|  |
| --- |
| tpi – Création d’un site web d’initiation à l’astronomie |

25 mai 2022

CPNV

JEGATHEESWARAN Nithujan

Table des matières

[1. Analyse préliminaire 3](#_Toc104388308)

[1.1. Introduction 3](#_Toc104388309)

[1.2. Objectifs 3](#_Toc104388310)

[1.3. Planification initiale 4](#_Toc104388311)

[1.3.1. Dates du projet 4](#_Toc104388312)

[1.3.2. Tâches 4](#_Toc104388313)

[1.3.3. Diagramme de Gantt 5](#_Toc104388314)

[2. Analyse / Conception 6](#_Toc104388315)

[2.1. Concept 6](#_Toc104388316)

[2.1.1. Base de données 6](#_Toc104388317)

[2.1.2. Maquettes des pages web 8](#_Toc104388318)

[2.2. Stratégie de test 14](#_Toc104388319)

[2.3. Risques techniques 14](#_Toc104388320)

[2.4. Planification 15](#_Toc104388321)

[2.5. Dossier de conception 16](#_Toc104388322)

[2.5.1. Logiciels utilisés 16](#_Toc104388323)

[2.5.2. Base de données 17](#_Toc104388324)

[2.5.3. Site Web 18](#_Toc104388325)

[3. Réalisation 19](#_Toc104388326)

[3.1. Dossier de réalisation 19](#_Toc104388327)

[3.1.1. Base de données 19](#_Toc104388328)

[3.1.2. Site web 21](#_Toc104388329)

[3.2. Description des tests effectués 23](#_Toc104388330)

[3.3. Erreurs restantes 25](#_Toc104388331)

[3.4. Liste des documents fournis 25](#_Toc104388332)

[4. Conclusions 26](#_Toc104388333)

[5. Annexes 27](#_Toc104388334)

[5.1. Résumé du rapport TPI 27](#_Toc104388335)

[5.2. Sources – Bibliographie 28](#_Toc104388336)

[5.3. Journal de travail 32](#_Toc104388337)

[6. Glossaire 36](#_Toc104388338)

# Analyse préliminaire

## Introduction

Ce projet est réalisé dans le cadre d’une formation FPA (Formation Professionnelle Accélérée) permettant d’obtenir un CFC. Aucun travail n’a été effectué au préalable.

Ce projet a pour but de créer un site web qui a pour thème l’astrologie. Il devra permettre de consulter diverses informations sur les objets du catalogue de Messier et d’organiser des sorties nocturnes tout en offrant la possibilité aux visiteurs du site de s’y inscrire.

Ce projet est réalisé en Python car c’est un langage de programmation polyvalent. Le site web communiquera avec un serveur PostgreSQL qui contiendra toutes les données persistantes. Ce SGBDRO[[1]](#footnote-1) a été choisi pour faciliter la mise en ligne du site mais également car il est entièrement gratuit.

## Objectifs

Les objectifs généraux du projet sont :

* Une page d’accueil qui présente le site et ses objectifs
* Une page contenant les objets du catalogue de Messier avec la possibilité de faire des recherches et de les trier par ordre alphabétique ou numérique selon la donnée
* Une page sur les sorties nocturnes disponibles avec la possibilité pour le visiteur du site de s’y inscrire
* Une section réservée aux administrateurs qui permet la création de sorties nocturnes
* Le site sera intégralement responsive
* L’automatisation et la réussite de l’import des données nécessaires (fichier CSV[[2]](#footnote-2) et images sur internet) dans la base de données choisie

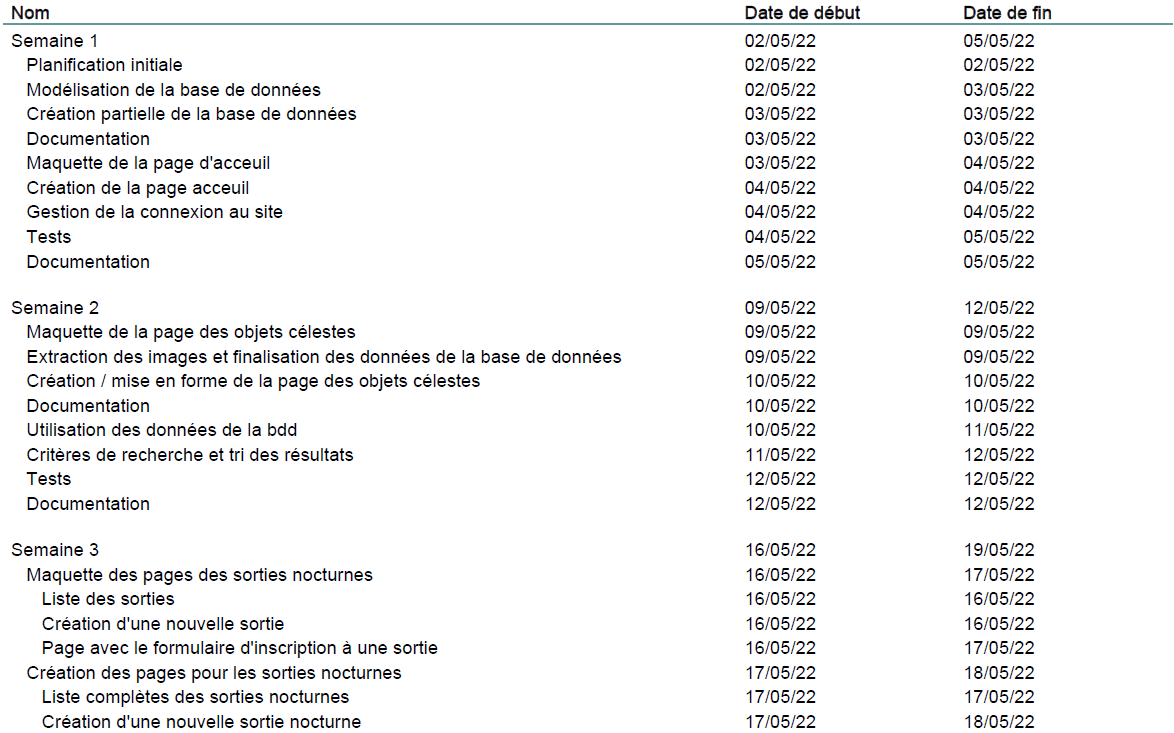
## Planification initiale

La méthode agile est la méthode de gestion de projet qui a été utilisée lors de la réalisation de ce projet. Cette méthode permet de diviser le temps de travail en itérations appelées et de travailler sur des mini-projets sur chacun d’entre eux. Les fonctionnalités demandées par le cahier des charges permettent de planifier de telle sorte qu’à la fin d’un sprint le projet est fonctionnel car il n’y a pas d’éléments qui manquent.

