

Mise en place de solutions de gestion et développement d'une application web

Rapport de stage effectué du 7 janvier au 14 février 2025

Imaho Timothée
BTS SIO – option SLAM

Établissement : Institution Saint Aspais



Tuteur de stage : ABGO Wilfried

Entreprise d'accueil : Heaven Flow Group, 50 Avenue Maurice Bourges
Maunoury, 31200 Toulouse

Sommaire

<u>INTRODUCTION</u>	3
<u>I / Étude de l'entreprise et son environnement</u>	4
<u>A - Présentation de l'association Heaven Flow Group</u>	4
<u>B - Activités principales</u>	5
<u>II / Cahier des charges</u>	6
<u>A – Description des objectifs du stage</u>	6
<u>III / Travail accompli</u>	8
<u>A – Présentation des tâches effectuées</u>	8
<u>1. Tâches les plus pertinentes</u>	8
<u>2. Problèmes rencontrés et résolution</u>	10
<u>B – Présentation d'une journée type</u>	11
<u>CONCLUSION</u>	12

Introduction

My internship journey was divided into two key phases. Initially, I focused on conducting a thorough benchmark¹ to evaluate various ERP² and CRM³ solutions, ultimately installing the most suitable ones tailored to the association's needs. This phase involved meticulous documentation to ensure smooth implementation and future maintenance, providing a solid foundation for the association's operations.

In the second phase, I had the opportunity to develop a web application using the [RedwoodJS](#) framework⁴ and the React library⁵. This application was designed to simplify the reservation process for communal halls, aiming to make the booking experience more accessible and user-friendly for the community.

I will start by sharing the context of the institution and its environment, providing a clear picture of the setting where my internship took place. Next, I will explore the goals of my internship, focusing on what my supervisor hoped to achieve through my contributions. Finally, I will discuss the tasks I undertook, highlighting the technical skills I honed and the personal insights I gained during this rewarding experience.

¹ Benchmarking is the process of comparing different solutions to identify the best fit for an organization's needs.

² ERP (Enterprise Resource Planning) is a software system that allows to manage and integrate the main activities of an organization (finance, logistics, human resources).

³ CRM (Customer Relationship Management) is a tool that allows you to centralize, organize and optimize interactions between external parties (beneficiaries, members, or customers).

⁴ A framework is a structured foundation of pre-written code that provides a skeleton for building applications. It dictates the architecture and offers tools and features to streamline the development process.

⁵ A library is a collection of pre-written code that developers can use to perform specific tasks. It provides functionalities that can be integrated into an application without dictating its overall structure.

I / Étude de l'entreprise et son environnement

A - Présentation de l'association Heaven Flow Group

L'association Heaven Flow Group est une organisation dynamique basée à Toulouse, dédiée à la promotion du gospel et des danses urbaines. Leur mission est de partager ces arts culturels avec un large public, en organisant divers événements et ateliers. Le département qui m'a accueilli est celui des NTICs (Nouvelles Technologies de l'Information et de la Communication), responsable de la gestion et de l'optimisation des infrastructures technologiques de l'association.

L'environnement informatique de Heaven Flow Group est principalement basé sur des solutions cloud et à distance, facilitant ainsi la collaboration et l'accès aux ressources depuis différents emplacements. L'association utilise des services d'hébergement et de cloud computing tels que O2Switch et IONOS ou encore AWS (Amazon Web Services), pour garantir la disponibilité et la performance de ses applications et services en ligne. Cette architecture permet une grande flexibilité et une capacité d'adaptation aux besoins évolutifs de l'association.

B - Activités principales

Heaven Flow Group se distingue par son engagement envers la promotion du gospel et des danses urbaines. L'association organise régulièrement des concerts, des ateliers de danse, et des événements communautaires visant à sensibiliser et à inspirer le public. Ces activités sont soutenues par une infrastructure technologique robuste, permettant une gestion efficace des événements, des inscriptions, et des communications avec les membres et les participants.

Le département NTICs joue un rôle crucial dans le soutien de ces activités en assurant le bon fonctionnement des systèmes informatiques et en développant des solutions technologiques adaptées aux besoins spécifiques de l'association. L'utilisation de technologies cloud permet une gestion centralisée et sécurisée des données, tout en offrant une accessibilité accrue pour les membres et les bénévoles.

II / Cahier des charges

A – Description des objectifs du stage

L'objectif principal de mon stage a été de contribuer à l'optimisation des infrastructures technologiques de l'association Heaven Flow Group en sélectionnant et mettant en place des solutions ERP* et CRM* adaptées, puis en développant une application web pour la gestion des réservations de salles communautaires.

