

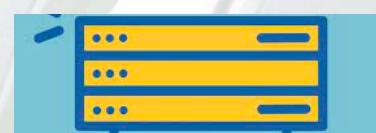
27 AVRIL 2021

KAMARA ABDULRAHMAN

RAPPORT DE PROJETS

Concepteur Développeur d'applications

Septembre 05/2021 à Août 31/2022



SOMMAIRE

CONCEVOIR ET DÉVELOPPER DES COMPOSANTS D'INTERFACE UTILISATEUR EN INTÉGRANT LES SÉCURITÉ.....PAGE.....04-18

- 1)- Maquetter une application
- 2)- Développer une interface utilisateur de type desktop
- 3)- Développer des composants d'accès aux données
- 4)- Développer la partie front-end d'une interface utilisateur web

CONCEVOIR ET DÉVELOPPER LA PERSISTANCE DES DONNÉES EN INTÉGRANT LES RECOMMANDATIONS DE SÉCURITÉ.....PAGE.....28-39

- 1)- Concevoir une base de données
- 2)- Mettre en place une base de données
- 3)- Développer des composants dans le langage d'une base de données
- 4)- Développer la partie back-end d'une interface utilisateur web

CONCEVOIR ET DÉVELOPPER UNE APPLICATION MULTICOUCHE RÉPARTIE EN INTÉGRANT LES RECOMMANDATIONS DE SÉCURITÉ.....PAGE.....34-50

- 1)- Collaborer à la gestion d'un projet informatique et à l'organisation de ???
- 2)- L'environnement de développement
- 3)- Concevoir une application
- 4)- Développer des composants métier
- 5)- Construire une application organisée en couches
- 6)- Développer une application mobile
- 7)- Préparer et exécuter les plans de tests d'une application
- 8)- Préparer et exécuter le déploiement d'une application

RÉALISATION.....	PAGE
MVC.....	50-53
Security (react-native).....	54
Security (api-platefom).....	55-56
RGPD.....	57

SOMMAIRE DU PROJET

Compétences du référentiel couvertes par le

Projets

Pour l'activité type 1 « **Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurités** » :

- Réalisation d'un logiciel en python
- Application de la Méthode Agile
- Maquetter une application
- Développer des composants d'accès aux données
- Réaliser une interface utilisateur web statique et adaptable
- Développer une interface utilisateur web dynamique

Exemple n° 1

Cadre du projet

LE PROJET

L'objectif est de créer une interface utilisateur de type desktop avec python.

Le projet exposé dans ce dossier a été développé

du 19/04/2022 au 20/04/2022

Planification

Le but est de décider des éléments du produit Backlog à développer lors de l'itération et de la façon de s'organiser pour parvenir à l'objectif du sprint.

OUTILS

VsCode



Langage python



Pygame



Py-to-exe



Outils utilisés

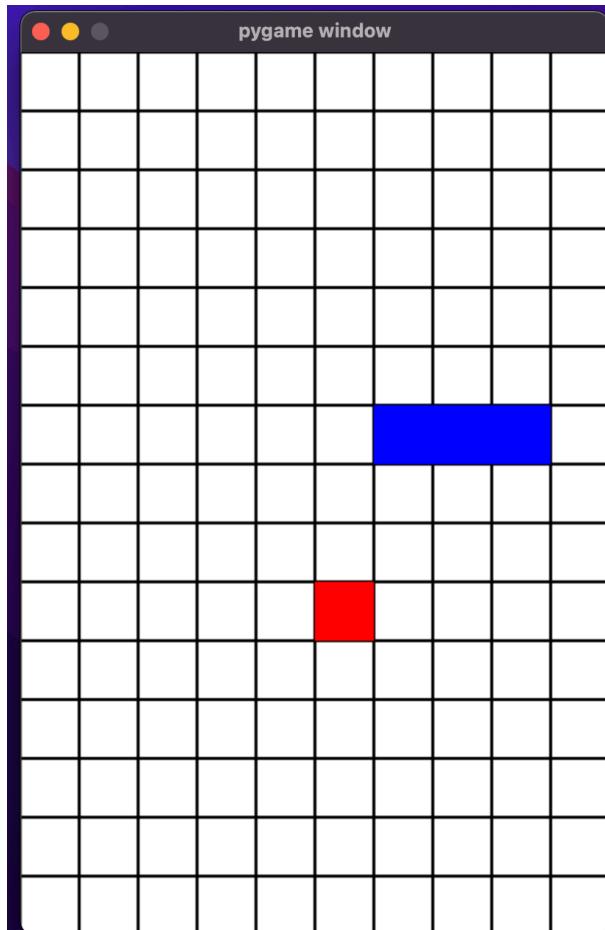
VsCode: J'ai utilisé vscode comme un éditeur de code. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré

