

Master 1 RSD Technologie des Applications Web Examen (Partie 1) - 20 Janvier 2025 Durée : 75 Minutes	Nom & Prénom	Gr.	Note
		...	

Partie 1 : QCM, cochez la bonne réponse !
 (+ 1 : Juste ; -1 : Fausse ; 0 : Pas de réponse)

Un total négatif est automatiquement ramené à zéro

1. Quelle est la définition exacte du Web ?

- ☐ Un protocole pour transférer des données sur Internet.
- ☐ Un système hypermédia permettant d'accéder aux ressources du réseau Internet.
- ☐ Un langage utilisé pour structurer des pages Web.
- ☐ Une application dédiée au partage de fichiers.

2. Quelle caractéristique distingue une ressource JSON d'une ressource XML ?

- ☐ JSON est orienté documents, tandis que XML est orienté données.
- ☐ XML est plus rapide pour échanger des données.
- ☐ JSON est plus léger et orienté données.
- ☐ JSON ne supporte pas les hiérarchies complexes.

3. Que signifie 'batteries included' dans le contexte de Django ?

- ☐ L'intégration d'outils tiers préinstallés.
- ☐ La disponibilité de nombreuses fonctionnalités prêtes à l'emploi.
- ☐ La possibilité de personnaliser tous les composants.
- ☐ La compatibilité avec tous les frameworks front end.

4. Dans Django, quel fichier est utilisé pour configurer la base de données ?

- ☐ views.py
- ☐ models.py
- ☐ settings.py
- ☐ urls.py

5. Dans une requête HTTP POST, où se trouvent les données envoyées au serveur ?

- ☐ Dans l'en-tête de la requête.
- ☐ Dans le corps de la requête.
- ☐ Dans l'URL de la requête.
- ☐ Les données ne sont pas envoyées au serveur.

6. Que fait la commande python manage.py makemigrations ?

- ☐ Crée un modèle de base de données.
- ☐ Applique les migrations à la base de données.
- ☐ Génère les fichiers de migration pour les modifications des modèles.
- ☐ Supprime toutes les migrations précédentes.

7. Quel est le rôle de csrf_token (Cross-Site Request Forgery Token) dans les formulaires Django ?

- ☐ Empêcher les soumissions multiples d'un formulaire.
- ☐ Assurer que le formulaire est envoyé depuis une source de confiance.
- ☐ Valider automatiquement les données envoyées dans le formulaire.
- ☐ Authentifier les utilisateurs envoyant le formulaire.

8. Comment est-ce que Django organise les fichiers statiques comme les images et les fichiers CSS ?

- ☐ En utilisant le dossier /templates.
- ☐ En les plaçant directement dans le dossier /views.
- ☐ En utilisant un dossier nommé /static dans chaque application.
- ☐ En stockant tous les fichiers dans /settings.

9. Comment activer l'administration Django pour un modèle ?

- ☐ Ajouter le modèle dans views.py.
- ☐ L'enregistrer dans admin.py avec admin.site.register().
- ☐ Le configurer dans urls.py.
- ☐ Ajouter un décorateur @admin.register() dans models.py.

10. Comment définir une relation un-à-plusieurs dans Django ?

- ☐ Utiliser un champ ForeignKey.
- ☐ Utiliser un champ ManyToManyField.
- ☐ Utiliser un champ OneToOneField.
- ☐ Définir manuellement les relations dans la base de données.

Master 1 RSD - Technologie des Applications Web Examen (Partie 2) 20 Janvier 2025 - Durée : 75 Minutes	Nom & Prénom	Gr.	Note
		...	

Partie 2 : Utilisation du Framework Django 5 (0,5 pt par ligne complétée)

Nous allons développer une application Django 5, *cultura*, permettant de gérer des événements culturels (concerts, expositions, conférences) sur une plateforme en ligne. Les utilisateurs pourront *consulter* les événements, *réserver* des places et *obtenir des détails* sur chaque événement.