### Dates du projet

Du 02.05.22 au 30.05.22

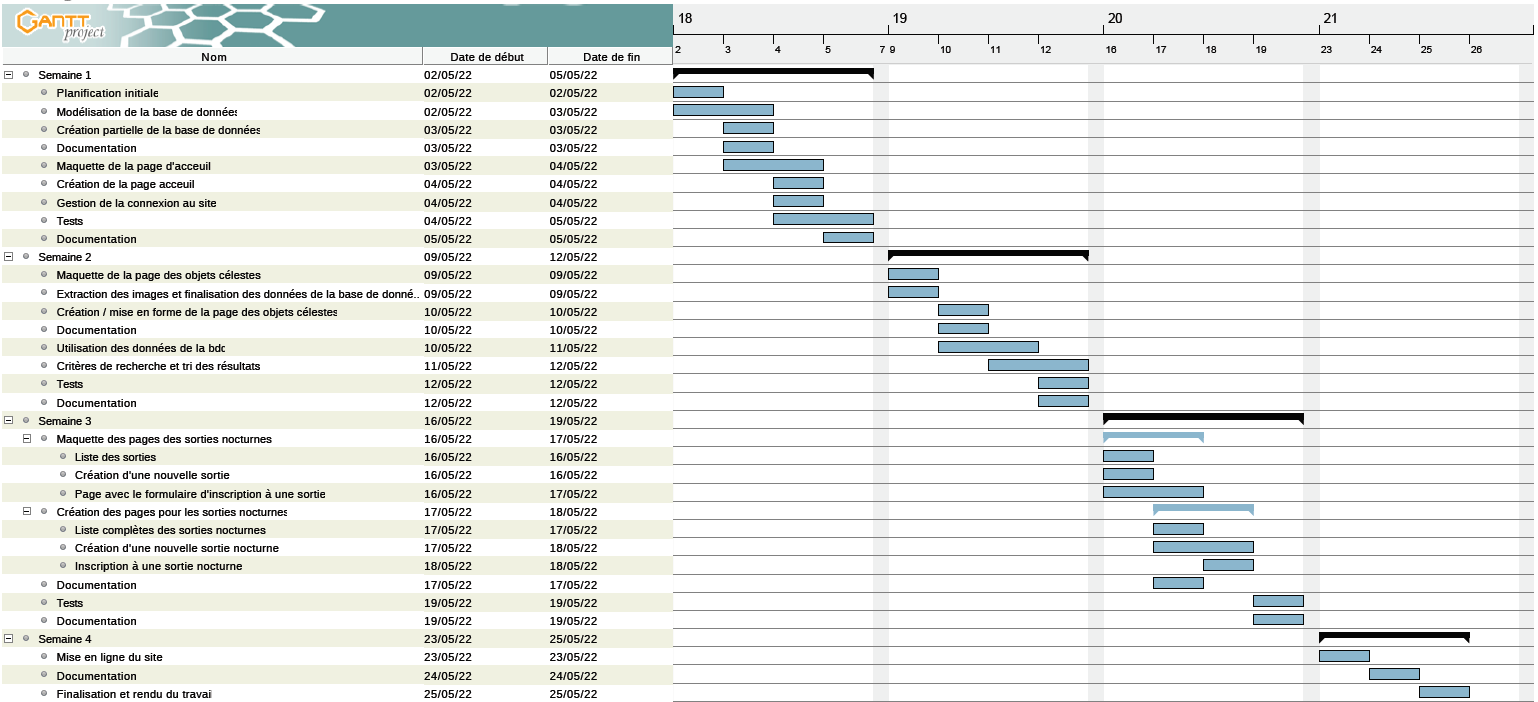
### Tâches



*Rem :*

* Le rendu du travail a été prévu le 25 mai à cause d’une indisponibilité le lundi 30 mai
* La gestion du tâches du projet est faite sur [IceScrum](https://icescrum.cpnv.ch/p/TPIASTRONO/#/project)

