

PROJET R SHINY  
SD2

SEPTEMBRE 2024



# À propos

GreenTech Solutions est une société de service fictive qui développe des applications.

Avec l'accélération du changement climatique et la hausse des prix de l'énergie, la sobriété énergétique est au cœur des préoccupations des Français.

C'est pourquoi Enedis nous sollicite en vue d'évaluer l'impact de la classe de Diagnostic de Performance Energétique (DPE) sur les consommations électriques de logements.



# Les livrables

Pour ce projet, le client attend plusieurs livrables. Il ne se contentera pas de l'application RShiny. Il est important pour lui d'être autonome sur la maintenance de l'application une fois celle-ci sera opérationnelle.



01

## Rapport d'étude RMarkdown

Le rapport est une étude statistique répondant à une ou deux problématiques sur les données à l'aide de visualisation et d'interprétation - 4 pages max

02

## L'application déployée sur shinyapp.io

L'application doit-être déployée sur <https://www.shinyapps.io/>. C'est un service gratuit permettant d'héberger jusqu'à 5 applications.

03

## Repository GitHub avec

- tous les scripts utilisés (avec commentaire)
- un README.md pour présenter les informations importantes
- documentation technique (orientée pour développeur) avec schéma de l'architecture dessiné avec [Draw.io](#) - 2 pages max
- documentation fonctionnelle (orientée utilisateur) - 2 pages max

04

## Captation vidéo de l'application

Réaliser une démo enregistrée de l'application RShiny pour présenter son fonctionnement et publier en privé sur YouTube.



## DEADLINE

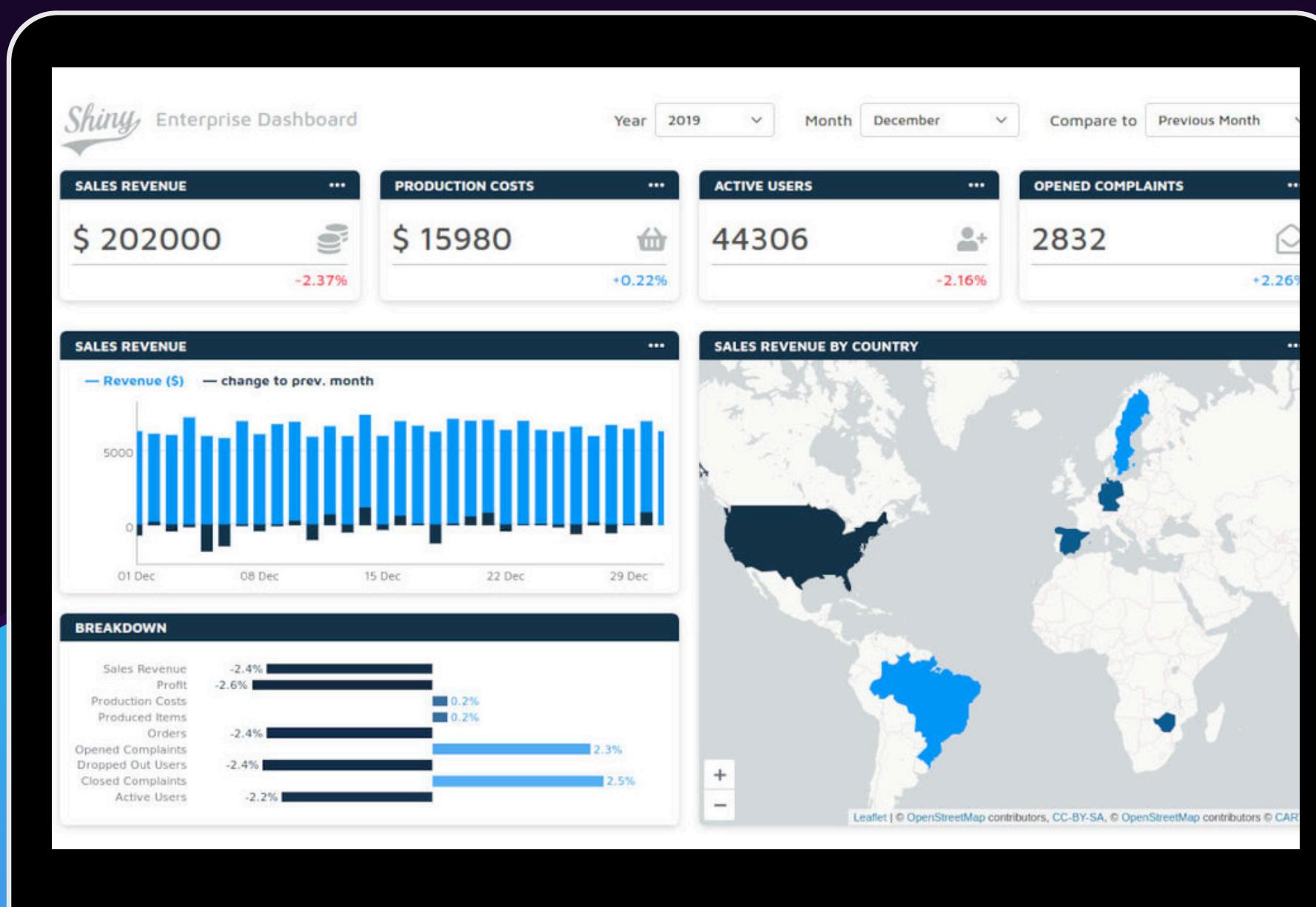
Pour réaliser ce projet, votre équipe a :