Module Pygame: est une bibliothèque multiplateforme gratuite et open source pour le développement d'applications multimédias telles que des jeux vidéo utilisant Python. Il utilise la bibliothèque Simple DirectMedia Layer et plusieurs autres bibliothèques populaires pour résumer les fonctions les plus courantes, ce qui rend l'écriture de ces programmes plus intuitive.

Module random : J'ai utilisé ce module pour mettre en oeuvre des générateurs de nombres pseudo-aléatoires pour différentes distributions.

Python : Langage utilisé pour développer le logiciel.

Démonstration



Démonstration

```
#Boucle de jeux
while game_on:

    #Fermeture de la fenêtre
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit

    #Maj du snake à la vitesse de 200milisec au lieu de 60 FPS
    if event.type == SCREEN_UPDATE:
        game.update()

    #Capte les touches pressés
    if event.type == pygame.KEYDOWN:

        if event.key == pygame.K_UP:
            if game.snake.direction != 'DOWN':
                game.snake.direction = 'TOP'

        if event.key == pygame.K_DOWN:
            if game.snake.direction != 'TOP':
                game.snake.direction = 'DOWN'

        if event.key == pygame.K_LEFT:
            if game.snake.direction != 'RIGHT':
                game.snake.direction = 'LEFT'

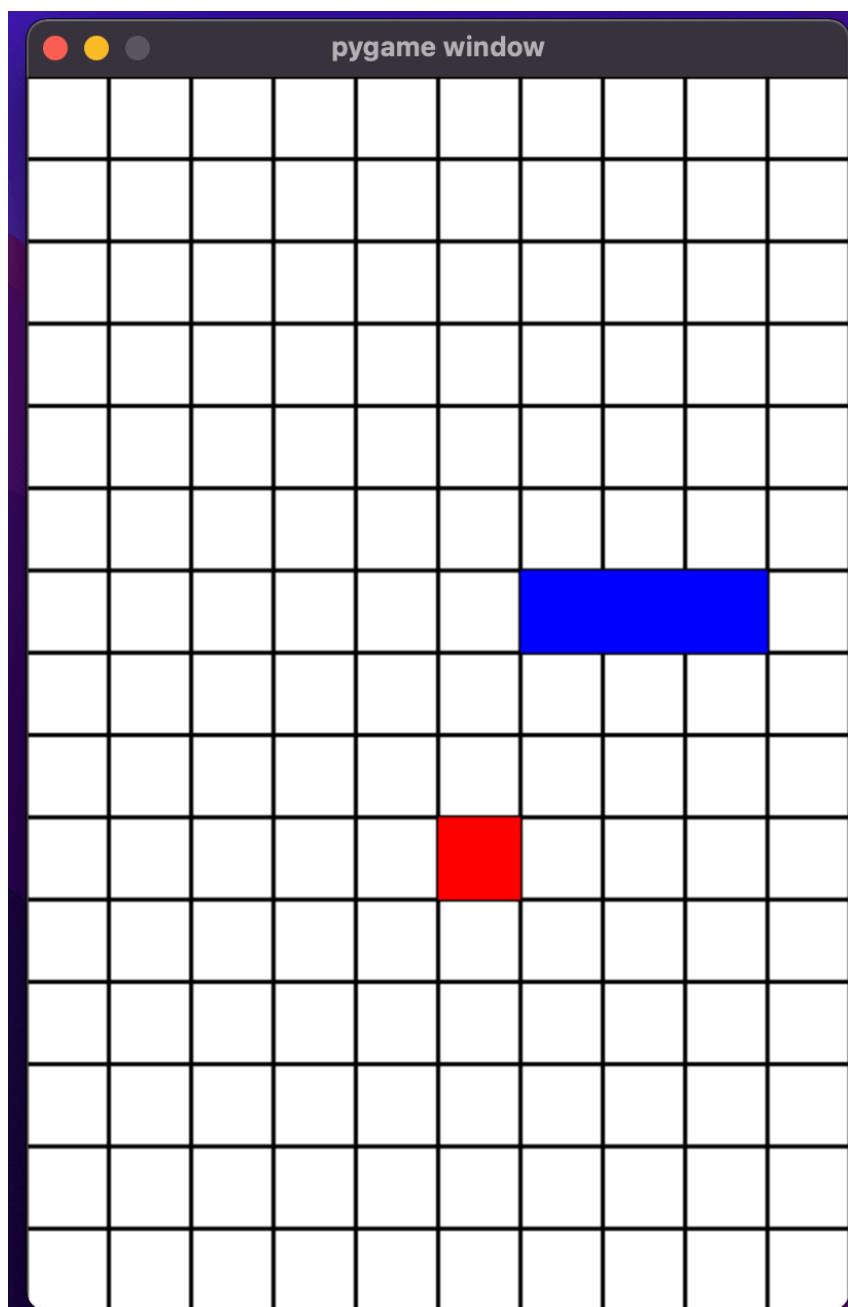
        if event.key == pygame.K_RIGHT:
            if game.snake.direction != 'LEFT':
                game.snake.direction = 'RIGHT'
```