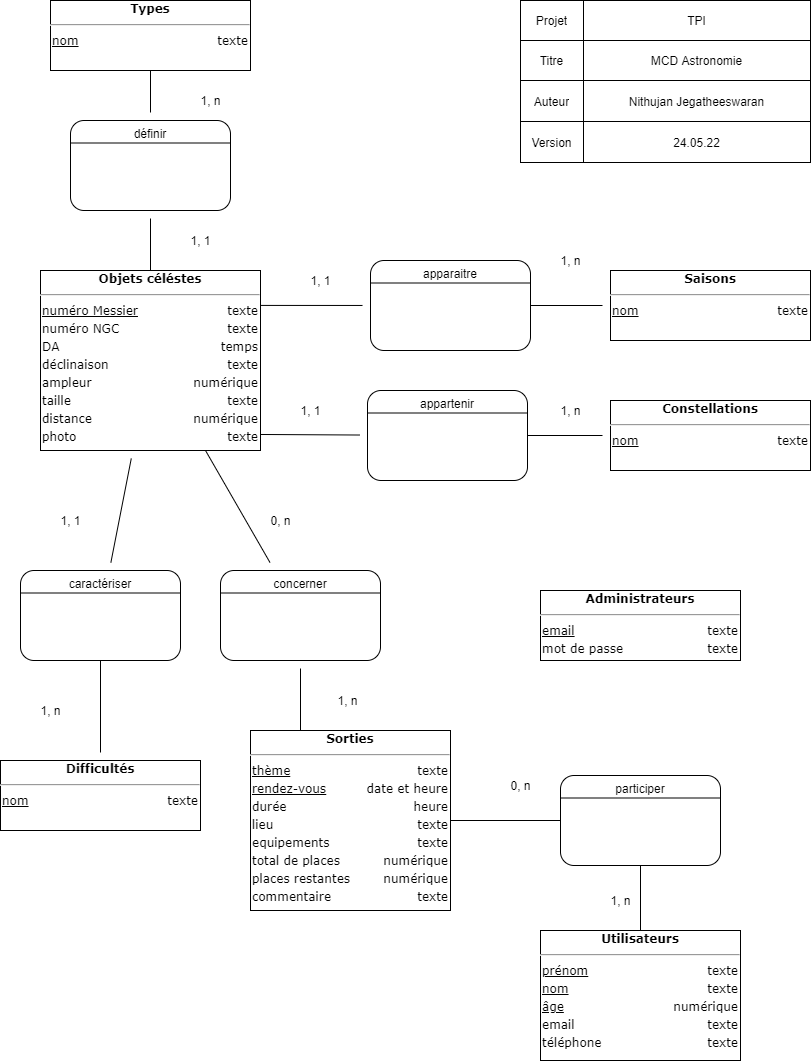
### Diagramme de Gantt

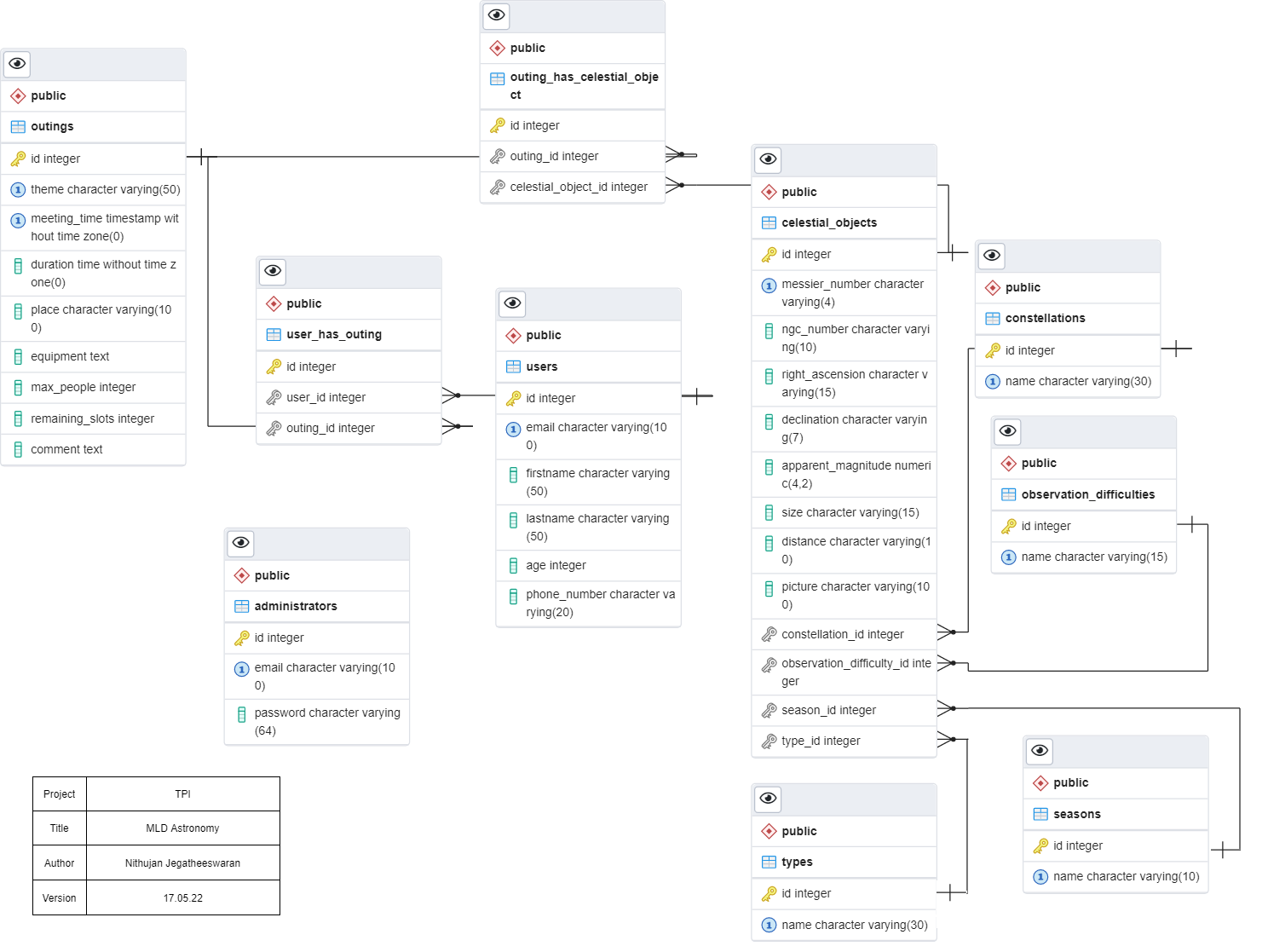


# Analyse / Conception

## Concept

### Base de données

**MCD (Modèle Conceptuel de Données)**

**MLD (Modèle Logique de Données)**

### Maquettes des pages web

|  |  |  |
| --- | --- | --- |
| Page d’accueil : | maquette-acceuil | |
| Page de login : | | Une image contenant texte, capture d’écran, moniteur, écran  Description générée automatiquement |

*Rem*:

Cette page a été réalisé sans l’aide de maquettes car elle reprend des éléments de la maquette précédente. Les éléments qui varient sont minimes.

|  |  |
| --- | --- |
| Page avec les objets du catalogue de Messier : | maquettes-Catalogue Messier |

|  |  |
| --- | --- |
| Page pour ajouter une sortie (admin) : | Sorties-Admin |

|  |  |
| --- | --- |
| Page avec la liste des sorties : |  |

|  |  |
| --- | --- |
| Page pour s’inscrire à une sortie : | Une image contenant texte, capture d’écran, moniteur, noir  Description générée automatiquement |

*Rem :*

Les deux dernières pages ont été conçues directement via du code car elles reprennent les éléments des autres pages

## Stratégie de test

Les tests de l’application web sont tous manuels. Ils ont été faits sur plusieurs navigateurs afin de garantir un maximum de compatibilité et ainsi donner plus de choix aux utilisateurs lors de leur visite sur le site. Les tests sont faits en parallèle à l’implémentation des diverses fonctionnalités.

Les tests seront effectués par moi-même mais également par mon frère et certaines connaissances si le temps le permet.

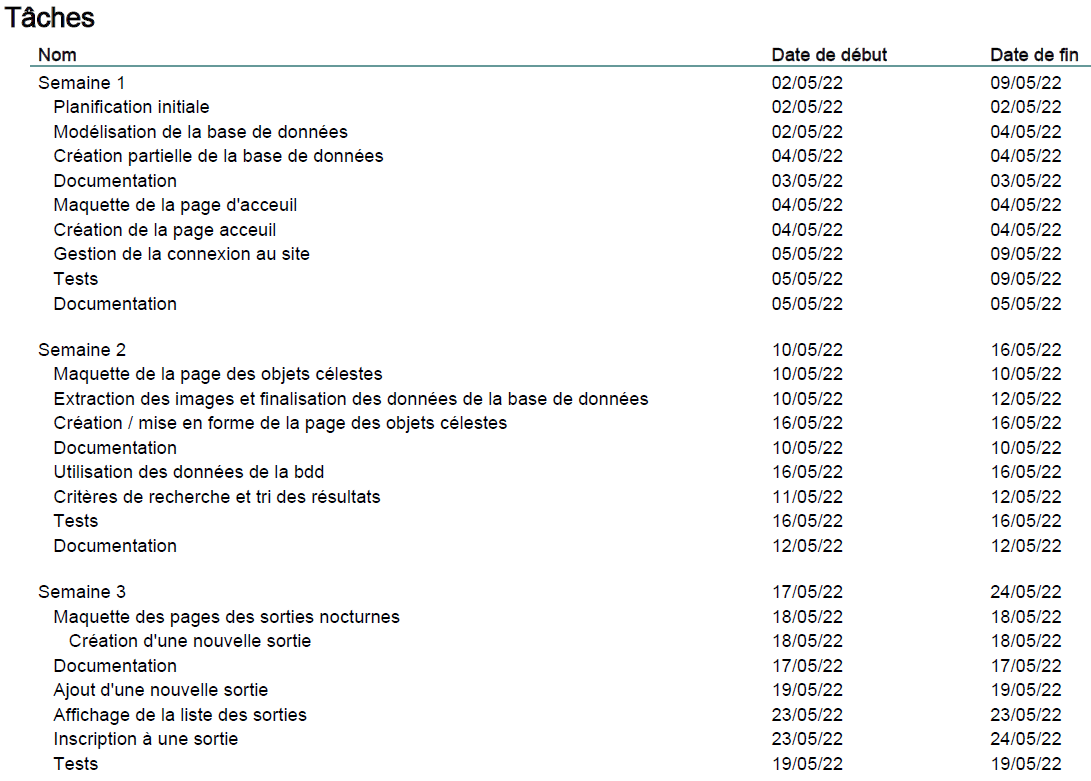
## Risques techniques

Les risques techniques majeurs du projet concernent :

* L’import des données depuis un fichier CSV dans la base de données
* Le téléchargement en local d’images d’un site internet
* L’utilisation d’une base de données PostgreSQL

Le cahier des charges fourni des pistes pour remédier à la première tâche. Quant à la seconde, une discussion durant la phase d’analyse avec la cheffe de projet, Mme Andolfatto, a permis d’apporter des idées de recherches à effectuer. PostgreSQL est quant à lui bien documenté et possède une communauté active. De plus, la conversion d’une base de données MySQL en une base de données PostgreSQL a dû effectuée lors du Pré-TPI.

## Planification



Les principales causes de retard sont liées au risques techniques : l’import des données depuis un fichier CSV et l’import des noms d’images prises depuis un site internet.

## Dossier de conception

### Logiciels utilisés

Ce projet est réalisé sur un ordinateur Windows 10 avec l’aide des logiciels suivants :

* draw.io v15.3.5
* pgAdmin 4 v6.8
* PyCharm 2021.3.3 (Professional Edition)
* Navigateurs web :
  + Firefox
  + Google Chrome
  + Opera

Le logiciel draw.io a été utilisé afin de concevoir les maquettes du site web, le MCD de la base de données ainsi que la balise du MLD car pgAdmin ne permet pas d’en ajouter une.

PgAdmin est le client utilisé pour la base de données PostgreSQL. Il est fourni par défaut quand on télécharge PostgreSQL sur le [site officiel](https://www.postgresql.org/). Il permet via un outil de créer une base de données à partir d’un MLD.

PyCharm a été utilisé pour toute la partie développement du site, que ce soit le code HTML/CSS ou Python. Les différents navigateurs web ont servi lors des tests.

Les navigateurs web ont servi lors des tests des différentes pages du site.

### Base de données

Les données du catalogue de Messier, une fois importées, ont peu de chances de devoir être modifiées. Après que la base de données aura été créée, un fichier de restauration sera généré ainsi la base de données pourra être facilement recréée sur une autre machine.

*Table administrators*

Table sans lien avec les autres car elle contient les adresses mail et les mots de passe des administrateurs du site. Ces informations sont nécessaires lorsqu’ils doivent se connecter pour gérer le site. Un administrateur a été créé pour utiliser toutes les fonctionnalités du site :

|  |  |
| --- | --- |
| **Email** | **Mot de passe** |
| admin@mail.com | Pa$$w0rd |

*Table celestial\_objects*

Table regroupant les informations des objets célestes du catalogue de Messier. Les données utilisées proviennent du fichier CSV disponible sur le site [starlust](https://starlust.org/fr/le-catalogue-messier/).