Pour assurer un suivi efficace de mes activités, j'ai utilisé [Trello](#), un outil de gestion de projet, pour organiser et prioriser mes tâches. Cela m'a permis de suivre la progression de chaque activité, de définir des objectifs clairs et de m'assurer que chaque étape soit accomplie dans les délais impartis. J'ai également utilisé [Git](#) pour la gestion de version du code, ce qui m'a permis de suivre facilement les modifications, de collaborer efficacement et de revenir à une version antérieure du code en cas de problème.

La première activité a été de réaliser un benchmark* pour comparer différentes solutions ERP et CRM open source. Cette étape a permis de déterminer les outils les plus pertinents pour l'association en fonction de critères tels que la flexibilité, l'accessibilité et la compatibilité avec les besoins de gestion des événements et des membres.

Une fois les outils choisis, la phase suivante consistait à installer ces solutions sur un serveur de test, en veillant à ce qu'elles répondent aux exigences de performance et de sécurité. La documentation complète de l'installation a été une étape essentielle pour assurer un déploiement et maintenance sans problèmes.

Dans la troisième activité, j'ai mis en place un serveur [AWS](#) (Amazon Web Services), avec une instance EC2¹, pour héberger les outils et garantir une infrastructure évolutive et sécurisée. Cela a permis de tester les outils dans un environnement de production simulé et d'assurer une disponibilité optimale.

Ensuite, la quatrième activité a impliqué la conception du design de l'application web avec [Figma](#). Une maquette a été réalisée pour visualiser l'interface utilisateur et s'assurer que l'application soit intuitive et adaptée aux utilisateurs finaux.

La dernière activité a été consacrée au développement de l'application web, avec un calendrier interactif pour gérer les réservations de salles. Ce calendrier permet de s'informer sur les salles, de visualiser les créneaux disponibles et de mettre en place des réservations. Pour la gestion du calendrier, j'ai utilisé [MongoDB](#) comme base de données et MongoDB Atlas pour le stockage et la gestion des données dans le cloud.

* voir page 3 pour la définition

¹ Amazon EC2 est un service AWS permettant de créer et gérer des serveurs virtuels (instances) pour héberger des applications, sites web ou traiter des données, avec la possibilité de choisir la puissance et le système d'exploitation.

III / Travail accompli

A – Présentation des tâches effectuées

1. Tâches les plus pertinentes

Au cours de mon stage, j'ai mené plusieurs missions clés pour répondre aux besoins spécifiques de l'association, en me concentrant sur l'optimisation des processus à travers la mise en place de solutions adaptées.

La première étape de mon travail a consisté à réaliser un benchmark des solutions ERP et CRM open source. J'ai comparé différentes options en fonction de leurs fonctionnalités, avantages, inconvénients, langages et bases de données avec les besoins spécifiques de l'association. Pour ce faire, j'ai créé une grille de critères d'évaluation qui m'a permis d'analyser chaque solution de manière structurée et objective. Après cette analyse, j'ai sélectionné les outils les plus adaptés à l'association, et j'ai rédigé un rapport détaillant les raisons de ces choix.

Ensuite, j'ai procédé à l'installation des ERP et CRM sur le serveur de l'association. Ce processus a impliqué la configuration de l'environnement serveur, l'intégration des bases de données nécessaires, et la mise en place des outils de manière fonctionnelle. Afin de garantir la bonne mise en place de l'outil, j'ai rédigé une documentation d'installation complète. Cette documentation détaillait chaque étape du processus pour faciliter la gestion des installations futures et les maintenances.

Une autre étape clé a été la mise en place d'un serveur [AWS](#) (EC2) pour héberger les outils ERP et CRM dans un environnement cloud flexible et sécurisé. J'ai configuré une instance EC2 et paramétré les règles de sécurité, afin d'assurer à la fois une performance optimale et une protection des

données. La gestion des groupes de sécurité et des logs d'accès a été essentielle pour garantir un environnement de travail sécurisé et performant.

Par la suite, j'ai conçu et développé une application web pour gérer la réservations de salles. Après avoir créé des maquettes détaillant l'interface utilisateur de l'application avec [Figma](#), j'ai utilisé le framework [RedwoodJS](#) et la bibliothèque React pour développer l'application. J'ai utilisé MongoDB comme base de données et [MongoDB Atlas](#) comme service cloud. L'intégration avec Prisma a permis de structurer et de gérer efficacement les données, en assurant une réponse rapide lors de la vérification des disponibilités.

L'interface permet aux utilisateurs de consulter les disponibilités des salles et de réserver des créneaux horaires en toute simplicité. L'un des principaux défis à ce stade a été de gérer les conflits de réservation. Pour cela, j'ai mis en place une logique côté serveur permettant de vérifier en temps réel la disponibilité des créneaux avant d'accepter une réservation.