Review

DEMO : Présentation du projet et test de la fonctionnalité.

FEED-BACK: Marquer le bug et les problèmes que je rencontre et retravailler dessus pour réaliser les buts du projet.

Livrable



Exemple n° 2

Cadre du projet

LE PROJET

La CRESS PACA souhaite faire évoluer son modèle économique. En effet, l'institution souhaite passer d'un modèle de subvention publique à un modèle où elle serait plus autonome : le modèle des trois-tiers. Ainsi, elle serait financée de la manière suivante :

- 1/3 de subventions,
- 1/3 cotisations des membres de la CRESS PACA,
- 1/3 ventes de produits et services.

L'outil ici développé, serait donc un service que proposerait la CRESS sous forme de plateforme numérique à toutes les structures ESS et non uniquement ses membres contre participation.

De même, la CRESS, avec cet outil, souhaite rendre le secteur de l'ESS plus compétitif sur le marché. Pour cela, une acculturation doit être menée. Cet outil serait une première étape pour atteindre cet objectif.

Aussi, il y a une forte volonté de digitaliser le réseau d'autant plus en cette période. Digitaliser le réseau pour gagner en visibilité mais surtout accélérer leur développement économique.

Et bien évidemment, cet outil digital participerait au rayonnement de l'ESS dans la région. Il permet à la CRESS de mener à bien sa mission et atteindre ses 3 objectifs.

Le projet exposé dans ce dossier a été développé pendant la période d'alternance du 01 janvier au 31 août 2022.