*Tables constellations, observation\_difficulties, seasons et types*

Ces tables décrivent les entrées de la table *celestial\_objects*. Ces différents éléments ont été placés dans des tables séparées car ils se répètent sur les diverses entrées de la table *celestial\_objects*.

*Table outings*

La table *outings* est destinée à recevoir les données du formulaire permettant d’ajouter de nouvelles sorties nocturnes Ce formulaire peut être rempli par les administrateurs lorsqu’ils sont connecté au site.

*Table users*

La table *users* permet d’enregistrer les données des personnes qui se sont inscrites à une sortie nocturne. Les utilisateurs n’ont pas la possibilité de créer un compte donc à chaque nouvelle inscription d’un/e même utilisateur/trice ses données seront comparées avec celles présentes dans la base de données afin de voir si cette personne est déjà dans la base de données. Si c’est le cas le seul changement qui sera apporté à la base de données concernera les entrées dans la table de jonctions *user\_has\_outing*. Les numéros de téléphones de ces derniers sont supposés être au format suisse.

*Table de jointure user\_has\_outing et outing\_has\_celestial\_object*

La table *user\_has\_outing* sert à définir la relation entre les utilisateurs et les sorties auxquelles ils se sont inscrits.

La table *outing\_has\_celestial\_object* fait le lien entre *outings* et *celestial\_objects* car le formulaire qui concerne les sorties permettra de préciser, ou non, les objets célestes qui vont être observés.

### Site Web

**Librairies tierces utilisées**

Flask – Framework web permettant de concevoir un site web en python, propose des fonctions minimalistes contrairement à son concurrent Django. Son minimalisme permet néanmoins une mise en place rapide d’un site. Ce framework utilise d’autres librairies externes, notamment Jinja2 et Werkzeug. Le premier permet de rendre plus dynamique la manipulation des fichiers HTML du site via des fonctionnalités de templating intéressantes qui permettent de ne pas se répéter. Werkzeug est un Web Server Gateway Interface (WSGI), soit une interface qui permet à l’application Flask de simuler sa présence dans un serveur web local afin de procéder à des tests avant de déployer l’application sur internet.

Flask-SQLAlchemy – Extension pour Flask qui a pour but l’utilisation de la librairie SQLAlchemy. Cette libraire est un ORM (Object Relational Mapper) qui permet de générer des requêtes SQL via l’utilisation de classes et de fonction prédéfinies. L’extension Flask permet de faire communiquer les deux notamment lors de la mise en route de l’application. L’utilisation de fonctions permet de ne pas avoir à gérer le risque d’injection SQL car celles-ci s’occupent sécuriser les entrées des utilisateurs utilisées lors des requêtes. En plus d’une base de données PostgreSQL, cette extension fonctionne également les bases de données SQLite, MySQL, Oracle, MS-SQL, Firebird, Sybase.

Psycopg2 – package python qui permet la communication avec une base de données PostgreSQL. Il est utilisé lorsque Flask-SQLAlchemy fait ses requêtes vers la base de données.

**Design**

Afin de respecter au mieux le critère d’un site responsive du cahier des charges, des classes Bootstrap ont été utilisées pour la création du site. En plus de cela deux plugins ont été utilisé : DataTables et Bootstrap-select. Le premier a servi lors de l’affichage dans un tableau des objets célestes du catalogue de Messier. Le plugin Bootstrap-select a permis d’améliorer les barres de recherches avec une liste de choix utilisées dans les formulaires.

**Hébergement**

Le site web est hébergé chez [Heroku](https://www.heroku.com). L’hébergement est fait gratuitement et comprend la mise en ligne de la base de données du site.

# Réalisation

## Dossier de réalisation

### Base de données

Les images des objets du catalogue de Messier, ont été téléchargées via une extension Firefox : [DownThemAll](https://www.downthemall.org/). Ces dernières viennent du site <https://www.messier-objects.com/messier-catalogue/>. Un script PowerShell a été utilisé pour renommer les images importées en local. Les images de la page [Wikipedia](https://en.wikipedia.org/wiki/Messier_object) concernant le catalogue de Messier avaient des noms qui compliquaient le renommage de celles-ci par un script. Le script utilisé a permis de renommer les fichiers et de les placer dans le dossier *assets/* du site web. La valeur des chemins vers ces images a été placée dans le fichier pictures.csv

Pour les données de la table celestial\_objects, un autre script a dû être utilisé : extract-fk-values.ps1. Ce script a pris les valeurs uniques des colonnes constellation, meilleur saison, difficulté d’observation et meilleure saison du fichier CSV pris sur le site [StarLust](https://starlust.org/fr/le-catalogue-messier/) pour les placer dans d’autre fichiers CSV à part.

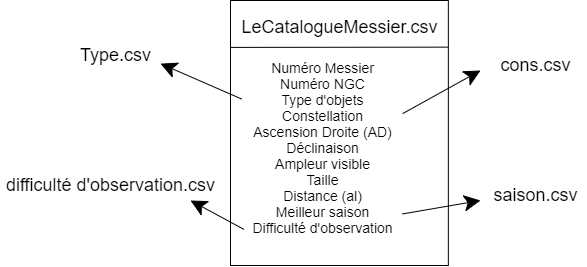


Diagramme 1 - Colonnes exportées vers des fichiers à part

Les fichiers CSV ont d’abord été partiellement importé dans la base de données. Le script SQL « copy-csv.sql » a été créé afin d’inscrire les données des tables sans clés étrangères. Les données entières du CSV ont été placées dans une table intermédiaire *tmp\_celestial\_objects*. Une opération similaire a été faite pour les données du fichier pictures.csv mais ce fichier contient une colonne en plus pour le numéro Messier de l’objet représenté.

Cette dernière action a conclu l’import intégral des données du fichier CSV principal.

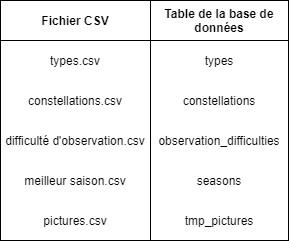
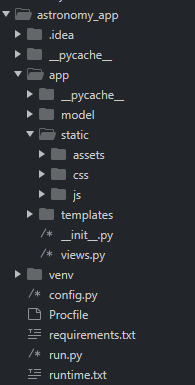


Figure – Fichiers CSV et tables qu’elles ont permis de créer

Le transfert des données depuis les tables temporaires vers la table *celestial\_object*s a été réalisée via un script CSV. Ce script a repris les valeurs des colonnes qui ne contenaient pas de clé étrangère de la table temporaire et les a insérées directement dans la table de destination. Les clés étrangères ont été ajoutées à *celestial\_object*s en comparant les valeurs des colonnes de la table temporaire avec les colonnes ‘noms’ des tables concernées. Finalement les chemins des images ont pu être inscrit via une comparaison du numéro de Messier de l’entrée du tableau des objets célestes à celui de l’entrée de la table *tmp\_pictures.*

### Site web

**Arborescence du site**

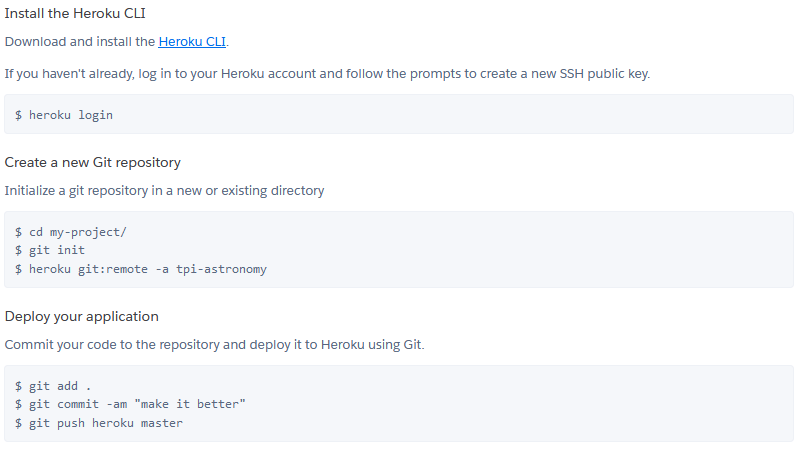


Description du contenu des répertoires :