Enfin, après avoir terminé le développement de l'application, j'ai effectué plusieurs séries de tests. J'ai réalisé des tests unitaires pour valider le bon fonctionnement des différentes fonctionnalités, ainsi que des tests d'intégration pour vérifier l'interaction des différentes parties de l'application. J'ai également mis en place des tests utilisateurs pour m'assurer de la convivialité de l'interface.

Tout au long de ces tâches, j'ai dû faire preuve de rigueur technique, de patience, et d'une capacité à résoudre des problèmes de manière autonome, tout en collaborant étroitement avec mes collègues pour répondre efficacement aux besoins de l'association.

2. Problèmes rencontrés et résolution

1. Problème de compatibilité des outils ERP/CRM avec les dépendances serveur

Lors de l'installation des ERP et CRM, j'ai rencontré des incompatibilités entre certaines versions des outils et les dépendances serveur nécessaires. Certaines configurations requises par les outils n'étaient pas compatibles avec la version du serveur en place, ce qui a bloqué l'installation initiale.

Solution :

Pour résoudre ce problème, j'ai procédé à une mise à jour des composants du serveur et vérifié les versions des outils ERP et CRM en tenant compte de la compatibilité avec les dépendances. J'ai également consulté des forums de support technique pour trouver des solutions adaptées aux versions spécifiques des outils.

2. Problème de gestion des conflits de réservation dans l'application web

Lors du développement de l'application web de gestion des réservations, un problème majeur est survenu : la gestion des conflits de réservation. Il était possible que plusieurs utilisateurs essaient de réserver le même créneau horaire en même temps, créant des chevauchements et des erreurs de réservation.

Solution :

Pour résoudre ce problème, j'ai intégré une logique côté serveur qui vérifie la disponibilité d'un créneau avant de valider la réservation. Chaque demande de réservation est désormais comparée avec les réservations existantes pour s'assurer qu'il n'y a pas de conflit. Ce système permet de garantir une gestion fluide et sans erreur des créneaux horaires.

B – Présentation d'une journée type

Lors de mon stage, je travaillé en distanciel, ma journée de travail commençait à 9h. À 10h, je participais à une réunion quotidienne en ligne avec l'équipe, qui durait environ 30 minutes, durant laquelle ont partagée nos tâches effectuées le jour précédent et celles à effectuer le jour même ainsi que les points bloquants

Ensuite, je poursuivais mes activités jusqu'à midi, en me concentrant sur la résolution de problèmes techniques, en collaboration avec l'autre stagiaire présent.

Après une pause de 2h pour déjeuner et me reposer, je reprenais mon travail à 14h, où je me focalisais sur les activités de l'après-midi, notamment les tâche annoncer lors de la réunion et la documentation des processus.

Ma journée se terminait vers 17h30, après avoir fait un point sur les avancées et planifié les tâches pour le lendemain.

Conclusion

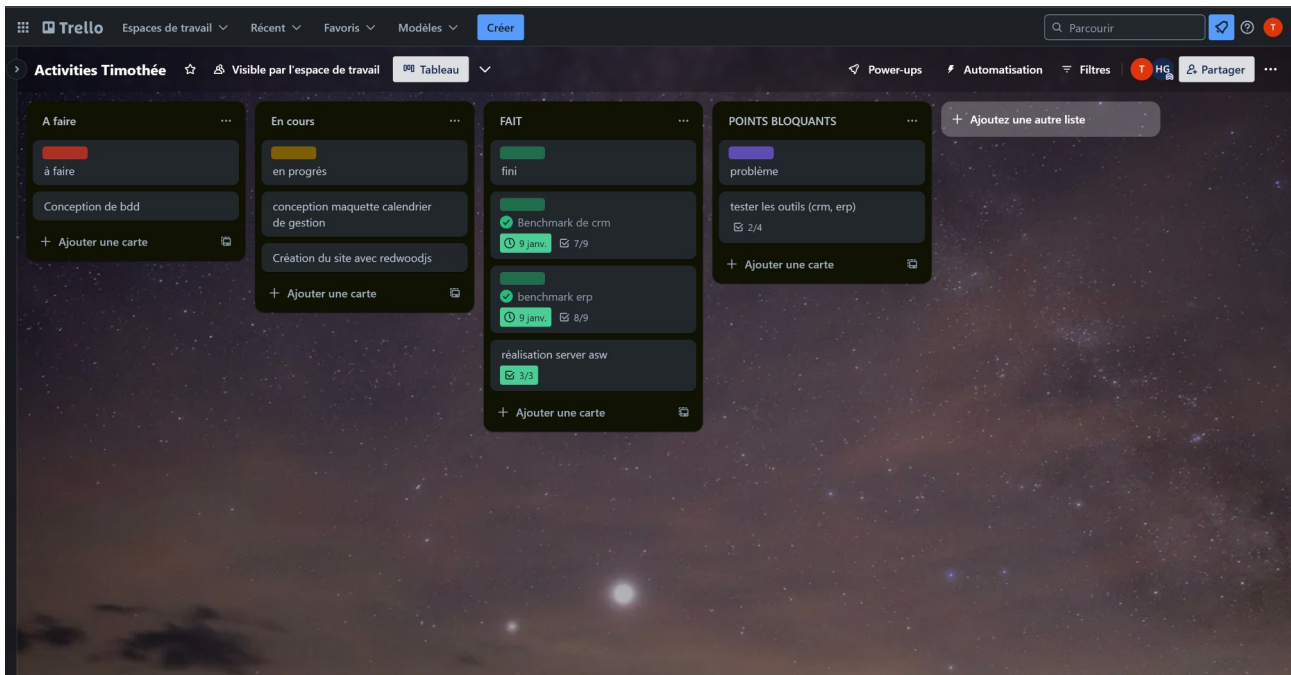
Ce stage a été une expérience profondément enrichissante, tant sur le plan technique que personnel. Il m'a permis non seulement de consolider mes compétences en développement web, administration système et gestion de projet, mais aussi de découvrir l'impact réel de la technologie dans le fonctionnement et l'efficacité d'une organisation.