Planification

Product Backlog

CAHIER DES CHARGES

Plateforme Numérique CRESS PACA

CAHIER DES CHARGES

Plateforme Numérique CRESS PACA



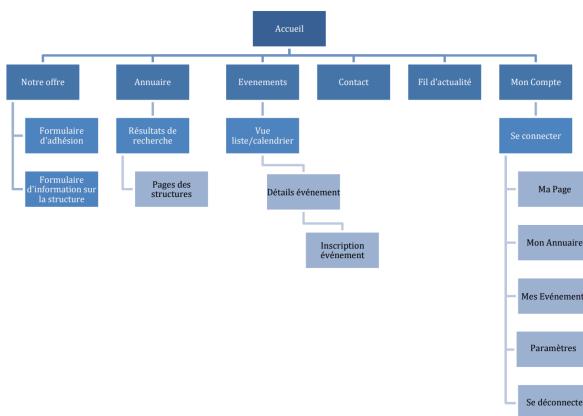
Table des matières

1. Définition, scope et objectifs du projet	3
1.1 L'entreprise : la CRESS PACA	3
1.2 Contexte de la demande	3
1.3 Nos objectifs	4
1.4 Cibles	4
2. Rôles et permissions utilisateur de la plateforme	5
2.1 Visiteur	5
2.2 Membre	5
2.2.1 Membre standard	6
2.2.2 Membre premium	6
2.3 Administrateur	7
3. Spécifications et user stories	8
3.1 Module 1 : Page accueil	8
3.2 Module 2 : Notre offre (inclus formulaire d'inscription et formulaire de renseignement)	8
3.3 Module 3 : Annuaire	9
3.4 Module 4 : Événements (inclus zoom événements et formulaire d'inscription à l'événement)	11
3.5 Module 5 : Fil d'actualité	12
3.6 Module 6 : Contact	13
3.7 Module 7 : Mon Compte	13
3.8 Module 8 : Fonctionnalités envisagées dans un second temps	15
3.8.1 Le chat	15
3.8.2 La page Groupes et clubs	15
3.8.3 La Page Tutorials et F.A.Q	15
4. Expérience utilisateur	16
...	..

sur le "Fil d'actualité" dès lors que l'administrateur aura validé le contenu.

2.2.2 Membre premium

Les membres du niveau premium auront exactement les mêmes droits que les membres standard. Toutefois, leur niveau leur donne la possibilité d'ajouter des événements - organisés par eux ou leurs partenaires - ce qui est impossible pour les membres standards.

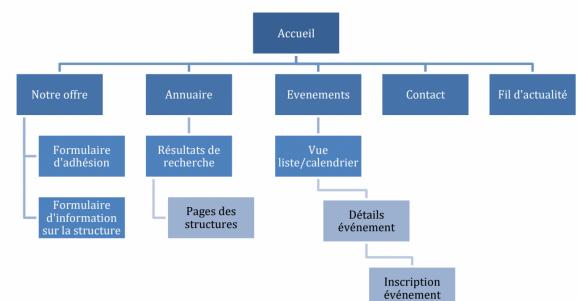


2. Rôles et permissions utilisateur de la plateforme

Trois rôles seront disponibles sur la plateforme. Chacun aura des permissions et des responsabilités différentes décrites ci-dessous :

2.1 Visiteur

Un visiteur est un utilisateur de la plateforme qui n'est pas connecté. Il a accès à toutes les pages du menu principal mais ne peut avoir accès au menu "Mon Compte" car il n'en possède pas. Il ne peut rien publier, simplement consulter les pages ou s'inscrire à la plateforme et aux événements.



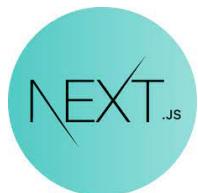
2.2 Membre

Outils :

VsCode



Next.js



Material UI



strapi.io



Git



Docker



Trello



Google Meet



Google Chart



Outils utilisés :

VsCode

Next.js : Exécuter des templates avec Nextjs(Framwork javascript).

Figma : Création de notre maquette.

Material UI : Pour structurer le site.

Strapi.io : Exécuter la partie back-end.

Git : Utilisé pour le travail collaboratif.

Docker: Déployer les applications en utilisant plusieurs container avec une base de données.