* Les répertoires commençant par des points (.) et ceux nommés \_\_pycache\_\_ sont créés automatiquement par l’IDE afin de faire fonctionner le programme
* App
  + Model :
    - Package contenant toutes les ressources utilisées pour communiquer avec la base de données
  + Static :
    - Fichiers statiques (images, fichiers CSS et Javascript)
  + Templates :
    - Contient tous les fichiers HTML
* Config.py :
  + Contient toutes les variables de configuration de l’application
* Run.py :
  + Fichier qui permet de lancer l’application
* Procfile et runtime.txt :
  + Fichiers de config utilisés par Heroku pour configurer la machine sur laquelle va être déployée l’application

**Mise en ligne**

Après avoir créé un compte Heroku et ajouté une application au profil, suivre la procédure de mise en ligne proposée par le [site](https://devcenter.heroku.com/articles/git) :



Suite à cela ajouter l’extension [Heroku Postgres](https://elements.heroku.com/addons/heroku-postgresql) à l’application.

Pour mettre en ligne la base de données faire un fichier SQL de backup et l’upload sur le site. Voici la commande utilisée :

heroku pg:push -d postgresql-animated-09754 -a tpi-astronomy < C:\tpi\data\db\_astronomy.sql

Cette commande prend le fichier backup db\_astronomy.sql et l’envoi sur la base de données allouée par Heroku : postgres-animated-09754 qui est liée au projet nommé tpi-astronomy.

Le site est accessible via le lien suivant : <https://tpi-astronomy.herokuapp.com/>.

**Apparence du site**

Les pages créées en se basant sur des maquettes n’ont pas de différences notables avec la conception de base.

## Description des tests effectués

Les tests d’acceptation ont été rédigés sur [IceScrum](https://icescrum.cpnv.ch/p/TPIASTRONO/#/backlog/all). Voici les résultats de ces derniers avec une description des résultats

**Affichage de la page d’accueil**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome et Opera |
| *Résultat attendu* | La page d’accueil s’affiche sans problèmes de mise en forme |
| *Résultat obtenu* | Affichage correct |
| *Date* | 05.05.22 |

**Login – Identifiants corrects**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox |
| *Résultat attendu* | La page d’accueil s’affiche avec un message qui confirme la réussite de l’opération |
| *Résultat obtenu* | Similaire au résultat attendu |
| *Date* | 19.05.22 |

**Login – Identifiants incorrects**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome et Opera |
| *Résultat attendu* | La page de login se recharge et un message d’erreur indique qu’il y a une erreur |
| *Résultat obtenu* | Message d’erreur vague pour mieux préserver la sécurité du site |
| *Date* | 10.05.22 |

**Login – Un ou plusieurs champ vide**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome et Opera |
| *Résultat attendu* | La page de login se recharge et un / plusieurs message/s concernant le / les champs vide/s s’affiche/nt |
| *Résultat obtenu* | Les messages s’affichent correctement selon le champ testé |
| *Date* | 10.05.22 |

**Affichage de la page du catalogue de Messier**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome |
| *Résultat attendu* | La page avec le tableau des objets célestes s’affiche |
| *Résultat obtenu* | La page s’affiche correctement |
| *Date* | 17.05.22 |

|  |  |
| --- | --- |
| *Qui* | Henry Burgat |
| *Environnement* | Firefox |
| *Résultat attendu* | La page avec le tableau des objets célestes s’affiche |
| *Résultat obtenu* | La page s’affiche correctement, amélioration : revenir en haut quand on change de page dans le tableau |
| *Date* | 18.05.22 |

**Tri des éléments du catalogue de Messier selon les colonnes**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome, Opera |
| *Résultat attendu* | Les éléments du tableau des objets célestes peuvent être triés selon le type de la colonne |
| *Résultat obtenu* | Les éléments peuvent être trié, amélioration ajouter une colonne avec l’id dans la base de données pour remettre dans l’ordre de base |
| *Date* | 17.05.22 |

**Ajout d’une nouvelle sortie - Réussite**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome, Opera |
| *Résultat attendu* | Quand les champs du formulaire sont correctement remplis, la sortie s’ajoute à la liste des sorties proposées sur le site |
| *Résultat obtenu* | Réussite |
| *Date* | 24.05.22 |

**Affichage de la liste des sorties**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome, Opera |
| *Résultat attendu* | La liste complète des sorties présentes dans la base de données s’affiche |
| *Résultat obtenu* | Réussite, selon les champs renseignées lors de l’ajout certaines colonnes prennent trop d’espace |
| *Date* | 24.05.22 |

**Inscription à une sortie - Réussite**

|  |  |
| --- | --- |
| *Qui* | Nithujan Jegatheeswaran |
| *Environnement* | Firefox, Chrome, Opera |
| *Résultat attendu* | Quand le formulaire pour s’inscrire à une sortie est envoyée, la base de données ajout une entrée dans la table de jointure user\_has\_outing et une autre dans la table user si ce/cette dernière ne s’est jamais inscrit/e auparavant |
| *Résultat obtenu* | Réussite |
| *Date* | 24.05.22 |

## Erreurs restantes

Les cookies sont utilisés pour permettre à un/e utilisateur/trice d’envoyer les formulaires qu’il/elle a remplis. Mais dans le cas où l’utilisateur/trice a désactivé les cookies sur tous les sites via son navigateur, il sera impossible d’utiliser les formulaires du site. La seconde erreur concerne la mise en forme des tableaux qui a été faite de manière simple ce qui engendre des colonnes parfois disproportionnées, notamment sur la page avec la liste des sorties.

Ces différentes erreurs peuvent impactées négativement l’expérience utilisateur du site. Bien qu’elles ne concernent pas les fonctionnalités proposées par le site, un site web doit avoir une gestion des sessions et un affichage irréprochable.

En premier lieu, il serait possible de corriger le problème de la mise en forme du tableau en modifiant le fichier CSS qui a été importé directement dans l’arborescence du site pour permettre ce genre d’opérations. Concernant le problème des cookies, aucune solution fiable n’a pu être trouvée durant le temps mis à disposition

## Liste des documents fournis

Voici la liste exhaustive des documents fournis :

* Le présent rapport de projet
* Le manuel d’installation (fichier readme.md du Github)

# Conclusions

Tous les objectifs du cahier des charges ont pu être atteint malgré certains points qui ont pris plus de temps que prévu initialement. Certains points peuvent tout de même être remis en question.

La base de données aurait pu être faite en y ajoutant des tables à part entière pour les équipements et les lieux de rendez-vous mais des champs de textes ont été privilégiés pour permettre une plus grande liberté. Une liste de choix pour ces éléments aurait permis de minimiser le problème de mise en forme du tableau avec la liste des sorties.

Peu de tests de formulaires soumis avec des erreurs ont été effectués. Les expressions régulières et la gestion des erreurs lors de l’analyse des données des formulaires ont permis d’accorder moins d’importance au cas limites lors des tests. Du retard a été pris lors de l’import des éléments dans la base de données depuis un fichier CSV.

D’abord l’unicité des éléments de la table *users* a été définie via l’adresse email mais suite à une discussion avec la cheffe de projet, en plus de l’adresse mail, les colonnes nom, prénom et âge ont été choisie pour garantir des entrées uniques.

La principale difficulté de ce projet concerne l’import des données depuis un fichier CSV unique. Ce problème a dû être réglé via l’utilisation de script PowerShell et Batch qui ont permis d’automatiser au maximum l’opération.

