesivir	5	5	
Tocilizumab	5	5	
Lopinavir	5	5	
Ritonavir	5	5	
Imdevimab	5	5	

a Demonstrated risk factors for COVID- infections or severe outcomes that occurred any time before to the same day as index event of full vaccination are shown. The status of adverse socioeconomic determinants of health was based on the International Statistical Classification of Diseases and Related Health Problems, Tenth Revision (ICD-10) code ecPersons with potential health hazards related to socioeconomic and psychosocial circumstances (Z-Z), which includes ecProblems related to education and literacy (Z), ecProblems related to employment and unemployment (Z), ecOccupational exposure to risk factors (Z), ecProblems related to housing and economic circumstances (Z), ecProblems related to upbringing (Z), among others. Figure. Risk of Hospitalization for Patients With vs Without

© 2022 Copyright

Contact | Mention legal

Finalement on peut visualiser des statistiques portant sur tous les articles grâce à la page */paper/graph*, ici on voit le nombre d'articles présent dans la base de données, ainsi que la fréquence de certains mots clés (liés au Covid-19).



4. Le déploiement docker

a. Le dockerfile

Le dockerfile est un fichier que docker va utiliser pour build notre image de conteneur. Tout conteneur se base sur une image. Ces images sont des briques que l'on va assembler pour créer un conteneur. Nous allons nous baser sur une image docker déjà existante pour avoir une image de base pour développer en python https://hub.docker.com/_/python.

A notre conteneur nous allons rajouter l'entièreté de nos sources avec la commande docker COPY, puis installer les librairies pythons qui sont présentes dans notre fichier de requirement. Ensuite nous allons effectuer des téléchargements complémentaires pour une librairies que nous avons utilisé.

b. Le docker-compose

Docker-compose est un outil docker qui permet de déployer des conteneurs facilement grâce à des options rentrées dans un fichier YAML.

Dans le fichier docker-compose.yml vous allez trouver trois services très basiques

service web : attache un environnement et un volumes

service web migrate : hérite du service web, ce service va se charger de lancer les migrations django pour initialiser nos table dans notre base de données

service web run : va lancer le serveur python sur l'url 0.0.0.0:8090 // localhost:8090

Le docker compose va utiliser notre image créée par notre dockerfile pour instancier les conteneurs.

c. Lancer l'application sur un conteneur

Pour lancer l'application, rien de plus simple ! Il vous suffit de vous rendre dans le dossier où se trouve le fichier dockerfile et docker-compose.yml et de faire un `docker-compose up -d`. Vous pourrez ensuite accéder à l'url suivante : <http://localhost:8090/>

Si votre port 8090 est déjà utilisé, vous pouvez changer le port utilisé en un instant dans le fichier docker-compose.yml.

Si vous ne parvenez pas à pull les images, c'est peut-être que vous n'êtes pas connecté à docker <https://docs.docker.com/engine/reference/commandline/login/>.

5. Annexes

Comprendre docker avec django : <https://docs.docker.com/samples/django/>

Documentation Django 4 : <https://docs.djangoproject.com/en/4.0/>