Au-delà des aspects techniques, cette expérience m'a offert une occasion précieuse de perfectionner mes compétences en collaboration et en communication au sein d'une équipe. Partager des idées, résoudre des problèmes ensemble et savoir m'adapter à des situations imprévues ont été des moments clés lors de mon stage. De plus, le travail à distance a renforcé mon autonomie, m'obligeant à organiser efficacement mon temps tout en maintenant une bonne communication et cohésion avec mes collègues, ce qui m'a permis de rester productif dans un environnement flexible.

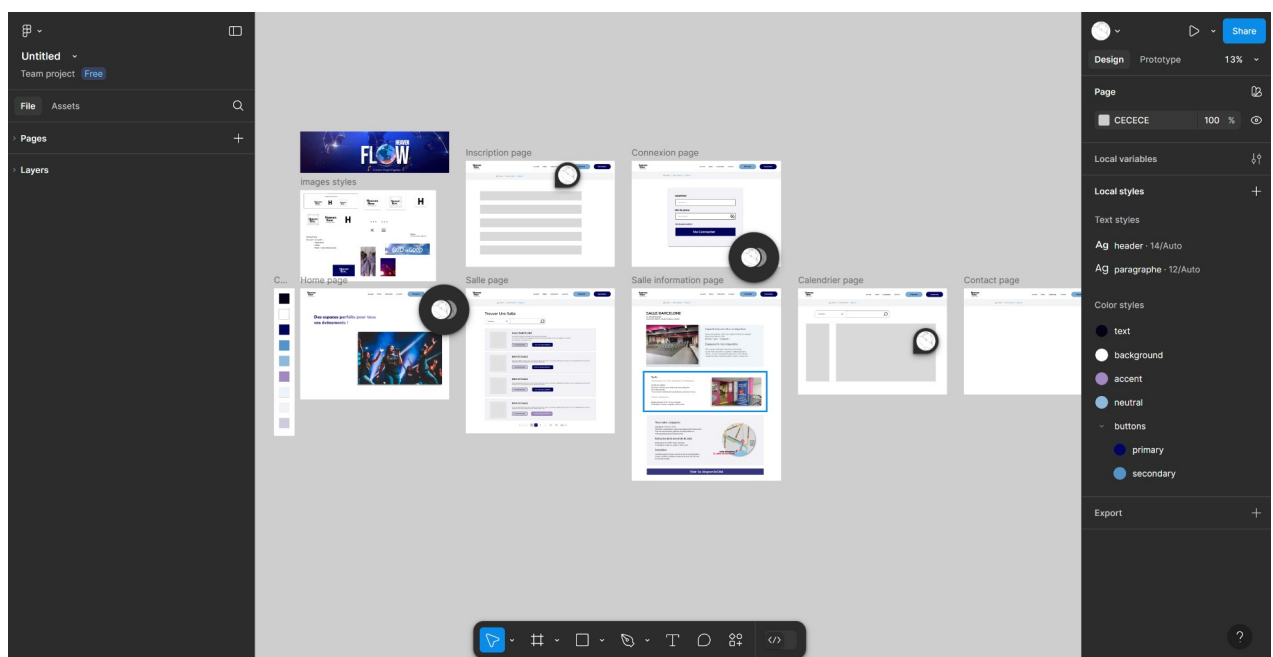
Bien que j'aie acquis de nouvelles compétences techniques et pratiques, j'ai également pris conscience des défis qui se présentent dans ces domaines. Ce stage m'a offert à nouveau une vision plus réaliste de ce que représente ce type de travail en entreprise, notamment la gestion des imprévus et des contraintes techniques.