Parmi les possibles améliorations qui peuvent être apportées au projet, la première concerne la mise en forme du tableau affichant la liste des sorties.

# Annexes

## Résumé du rapport TPI

Ce TPI est réalisé du 02.05.22 au 25.05.22 à l’aide d’une planification en agile. Son sujet est la création d’un site web qui a pour thème l’astronomie. Ce site doit posséder : une page d’accueil qui explique les objectifs du site, une page présentant les objets célestes du catalogue de Messier et finalement la possibilité de s’inscrire à des sorties nocturnes qui auront été ajoutées au site par les administrateurs de ce dernier.

Afin de pouvoir garantir le stockage des données persistantes, une base de

Données PosteSQL est utilisée. Cette base de données a pour but de contenir la liste complète des objets célestes et tous les champs qui les définissent : les saisons, difficultés d’observation, type d’objet et les constellations auxquelles ils appartiennent. Les données ont été importés via un fichier CSV et des scripts. Le projet est réalisé en Python. Les packages choisies sont Flask, Flask-SQLAlchemy, Flask-WTF et psycopg2. Flask est un framework qui permet développer des sites web. Flask-SQLAlchemy est une

Lextension Flask permet l’utilisation de la librairie SQLAlchemy. Cette dernière permet de générer des requêtes SQL pour la base de donnnées. est utilisé par Flask-SQLAlchemy pour communiquer avec la base de données.

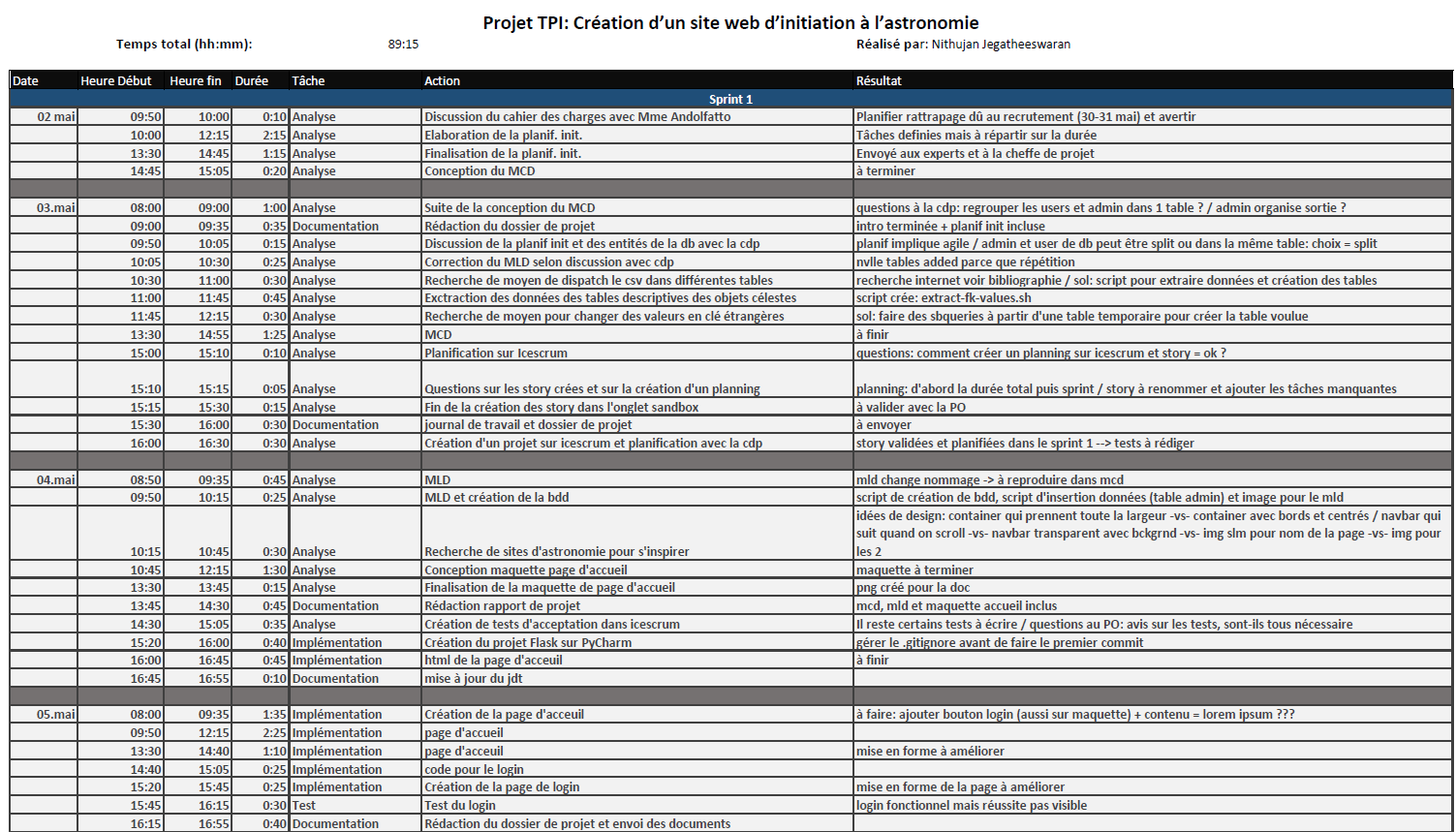
Les principales fonctionnalités de l’application ont pu être implémentés et testées.

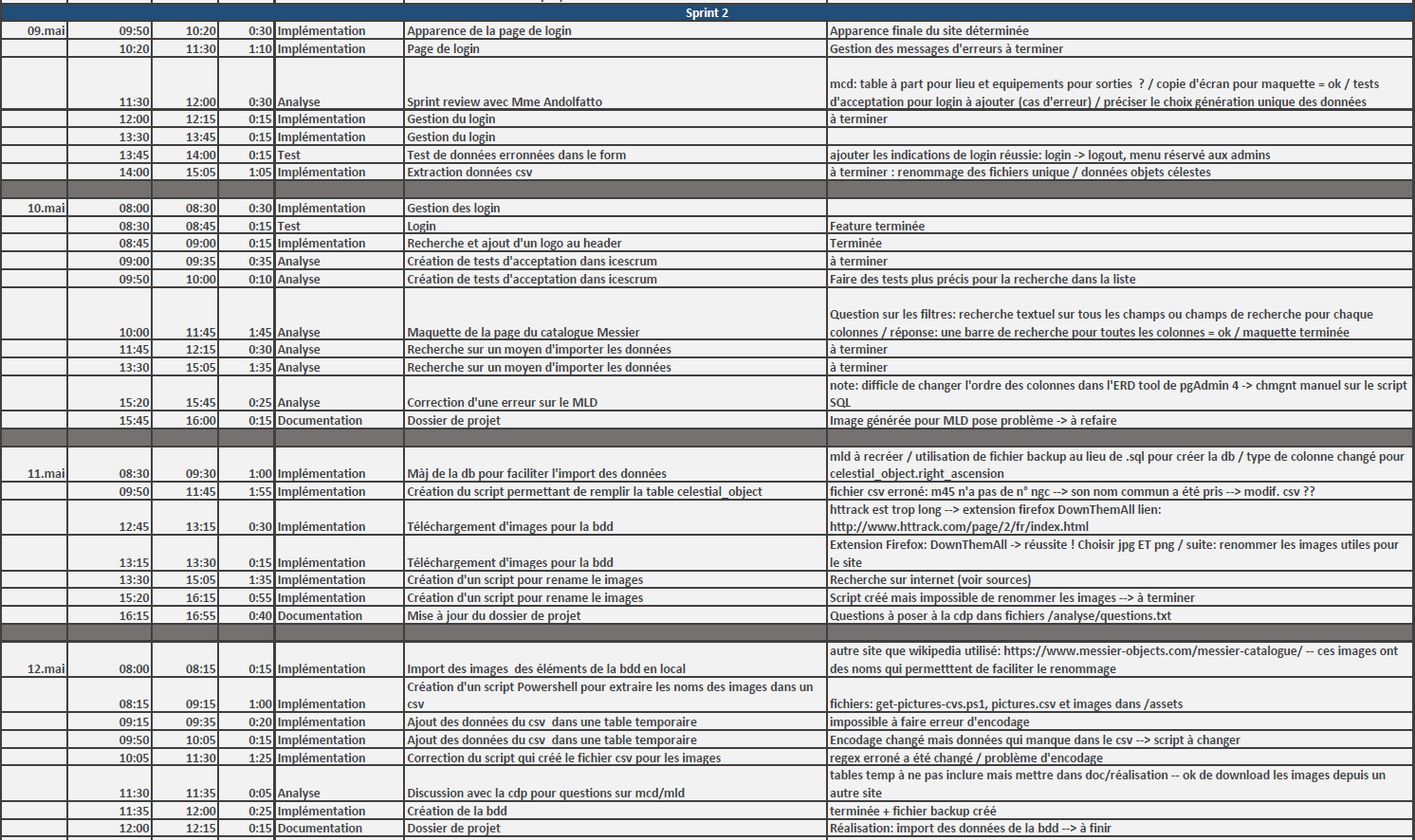
L’utilisation de classes bootstrap et de deux plugins a permis de créer des pages responsives. Les tableaux utilisés pour le site doivent être revus afin de permettre une meilleure adaptation à certaines données. Le site a pu être mis en ligne chez un hébergeur, Heroku, à l’adresse suivante : <https://tpi-astronomy.herokuapp.com/>