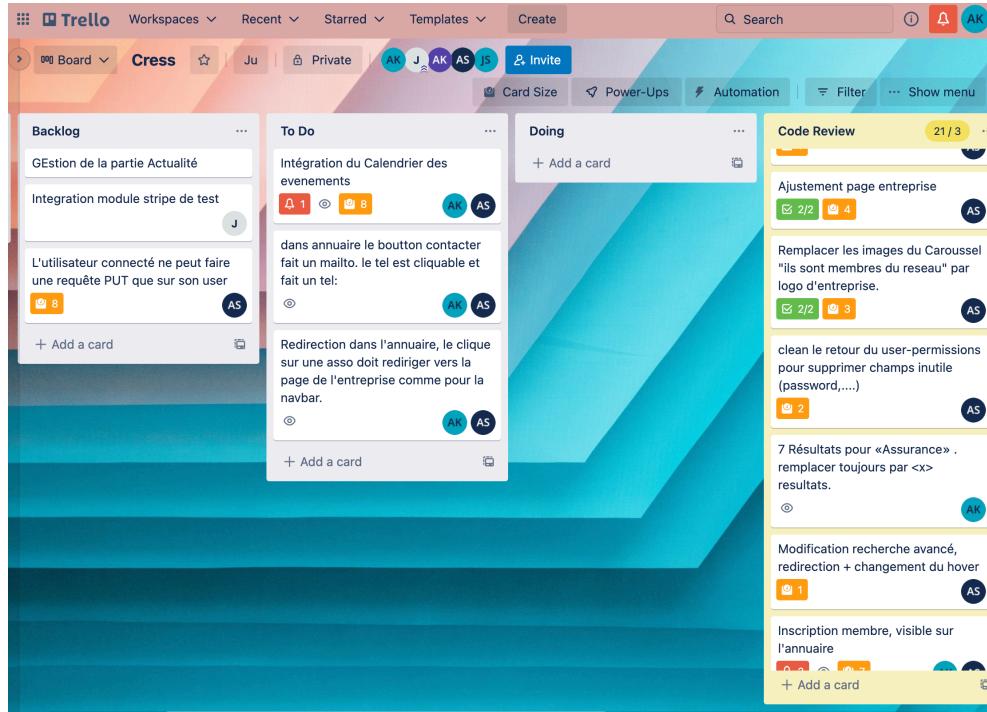
Trello : Créer une interface de travail en collaboration.

Google Meet : Pour organiser notre réunion chaque matin.

Google Chart: Utilisé comme un espace de discussion avec des suggestions.

Work Scheme

Ci-dessous la capture d'écran de mon trello qui me permet de découper mon travail par rapport aux user stories.



Next.js: Initialiser le projet et exécuter des templates avec Nextjs

```
Next.js/cress-paca
├── node_modules
├── pages
├── public
└── styles
    └── .eslintrc.json
    ├── .gitignore
    └── package.json
    └── README.md
    └── yarn.lock
    └── Read.me

string.prototype.trimstart@1.0.5
strip-ansi@6.0.1
strip-bom@3.0.0
strip-json-comments@3.1.1
supports-color@7.2.0
text-table@0.2.0
to-regex-range@5.0.1
tslib@1.14.1
tsutils@3.21.0
type-check@0.4.0
type-fest@0.20.2
unbox-primitive@1.0.2
uri-js@4.4.1
v8-compile-cache@2.3.0
which-boxed-primitive@1.0.2
which@2.0.2
word-wrap@1.2.3
yallist@4.0.0
: Done in 27.49s.

Initialized a git repository.

Success! Created cress-paca at /Applications/MAMP/htdocs/juniorApp/React_native/Juni
or_v1/Next.js/cress-paca
Inside that directory, you can run several commands:

yarn dev
  Starts the development server.

yarn build
  Builds the app for production.

yarn start
  Runs the built app in production mode.

We suggest that you begin by typing:
cd cress-paca
yarn dev
→ Next.js
```

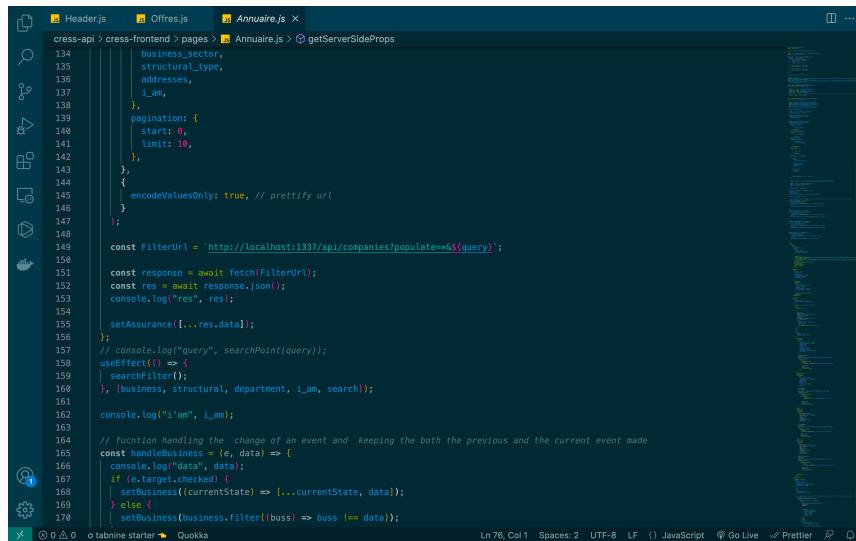
Figma : Ci-dessous la capture d'écran de maquette de cress