Annexes

Trello



Figma



Documentation AWS

(juste des exemples)

Amazon Web Service

Amazon Web Services (AWS) est une plateforme de services cloud proposée par Amazon.

Elle permet aux entreprises et aux particuliers d'accéder à des ressources informatiques (comme des serveurs, du stockage, des bases de données, etc.) via Internet au lieu d'avoir à les gérer physiquement. Pour faire simple, c'est comme louer de la puissance informatique au lieu d'acheter et de maintenir des machines coûteuses.



- Service EC2

Amazon EC2 (Elastic Compute Cloud) est l'un des services d'AWS. Il permet de créer et gérer des serveurs virtuels appelés "instances". Ces serveurs peuvent être utilisés pour héberger des applications, des sites web, ou traiter de grandes quantités de données. On peut choisir la puissance et le système d'exploitation, et le serveur fonctionne à distance.



Lancer une instance Informations

Amazon EC2 vous permet de créer des machines virtuelles, ou des instances, qui s'exécutent sur le Cloud AWS. Démarez rapidement en suivant les étapes simples indiquées ci-dessous.

Nom et balises Informations

Nom

AL2023-server1

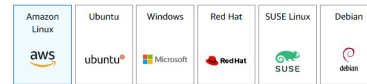
[Ajouter des balises supplémentaires](#)

Images d'applications et de systèmes d'exploitation (Amazon Machine Image) Informations

Une AMI est un modèle contenant la configuration logicielle (système d'exploitation, serveur d'applications et applications) requise pour lancer votre instance. Parcourez ou recherchez des AMI si vous ne trouvez pas ce que vous recherchez ci-dessous.

Effectuer une recherche dans notre catalogue complet, qui comprend des milliers d'images d'applications et de systèmes d'exploitation

Démarrage rapide



Explorer plus d'AMI
Y compris les AMI d'AWS, de Marketplace et de la communauté

Amazon Machine Image (AMI)

AMI Amazon Linux 2023
ami-0f38b927e6597da05 (64 bits x86, uefi-preferred) / ami-09394d73729fa99 (64 bits ARM, uefi)
Virtualisation: hvm ENA activé: true Type de périphérique racine: ebs

Éligible à l'offre gratuite

Nom du groupe de sécurité - obligatoire

launch-wizard-1

Ce groupe de sécurité sera ajouté à toutes les interfaces réseau. Le nom ne peut pas être modifié après la création du groupe de sécurité. La longueur maximale est de 255 caractères. Caractères valides : a-z, A-Z, 0-9, espaces et _ - / () #, @ [] + = & ; ! \$ *

Description - obligatoire Informations

launch-wizard-1 created 2025-01-20T10:54:55.756Z

Règles entrantes des groupes de sécurité

▼ Règle de groupe de sécurité 1 (TCP, 22, 88.124.107.218/32)

Supprimer

Type Informations

ssh

Protocole Informations

TCP

Plage de ports Informations

22

Type de source Informations

Mon IP

Nom Informations

Ajouter une adresse CIDR, une liste de pré

2 X

Description - facultatif Informations

par exemple, SSH pour le bureau de l'administ

Ajouter une règle de groupe de sécurité

► Configuration réseau avancée

Instances (1/1) Informations

Date de la dernière mise à jour
Il y a 1 minute

Se connecter

État de l'instance

Actions

Lancer des instances

Rechercher Instance par attribut ou identification (case-sensitive)

Tous les é...