- 3 x 2H de TP
- 3 x 2H de TD

Le projet est à rendre avant le : 7/10/2024. Un jour de retard supplémentaire entraîne -1 pt de pénalité.



# Les technologies et outils à utiliser



- RShiny
- Draw.io
- RStudio
- CapCut
- GitHub Desktop
- ChatGPT

# Votre équipe

Pour ce projet, vous serez en équipe de 2. Vous aurez plusieurs rôles\* à jouer !



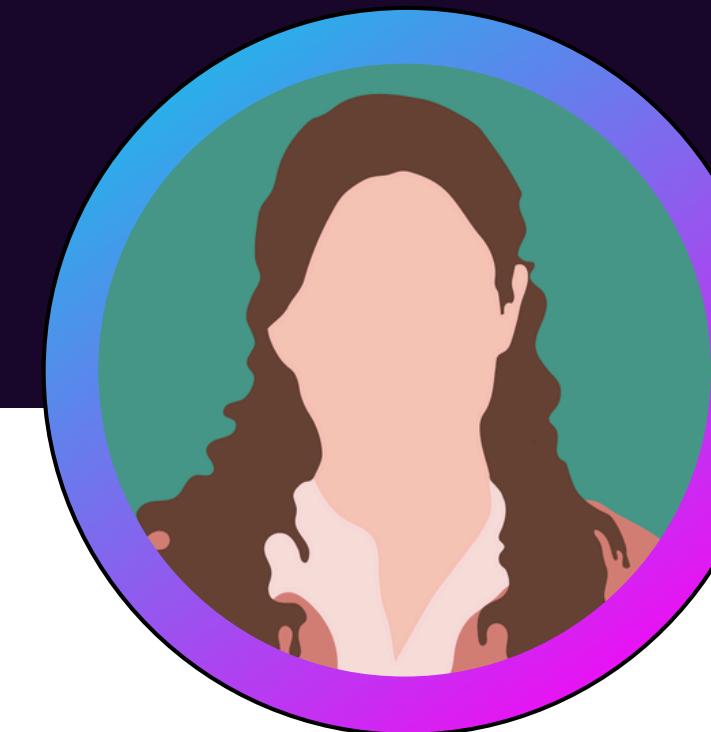
**Anthony**  
*Client*



**Rôle 1**  
*Chef de projet*



**Rôle 2**  
*Développeur*



**Rôle 3**  
*Data Scientist*



# Les rôles

## CHEF DE PROJET

est chargé de planifier, coordonner et gérer toutes les activités liées au projet afin de fournir une solution efficace, fonctionnelle et répondant aux besoins du client / des utilisateurs.

## DÉVELOPPEUR

joue un rôle essentiel dans le développement et le déploiement de la solution. Il veille à effectuer des tests pour éviter les bugs. Versionner et documenter son code. Il garantit le bon fonctionnement de l'application (back-end / front-end).

## DATA SCIENTIST

joue un rôle essentiel dans l'analyse des données, la préparation des données et la création de visualisations pertinentes afin de fournir des statistiques exploitables aux utilisateurs finaux. Il est également chargé de construire et d'interpréter des modèles prédictifs.

