

---

# Manuel d'utilisation Bacchanator

---

## 1 Table des matières

1	Présentation du projet .....	3
1.1	Introduction.....	3
1.2	Description du jeu.....	3
1.3	Membres .....	3
2	Installation .....	4
2.1	Git.....	4
2.2	Configuration Django.....	4
3	Site d'administration et algorithme .....	5
3.1	Descriptions des objets .....	6
3.1.1	Objet Image .....	6
3.1.2	Objet Tag .....	6
3.1.3	Objet Contenir.....	6
3.1.4	Objet Question .....	6
3.1.5	Objet Concerner.....	7
3.1.6	Objet Réponse .....	7
3.1.7	Objet Associer.....	7
3.2	Description de l'algorithme .....	7



# 1 Présentation du projet

## 1.1 Introduction

Dans le cadre de l'UE Projets de communication transdisciplinaire en L3 informatique, l'équipe est amenée à proposer un projet pour le musée des beaux-arts de Bordeaux.

Dans ce contexte *Bacchanator* a été proposé. Le projet est un jeu consistant à deviner l'œuvre à laquelle l'utilisateur pense.

## 1.2 Description du jeu

Dans ce jeu une série de questions seront posées au joueur, des questions portant sur les caractéristiques de l'œuvre, auxquelles il devra répondre parmi les réponses disponibles. Cela permet de deviner celle-ci.

À la fin l'œuvre sera affichée, ainsi que toutes ses descriptions (l'auteur, date de création, domaine, ...etc.).

## 1.3 Membres

- **Abdoulhay Ibrahim Aridouane :**
  - Chef de projet, élaboration de l'algorithme de recherche sur la base de données (BD), remplissage de la BD.
- **Mouctar Diallo**
  - Élaboration de l'algorithme de recherche sur la BD, remplissage de la BD.
- **Aguibou Barry :**
  - Responsivité du site, design du site, test de la BD.
- **Ousmane Bah :**
  - Structure du site, test de la BD.
- **Ibrahima Diallo :**
  - Design du site, remplissage de la BD.

- **Mamadou Saliou Diallo :**
  - Test de la BD et de la responsivité du site.

Enseignants : Guillaume Blin, Aurélie Bugeau.

## 2 Installation

Les prochaines commandes doivent être exécutées dans un terminal.

Ouvrir un terminal :

- Sur Linux (Ubuntu) : **Ctrl + Alt + T**

### 2.1 Git

Cloner le dépôt du git :

`git clone https://github.com/aarid/bacchanight.git`

### 2.2 Configuration Django

Entrer dans le dépôt :

- `cd bacchanight`

Créez un environnement virtuel **env** :

- `python3 -m venv env`

Placer vous à l'intérieur et installer les applications de Django :

- `source myvenv/bin/activate`
  - `python -m pip install --upgrade pip`
  - `pip install -r requirements.txt`
  - `pip install Pillow`
  - `pip install django-cleanup`

Lancement du serveur :

- `python manage.py runserver`
- Se connecter sur <http://127.0.0.1:8000/> et <http://127.0.0.1:8000/admin/> pour le site d'administration

Migration des tables (pour actualiser les modifications faites dans le code) :

- `python manage.py makemigrations blog`

Application des migrations :