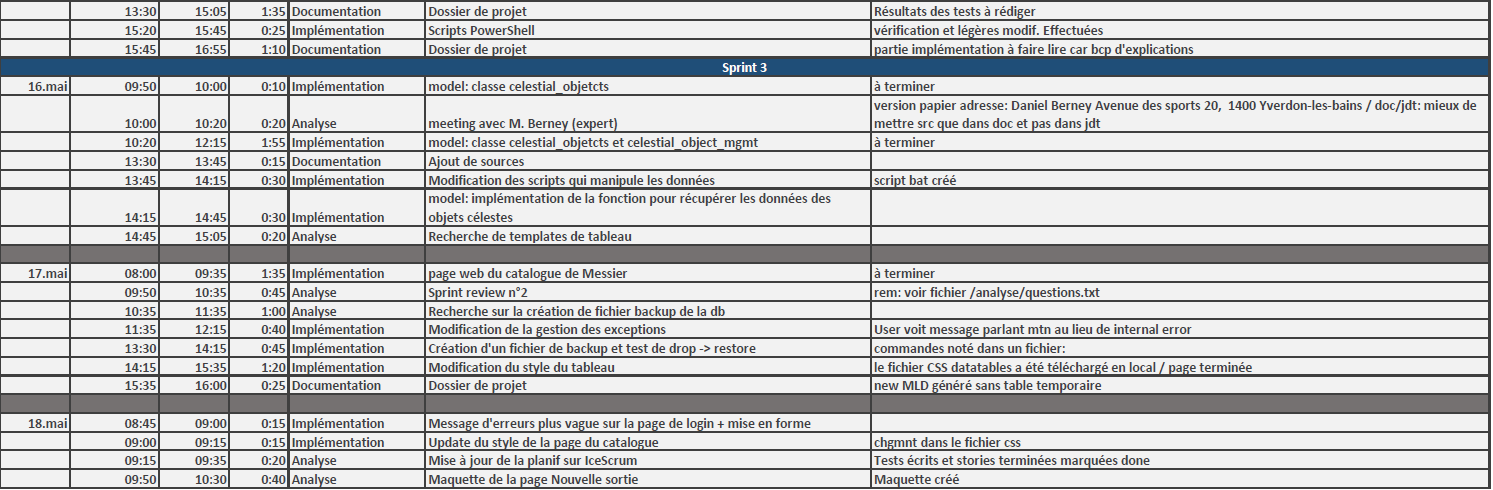
## Sources – Bibliographie

|  |  |
| --- | --- |
| **Date** | **Source** |
| 03.05 |  |
|  | <https://dba.stackexchange.com/questions/86349/postgresql-csv-to-db-translate-foreign-keys> |
|  | <https://dataschool.com/learn-sql/importing-data-from-csv-in-postgresql/> |
|  | <https://stackoverflow.com/questions/61841200/i-would-like-to-select-only-unique-items-from-a-csv-file-in-powershell> |
|  | <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/export-csv?view=powershell-7.2> |
|  | <https://stackoverflow.com/questions/19450616/export-csv-exports-length-but-not-name> |
|  | <https://www.catapultsystems.com/blogs/how-to-export-and-import-foreign-characters-with-powershell/> |
|  | <https://stackoverflow.com/questions/32459269/exporting-a-variable-to-a-csv-file> |
|  | <https://dba.stackexchange.com/questions/130105/convert-a-lookup-value-to-foreign-key-when-a-query-is-run> |
|  | <https://www.postgresql.org/docs/current/sql-copy.html> |
|  | <https://dba.stackexchange.com/questions/130105/convert-a-lookup-value-to-foreign-key-when-a-query-is-run> |
|  | <https://stackoverflow.com/questions/12060546/remove-timestamp-precision-from-now-result-in-postgresql> |
|  | <https://www.postgresql.org/docs/current/datatype-numeric.html#DATATYPE-NUMERIC-DECIMAL> |
|  | <https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es_relationnel-objet> |
| 04.05 |  |
|  | <https://www.postgresql.org/docs/current/datatype-datetime.html> |
|  | <https://www.astronomes.com/> |
|  | <https://www.afastronomie.fr/manifestations> |
|  | <http://www.planete-astronomie.com/> |
|  | <https://hubblesite.org/science/solar-system> |
|  | <https://www.nasa.gov/topics/humans-in-space> |
|  | <https://astronomynow.com/magazine/> |
|  | <https://mdbootstrap.com/how-to/bootstrap/navbar-transparent/> |
|  | https://getbootstrap.com/docs/5.1/utilities/text/#text-transform |
|  | https://getbootstrap.com/docs/5.1/components/navbar/ |
|  | <https://www.toptal.com/developers/gitignore> |
| 05.05 |  |
|  | <https://jinja.palletsprojects.com/en/3.1.x/templates/?highlight=macro> |
|  | https://getbootstrap.com/docs/5.1/forms/overview/ |
|  | <https://stock.adobe.com/fr/252441635/> |
|  | <https://www.pexels.com/fr-fr/photo/groupe-d-etoiles-1341279/> |
|  | https://stackoverflow.com/questions/26332319/finding-local-postgresql-database-url |
| 09.05 |  |
|  | https://getbootstrap.com/docs/5.0/utilities/borders/ |
|  | https://www.pexels.com/fr-fr/photo/ciel-etoile-998641/ |
|  | https://getbootstrap.com/docs/5.0/utilities/text/#font-weight-and-italics |
|  | https://fjolt.com/article/css-states |
|  | https://www.w3schools.com/python/python\_try\_except.asp |
|  | https://docs.sqlalchemy.org/en/14/errors.html#error-e3q8 |
| 10.05 |  |
|  | https://flask.palletsprojects.com/en/2.1.x/quickstart/#sessions |
|  | Logo du site : https://www.pexels.com/fr-fr/photo/photo-de-la-lune-47367/ |
|  | Dernière date d’ajout d’un objet au catalogue Messier : <https://en.wikipedia.org/wiki/Messier_110> |
|  | PowerShell - remplacer du texte par un autre : <https://www.tutorialspoint.com/how-to-find-and-replace-the-word-in-a-text-file-using-powershell> |
|  | Template de tableau : <https://datatables.net/examples/styling/bootstrap5> |
|  | Choisir une base de données en psql : <https://www.tutorialspoint.com/postgresql/postgresql_select_database.htm> |
|  | Utilisation de chemin relatif en psql : <https://dba.stackexchange.com/questions/158466/relative-path-for-psql-copy-file> |
|  | Documentation postgres sur le type SERIAL : <https://www.postgresql.org/docs/current/datatype-numeric.html#DATATYPE-SERIAL> |
|  | Copy csv dans la bdd : <https://stackoverflow.com/questions/24483582/copy-only-some-columns-from-an-input-csv> |
| 11.05 |  |
|  | <https://stackoverflow.com/questions/40904836/how-to-get-n-files-in-a-directory-order-by-last-modified-date> |
|  | <https://stackoverflow.com/questions/52152228/how-to-filter-a-list-with-a-list-of-strings-in-powershell> |
|  | <https://stackoverflow.com/questions/11816218/renaming-files-in-powershell-using-the-folder-name> |
|  | <https://stackoverflow.com/questions/51818485/increment-variable-in-powershell-from-within-if-statement-within-a-foreach-loop> |
|  | Fichier pour créer la bdd : <https://stackoverflow.com/questions/7804698/how-to-create-sql-script-of-postgres-database-schema> |
|  | Télécharger de images de site : <https://www.wikihow.com/Download-All-Images-on-a-Web-Page-at-Once> |
| 12.05 |  |
|  | <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_regular_expressions?view=powershell-7.2> |
|  | <https://stackoverflow.com/questions/54882043/powershell-variable-assignment-vs-pipeline> |
|  | <https://stackoverflow.com/questions/27970441/powershell-string-does-not-contain> |
|  | PowerShell aller vers le répertoire du script <https://stackoverflow.com/questions/4724290/powershell-run-command-from-scripts-directory> |
| 16.05 |  |
|  | Types dans SQLAlchemy : <https://docs.sqlalchemy.org/en/14/core/type_basics.html> |
|  | Syntaxe command « copy » psql : <https://stackoverflow.com/questions/48872965/postgres-copy-syntax> |
|  | Chemin relatif pour copy : <https://dba.stackexchange.com/questions/158466/relative-path-for-psql-copy-file> |
|  | Jointure dans Flask-SQLAlchemy : <https://flask-sqlalchemy.palletsprojects.com/en/2.x/models/?highlight=join> |
|  | Template de tableau : <https://www.datatables.net/manual/installation>  <https://datatables.net/manual/styling/classes> |