< 1 >

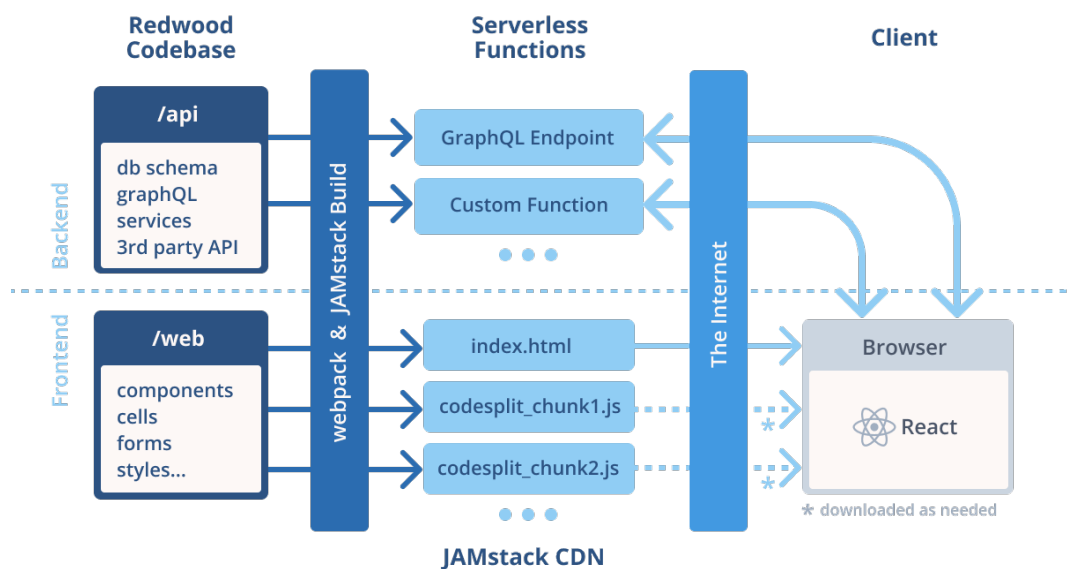
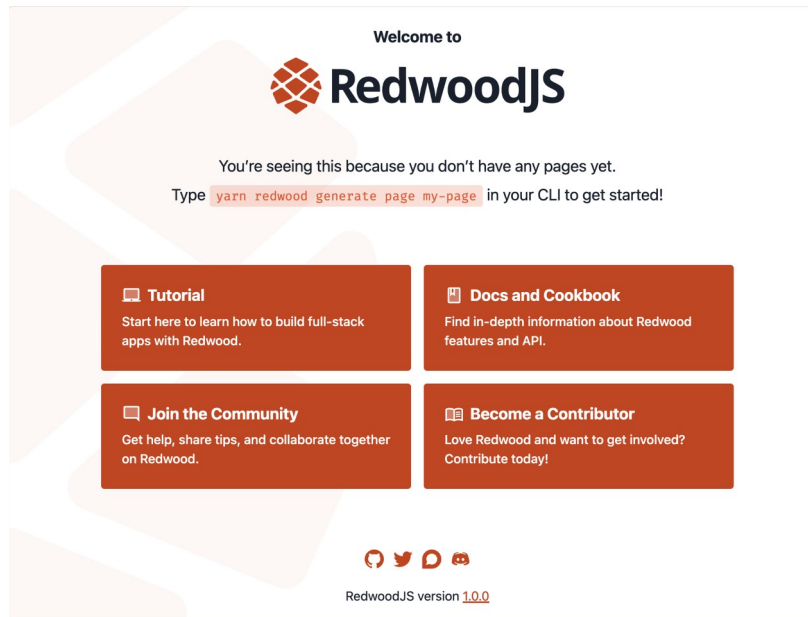
<input checked="" type="checkbox"/>	Name	ID d'instance	État de l'instance	Type d'insta...	Contrôle des statuts	Statut d'alarme	Zone de dispon...	DNS II
<input checked="" type="checkbox"/>	AL2023-server1	i-03289d9838566e5d3	En cours d'exécution	t2.micro	2/2 vérifications réussies	Afficher les alarm	eu-west-3a	ec2-5