The screenshot shows the CRESS website's search interface. At the top, there is a navigation bar with links to Accueil, Notre offre, Annuaire, Evénements, Actualités, and Contact. A "Se connecter" button is also present. Below the navigation, there is a banner featuring several small images related to business and social issues. A search bar with placeholder text "Votre recherche..." and a "Recherche avancée" dropdown is visible. The main content area is titled "Annuaire" and contains a search bar with the placeholder "Recherche avancée". Below this, there is a list of three companies: MACIF, SOLIMUT MUTUELLE DE FRANCE, and MUTUALITÉ FRANÇAISE, each with its logo and a brief description of their services.

The screenshot shows the CRESS website's search results for the term "Assurance". The search bar at the top has "Assurance" typed into it. The main content area is titled "Annuaire" and displays a search result for "Assurance". The results list includes two entries for MACIF, both with their logos and descriptions of their online insurance services. There is also a "Contactez-nous" button with the phone number 03 20 54 21 85. At the bottom of the page, there is a dark banner with a PDF file named "webdesign-cress-paca.pdf" and its details: "Added 14 Mar at 10:46 - 2.73 MB", "Open in new tab", and "Delete".

Itération

Ci-dessous la capture d'écran d'une page annuaire de cress

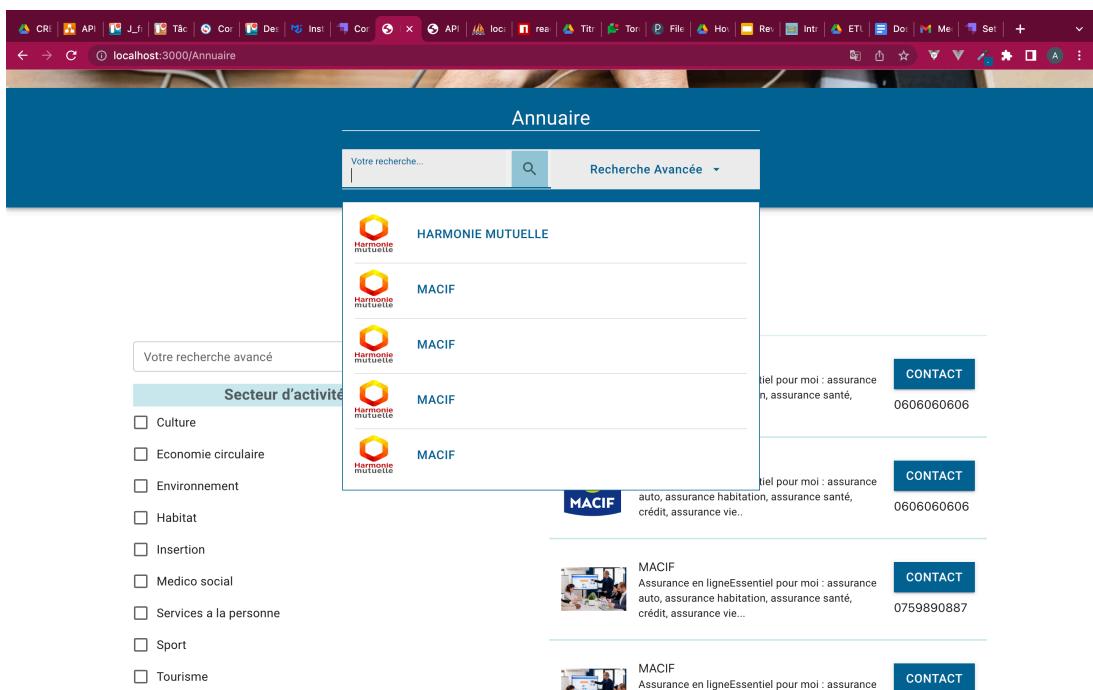


```
134     business_sector,
135     structural_type,
136     addresses,
137     i_am,
138   },
139   pagination: {
140     start: 0,
141     limit: 10,
142   },
143   {
144     encodeValuesOnly: true, // prettify url
145   }
146 );
147 }

148 const FilterUrl = "http://localhost:1337/api/companies?populate=*_query";
149
150 const response = await fetch(FilterUrl);
151 const res = await response.json();
152 console.log("res", res);
153
154 setAssurance([...res.data]);
155
156 // console.log("query", searchPoint(query));
157 useEffect(() => {
158   searchFilter();
159 }, [business, structural, department, i_am, search]);
160
161 console.log("i_am", i_am);
162
163 // function handling the change of an event and keeping the both the previous and the current event made
164 const handleBusiness = (e, data) => {
165   console.log("data", data);
166   if (e.target.checked) {
167     setCurrentState({currentState: data});
168   } else {
169     setBusiness(business.filter((buss) => buss !== data));
170   }
171 };
172
173
174
175
176
177
178
```