Les modèles de l'application *cultura* :

Modèle Participant	
first_name	Prénom de l'utilisateur, limité à 100 caractères.
last_name	Nom de l'utilisateur, limité à 100 caractères.
email	Adresse email unique de l'utilisateur.
phone	Numéro de téléphone, limité à 15 caractères.
Modèle Event	
tag	Code unique 50 caractères pour l'évènement.
title	Titre de l'évènement, limité à 200 caractères.
date	Date de l'évènement.
location	Lieu de l'évènement, limité à 200 caractères.
description	Description de l'évènement.
seats_number	Le nombre d'emplacement disponibles
Modèle Booking	
user	Clé étrangère avec le modèle Participant (User n'existe pas !)
event	Clé étrangère avec le modèle Event.
seats	Nombre de places réservées, de type entier.

1. Event :

tag	title	date	location	description	seats_number
eac25	Exposition d'Art Contemporain 2025	2025-03-15	Musée des Beaux-Arts	Découvrez les chefs-d'œuvre modernes.	800
ia25	Conférence sur l'IA 2025	2025-04-10	Centre Culturel	Présentation des dernières avancées en IA.	600
c25r1	Concert Rai – Premier 2025	2025-07-05	Stade 5 Juillet	Concert musical pour la fête d'indépendance.	50 000

Echantillons de données :

1. Participant :

first_name	last_name	email	phone
Lamis	Foughali	lamis.f@mail.com	0771234567
Louey	Soltane	louey.f@mail.com	0669987654
Ouail	Ziwane	ouail.f@mail.com	0557891234

2. Booking :

email	tag	seats
lamis.f@mail.com	eac25	2
ouail.f@mail.com	ia25	1
louey.f@mail.com	c25r1	5

Nous supposons qu'un environnement virtuel Python a été activé et que Django 5 est y installé. Il est demandé de :

1. Crée le projet `cfg` dans le dossier de l'environnement virtuel Python (0,5 pt)

```
(venv) PS E:\cultura>. . . . .
```

2. Crée l'application `cultura` (0,5 pt)

```
(venv) PS E:\cultura>. . . . .
```

3. On supposant que `cultura` a été ajoutée au projet, Complète `cfg/urls.py` pour inclure `urls.py` dans celui de `cfg` (0,5 pt)

```
# cfg/urls.py
```

```
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls), # Décalage, indentation incorrect ! Donc, je pénalise pas ici.
```

```
. . . . .
```

```
]
```

4. Complète le fichier `cultura/urls.py` pour inclure la vue `book` (0,5 pt)

```
# cultura/urls.py
```

```
# Format: http://127.0.0.1/cultura/book/tag/email/seats (http://127.0.0.1/cultura/book/eac25/lamis.f@mail.com/2)
```

```
from django.urls import path
from .views import book_event
```

```
urlpatterns = [
```

```
. . . . .
```

```
]
```

5. Complète les modèles de l'application cultura (0,5 pt x 13 = 6,5 pts)

```
# cultura/models.py
```

```
from django.db import models
```

```
class Participant(models.Model):
```

```
.....
```

```
class Event(models.Model):
```

```
.....
```

```
class Booking(models.Model):
```

```
.....
```

6. Complète la vue book_event qui permet la réservation (0,5 pt x 3 = 1,5 pts)

```
# culura/views.py

from django.http import JsonResponse
from .models import Participant, Event, Booking

def book_event (request, tag, email, seats):
    try:

        # Récupère l'événement 'event' :

        . . . . .

        # Récupère le participant 'participant' :

        . . . . .

        # Crée la réservation 'booking' :

        . . . . .

        return JsonResponse({

            'status': 'succès',

            'message': f'Réservation réussie pour {seats} places à l'événement {event.title}.'

        })

    except Exception as e:

        return JsonResponse({

            'status': 'erreur',

            'message': f'Erreur lors de la réservation: {str(e)}'

        })
```