Redwoodjs

Heaven Flow Group - Toulouse

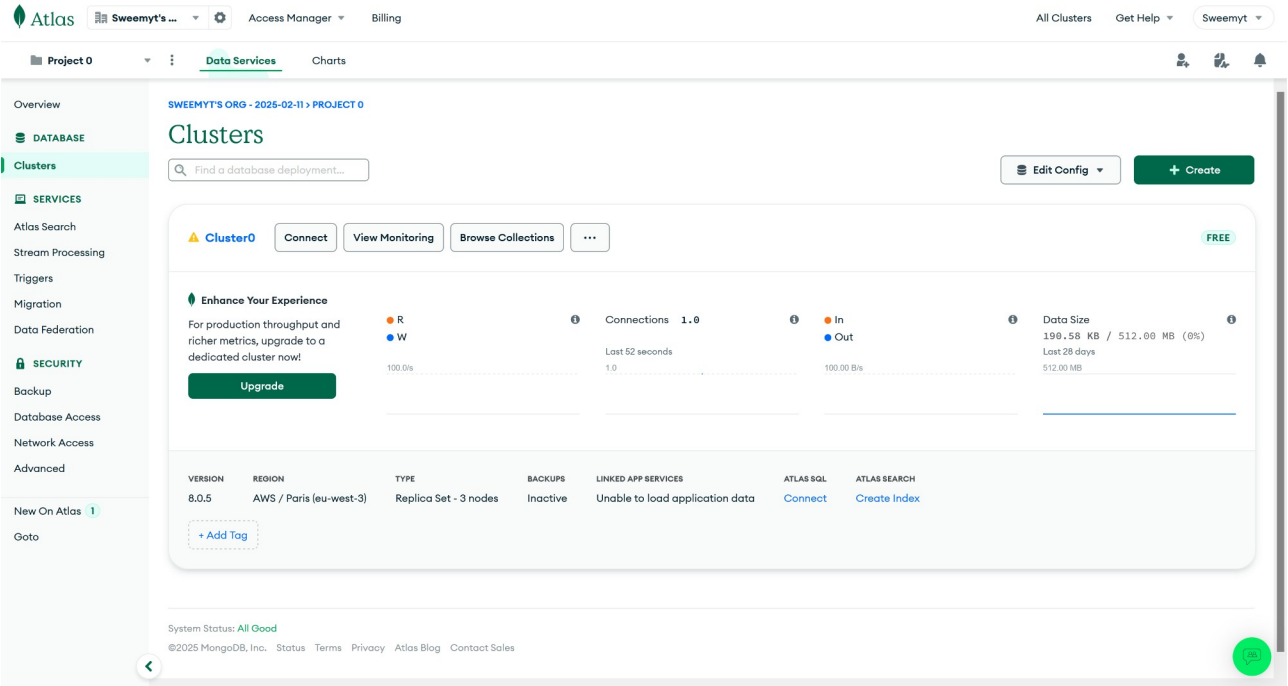
14

(comment ça fonctionne + page d'accueil)