| 17.05 |  |
|  | jQuery : <https://releases.jquery.com/> |
|  | Commande pg\_dump : <https://www.postgresql.org/docs/14/app-pgdump.html> |
|  | Doc Flask – passer des messages : <https://flask.palletsprojects.com/en/2.1.x/patterns/flashing/> |
|  | Exemple de message passé via Flask : <https://www.tutorialspoint.com/flask/flask_message_flashing.htm> |
|  | Variable d’environnement PostgreSQL : <https://www.postgresql.org/docs/current/libpq-envars.html> |
| 18.05 |  |
|  | Classe Flask-WTF : <https://wtforms.readthedocs.io/en/3.0.x/fields/?highlight=select#wtforms.fields.SelectMultipleField> |
|  | Bootstrap 5 et les formulaires : <https://getbootstrap.com/docs/5.0/forms/input-group/> |
|  | Conventions html : <https://www.w3schools.com/htmL/html5_syntax.asp> |
|  | WTForms - Validateurs de base : <https://wtforms.readthedocs.io/en/3.0.x/validators/> |
|  | Templates HTML pour les selects : <https://csshint.com/free-css-select-boxes-snippets/> |
|  | Bootstrap-select : <https://developer.snapappointments.com/bootstrap-select/>  Examples de champs <select> : <https://developer.snapappointments.com/bootstrap-select/examples/#basic-examples> |
|  | Erreur avec Bootstrap-select : <https://stackoverflow.com/questions/34954785/bootstrap-select-didnt-show-on-page-load> |
| 19.05 |  |
|  | Validation des champs datetime avec WTForms : <https://stackoverflow.com/questions/58293843/wtforms-datetimefield-returning-true-not-matter-input> |
|  | Avoir la date en Python : <https://www.geeksforgeeks.org/get-current-date-and-time-using-python/> |
|  | PostgreSQL format timestamp par défaut : <https://stackoverflow.com/questions/8723574/in-postgres-can-you-set-the-default-formatting-for-a-timestamp-by-session-or-g> |
| 23.05 |  |
|  | regex Python : <https://www.w3schools.com/python/python_regex.asp> |
|  | Déf. SGBDRO: <https://fr.wikipedia.org/wiki/Système_de_gestion_de_base_de_données_relationnel-objet> |
| 24.05 |  |
|  | Définition de WSGI : <https://wsgi.readthedocs.io/en/latest/what.html> |
|  | Définition d’ORM : <https://www.educba.com/what-is-orm/> |
|  | Déf. Modules, packages, librairie, framework : <https://learnpython.com/blog/python-modules-packages-libraries-frameworks/> |
|  | Rendre les retours à la ligne : <https://stackoverflow.com/questions/42547885/how-to-recognize-new-line-character-n-in-html> |
| 25.05 |  |
|  | Doc HTML : <https://www.sphinx-doc.org/en/master/tutorial/getting-started.html> / <https://stackoverflow.com/questions/13333190/sphinx-generates-empty-documentation> |
|  | Problème de cookies :   * <https://stackoverflow.com/questions/46135554/flask-login-session-without-cookies> * <https://security.stackexchange.com/questions/95404/how-can-i-uniquely-identify-an-user-when-cookies-are-not-an-option> * <https://stackoverflow.com/questions/15966812/user-recognition-without-cookies-or-local-storage> * <https://stackoverflow.com/questions/71181279/how-can-i-use-flask-wtforms-and-csrf-without-session-cookie> |

## Journal de travail









# Glossaire

B

Bootstrap

Collection d'outil permettant de mettre en forme des pages web sans devoir créer du code CSS 18, 29, 30

C

clé étrangère

Dans une base de données relationnel, référence d'une entrée d'une table dans une autre entrée de la même ou d'une autre table 20

CSS

Cascading Style Sheets, code utilisé pour concevoir la mise en forme des pages web 16

CSV

Comma Separated Values, un fichier contenant des données le plus souvent séparées par des virgules 3, 14, 16, 19, 20

F

Framework

Ensemble d'éléments liés à la programmation qui permettent de mettre en place l'architecture d'un logiciel 17

H

HTML

HyperText Markup Language, language utilisé pour coder les pages web 16, 29

I

IDE

Integrated Development Environment, logiciel qui fournit un environnemnt permettant de créer des programmes 21

L

librairie

Groupe de fichiers qui contiennent des ressources pouvant être utilisées pour créer des logiciels 17

M

**MCD**

Modèle Conceptuel de Données, représentation des données d'un système d'information qui est facilement compréhensible 6, 16

**MLD**

Modèle Logique de Données, représentation de la structure de données d'un système d'information faite en précisant le type des données utilisé par ce système 7, 16

O

ORM

Object Relational Mapper, permet de communiquer avec une base de données via l'utilisation d'objets de programmation 17

P

package

Ensemble de fichiers contenant des ressources de programmation Python (modules) 18

plugin

Logiciel ou composant logiciel destiné à être ajouté à un autre logiciel afin d'apporter de nouvelles fonctionnalités 18

PowerShell

Language permettant d'automatiser des tâches via des scripts 19, 28

S

SGBDRO

Système de Gestion de Base de Données Relationnel-Objet, base de données où les données sont représentées sous forme d'objets comme dans la programmation orientée objet 3

SQL

Structured Query Language, langage permettant de communiquer avec des bases de données 19

W

WSGI

Web Server Gateway Interface, interface qui gère la communication entre un serveur web et une application web 17

1. SGBRDO : Système de Gestion de Base de Donnée Relationnelle-Objet, base de données où les données sont représentées sous forme d'objets comme dans la programmation orientée objet [↑](#footnote-ref-1)
2. CSV: Comma Separated Values, un fichier contenant des données le plus souvent séparées par des virgules [↑](#footnote-ref-2)