# Les fonctionnalités de l'application



## Standard

- KPI, graphiques et tableaux sur les logements neufs et anciens
- Filtres sur les données pour mettre à jour les statistiques et visualisations
- Plusieurs onglets
- Des images et icônes dans l'application
- Cartographie
- Déploiement sur shinyapps.io



## Intermédiaire

- Fonctionnalités Standard
- Bouton pour exporter les graphiques en .png
- Bouton pour exporter les données sélectionnées en .csv
- L'utilisateur peut sélectionner un thème de son choix
- Corrélation entre les variables quantitatives sélectionnées
- L'utilisateur peut modéliser une variable X selon une variable Y avec la méthode de régression linéaire simple



## Expert

- Fonctionnalités Intermédiaire
- Charte visuelles avec un script CSS
- L'utilisateur peut rafraîchir avec les dernières données ajoutées
- Connexion utilisateur avec mot de passe pour accéder à l'application



# Les étapes



Rappels du langage R  
Extraction des données du projet



Initiation au RMarkdown et GitHub



Initiation au RShiny et développement du front-end



Développement du back-end



Deploiement de l'application non terminée sur shinyapps.io



Finalisation des fonctionnalités "standard"



Finalisation des fonctionnalités "intérmédiaire"



Finalisation des fonctionnalités "expert"



Finalisation des livrables



# LIENS UTILES

- Contexte du Challenge ENEDIS : <https://defis.data.gouv.fr/defis/65b76f15d7874915c8e41298>
- Initiation au RMarkdown : <https://rmarkdown.rstudio.com/lesson-1.html>
- Initiation à GitHub Desktop : <https://docs.github.com/fr/desktop/overview/>
- Cours de R de SD1 : [https://github.com/asardell/IUT\\_SD1](https://github.com/asardell/IUT_SD1)
- API avec R : <https://statisticsglobe.com/api-in-r#what-is-an-api>
- API des logements neufs : <https://data.ademe.fr/datasets/dpe-v2-logements-neufs/api-doc>
- API des logements existants : <https://data.ademe.fr/datasets/dpe-v2-logements-existants/api-doc>
- API - GET query with range parameters from ElasticSearch :  
[https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html#\\_ranges](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html#_ranges)
- Base Adresse Nationale au format .csv : <https://adresse.data.gouv.fr/donnees-nationales>
- Carte interactive avec leaflet : <https://rstudio.github.io/leaflet/articles/markers.html>
- Déployer l'application : <https://www.shinyapps.io/>
- Charte graphique et palette : <https://colorhunt.co/>
- Dessiner des schémas : <https://www.drawio.com/>
- Apprendre RShiny et gallery : <https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html>
- Exemple de graphique en R : <https://r-graph-gallery.com/>
- Gestion des users dans RShiny : <https://datastorm-open.github.io/shinymanager/>

# Cahier des charges 1/2

- Non réalisé
- : Inachevé
- ✓ : Réalisé



## Rapport d'étude RMarkdown (4 pages max)


Mise en forme, rédaction et export au format HTML

KPI pertinents, Statistique bivariés et différents type de graphique

Automatisation avec choix du code postal et type de logement (ancien, neuf ou les deux).


## Repository GitHub


Création d'un repository public appelé "iut\_sd2\_rshiny\_enedis"

Tous les scripts (avec commentaires) utilisés dans le projet

Un fichier README.md pour présenter les informations importantes du repository

## Une documentation technique de l'application (2 pages max) :


- avec un schéma de l'architecture
- qui explique comment installer l'application sur son poste
- qui présente les packages nécessaires
- mise en forme, rédaction

## Une documentation fonctionnelle de l'application (2 pages max) :


- qui présente l'intérêt de chaque page de l'application
- qui présente les fonctionnalités majeures de l'application
- mise en forme, rédaction

## Captation vidéo de l'application (5 min max)


La vidéo explique comment installer l'application en local et présente les fonctionnalités majeures de l'application

# Cahier des charges 2/2

- : Non réalisé
- : Inachevé
- ✓ : Fonctionnel



## Pack "Standard"


L'application est déployée sur shinyapps.io l'application disponible via son url

L'application dispose d'au moins 3 onglets/pages différentes

Il y a des images et des icônes dans l'application

L'application dispose d'au moins une cartographie interactive avec des markers

Il y a une page "Contexte" qui permet de présenter et visualiser les données disponibles

L'utilisateur peut interagir avec l'application via les widgets : select, checkbox, sliders et radio buttons

L'utilisateur peut filtrer dynamiquement les données pour actualiser des tableaux et visualisations

L'application propose plusieurs KPI

L'application propose au moins 4 types de graphiques différents (histogramme, boîte à moustache, diagramme, nuage de point)

## Pack "Intermédiaire"


L'utilisateur peut sélectionner un thème de son choix dans l'application

Il y a des boutons pour exporter les graphiques en .png et exporter les données sélectionnées en .csv

L'utilisateur peut sélectionner 2 variables (X et Y) pour calculer le coefficient de corrélation et une régression linéaire simple dans un nuage de point avec la droite de régression

## Pack "Expert"


L'application dispose d'une charte visuelle avec un script CSS

L'utilisateur peut rafraîchir les données actuelles avec les nouvelles via l'API et la date de réception du DPE

L'application demande de se connecter avec un user / password pour accéder à l'application

# THANK YOU