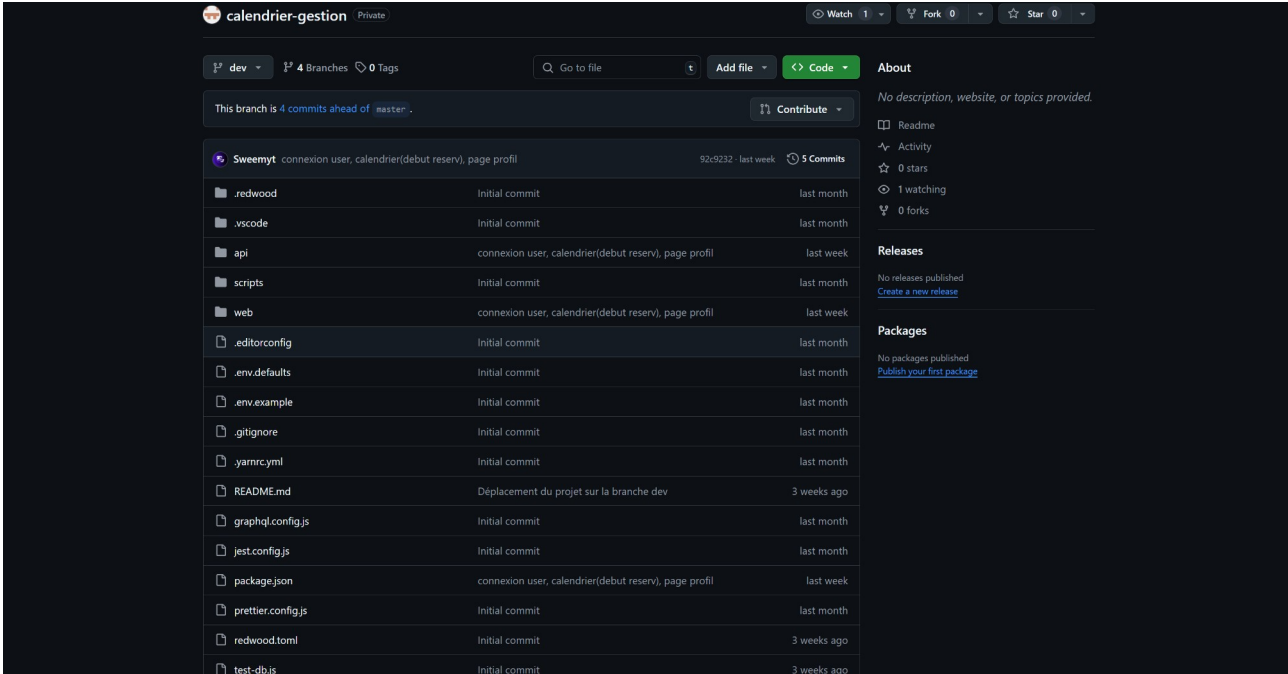




MongoDB atlas



Github



Exemple du code de l'application web

routes du site

```

1 // In this file, all page components from 'src/pages' are auto-imported. Nested
2 // directories are supported, and should be uppercase, each subdirectory will be
3 // prepended onto the component name.
4 // Examples:
5 // - 'src/pages/HomePage/homePage.js' -> HomePage
6 // - 'src/pages/Admin/booksPage/booksPage.js' -> AdminBookPage
7
8 import { PrivateSet, Router, Route, Set } from '@redwoodjs/router'
9
10 import HeaderLayout from 'src/layouts/HeaderLayout'
11 import ScaffoldLayout from 'src/layouts/ScaffoldLayout'
12
13 import { useAuth } from './auth'
14
15 const Routes = () => {
16   return (
17     <Router useAuth={useAuth}>
18       <Set wrap={HeaderLayout}>
19         <PrivateSet unauthenticated="home">
20           <Route path="/" page={NotFoundPage} name="notfound" />
21           <Route path="/admin/territoires/new" page={TerritoireNewTerritoirePage} name="newterritoire" />
22           <Route path="/admin/territoires/{id}/edit" page={TerritoireEditTerritoirePage} name="editterritoire" />
23           <Route path="/admin/territoires/{id}" page={TerritoireTerritoirePage} name="territoire" />
24           <Route path="/admin/territoires" page={TerritoireTerritoiresPage} name="territoires" />
25           <Set>
26             <Route path="/admin/salles/new" page={RoomNewRoomPage} name="newroom" />
27             <Route path="/admin/salles/{id}/edit" page={RoomEditRoomPage} name="editroom" />
28             <Route path="/admin/salles/{id}" page={RoomRoomPage} name="room" />
29             <Route path="/admin/salles" page={RoomRoomsPage} name="rooms" />
30           </Set>
31           <Route path="/profil" page={UserProfilePage} name="userprofil" />
32           <Route path="/admin-profil" page={AdminProfilePage} name="adminprofil" />
33         </PrivateSet>
34       </Router>
35     )
36   }
37 }
38
39 export default Routes

```

schémas prisma

```

1 // Keep more about both here:
2 // https://www.prisma.io/docs/orm/prisma-migrate
3
4 You, 3 weeks ago | 1 author (You)
5
6 datasource db {
7   provider = "mongodb"
8   url      = env("DATABASE_URL")
9 }
10
11 Generate
12 generator client {
13   provider = "prisma-client-js"
14   binaryTargets = "native"
15 }
16
17 // Define your own datamodels here and run 'yarn redwood prisma migrate dev' -> in our case
18 // to create migrations for them and apply to your dev DB.
19 You, 3 weeks ago | 1 author (You)
20
21 model Territoire {
22   id String @id @default(auto()) @map("_id") @db.ObjectId
23   name String
24   Room Room[]
25 }
26
27 You, 7 days ago | 1 author (You)
28
29 model Room {
30   id String @id @default(auto()) @map("_id") @db.ObjectId
31   name String
32   address String // Adresse de la salle
33   access String // Accès (métro, bus, etc.)
34   description String // Description (vocation, animations, etc.)
35   capacity String // Normes (ex: Type L - Catégorie 3)
36   equipments String // Liste des équipements mis à disposition
37   tarifs String // Informations sur les tarifs
38   //imageurls String[] // Tableaux d'images pour plusieurs images
39
40   territoireId String @db.ObjectId
41   territoire Territoire @relation(fields: [territoireId], references: [id])
42 }
43
44 Reservation Reservation[] You, 7 days ago + connexion user, calendrier(debut reserv
45
46 You, 7 days ago | 1 author (You)
47
48 model User {
49   id String @id @default(auto()) @map("_id") @db.ObjectId
50   email String? @unique
51   hashedPassword String
52   role String @default("user")
53   salt String
54   resetToken String?
55   resetTokenExpiresAt DateTime?
56 }
57
58 Reservation Reservation[]
59
60 You, 7 days ago | 1 author (You)
61
62 model Reservation {
63   id String @id @default(auto()) @map("_id") @db.ObjectId
64   roomId String @db.ObjectId
65   userId String @db.ObjectId
66   startDate DateTime
67   endDate DateTime
68   createdAt DateTime @default(now())
69   updatedAt DateTime @updatedAt
70
71   room Room @relation(fields: [roomId], references: [id])
72   user User @relation(fields: [userId], references: [id])
73
74   @@index([roomId])
75 }

```