- `python manage.py migrate blog`

### 3 Site d'administration et algorithme

Administration du site

#### AUTHENTIFICATION ET AUTORISATION

Groupe	+ Ajouter	✎ Modifier
--------	-----------	------------

Utilisateurs	+ Ajouter	✎ Modifier
--------------	-----------	------------

#### BLOG

Associés	+ Ajouter	✎ Modifier
----------	-----------	------------

Concerners	+ Ajouter	✎ Modifier
------------	-----------	------------

Contenirs	+ Ajouter	✎ Modifier
-----------	-----------	------------

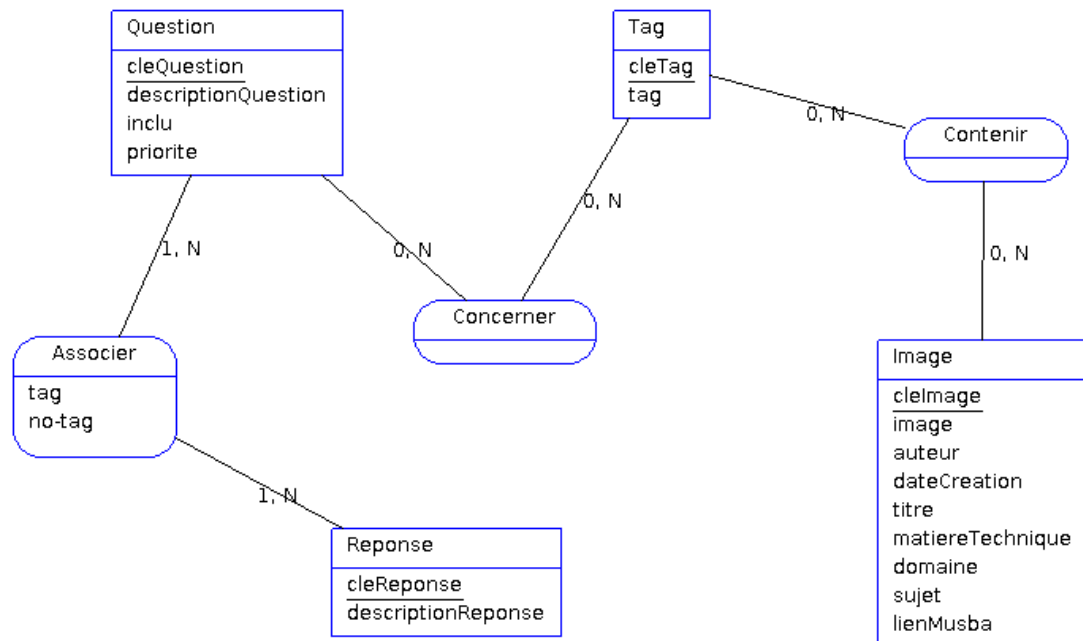
Images	+ Ajouter	✎ Modifier
--------	-----------	------------

Questions	+ Ajouter	✎ Modifier
-----------	-----------	------------

Reponses	+ Ajouter	✎ Modifier
----------	-----------	------------

Tags	+ Ajouter	✎ Modifier
------	-----------	------------

## 3.1 Descriptions des objets



### 3.1.1 Objet Image

Il représente une image dans la base de données ainsi que ses différentes informations : auteur, titre, l'image elle-même, date de création, matière technique, le domaine, sujet ainsi que lien du site du musée où elle a été téléchargée.

### 3.1.2 Objet Tag

Il représente une caractéristique sur une image, par exemple Personnes s'il y a des personnes sur l'image.

### 3.1.3 Objet Contenir

Une image peut contenir un ou plusieurs tags, par exemple pour un tableau contenant un seul homme, les tags seront **Personnes**, **Personne 1**, **Homme**.

### 3.1.4 Objet Question

Il représente une question.

- Une question a une priorité, par exemple la question « *Y a-t-il des personnes sur l'œuvre ?* » est plus prioritaire que « *Est-ce un portrait ?* ».
- Une question peut être incluse dans un tag, par exemple « *Existe-il un homme* » est inclus dans le tag **Personnes**.

### 3.1.5 Objet Concerner

Les questions peuvent concerner un ou plusieurs tags, cela permet de filtrer les questions aussi. Par exemple on ne posera des questions qui concernent des animaux si au bout d'un moment l'utilisateur a indiqué qu'il n'y avait pas d'animaux sur l'œuvre.

### 3.1.6 Objet Réponse

Les questions sont associées à une ou plusieurs réponses. Les réponses sont les tags qui seront appliqués aux images et aux questions.

### 3.1.7 Objet Associer

C'est la liaison entre les questions et les réponses. Il contient 2 champs (tag et no\_tag) qui font la liaison entre les réponses et les tags.

## 3.2 Description de l'algorithme

Une fois que l'utilisateur a choisi la salle de l'image :

- Tant qu'il y a des images :
  - Récupération de la prochaine question :
    - Grâce aux objets Concerner on récupère les questions liées au tag en faisant un filtre, puis choix de la question la plus prioritaire.
    - Filtre aussi des images.
      - S'il ne reste qu'une seule image → Affichage de l'image.
      - S'il ne reste aucune image ou question → Rejoue.
      - Sinon on continue.
  - Envoi de la question et des réponses associées à la prochaine page.
  - Affichage de la question et des réponses.