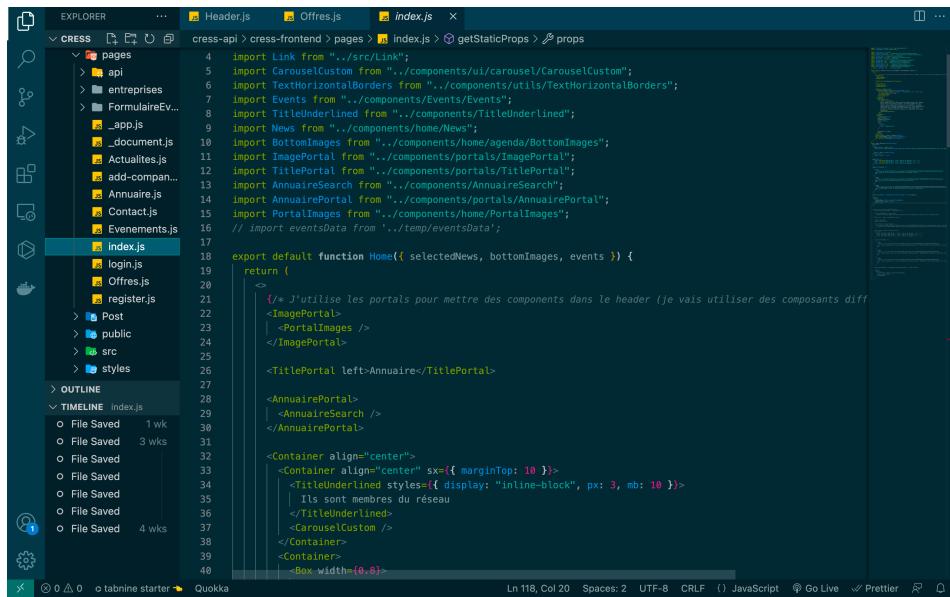
Code qui fait un appel de l'api avec une méthode query pour filtrer mes réponses.

Ci-dessous la capture d'écran de la plage annuaire en UX



Itération

Ci-dessous la capture d'écran des affichages de la page Accueil



```
EXPLORER Header.js Offres.js index.js ...
cress-api > cress-frontend > pages > index.js > getStaticProps > props
pages
> api
> entreprises
> FormulaireEv...
  _app.js
  _document.js
  Actualites.js
  add-compan...
  Annuaire.js
  Contact.js
  Evenements.js
  index.js
  login.js
  Offres.js
register.js
> Post
> public
> src
> styles
index.js
login.js
Offres.js
register.js
> OUTLINE
> TIMELINE.js
o File Saved 1 wk
o File Saved 3 wks
o File Saved 32
o File Saved 33
o File Saved 34
o File Saved 35
o File Saved 36
o File Saved 37
o File Saved 38
o File Saved 39
o File Saved 40
File Saved 1 wk
File Saved 3 wks
File Saved 32
File Saved 33
File Saved 34
File Saved 35
File Saved 36
File Saved 37
File Saved 38
File Saved 39
File Saved 40
Ln 118, Col 20 Spaces: 2 UTF-8 CRLF ⓘ JavaScript ⓘ Go Live ⓘ Prettier ⓘ
```



Ils sont membres du réseau

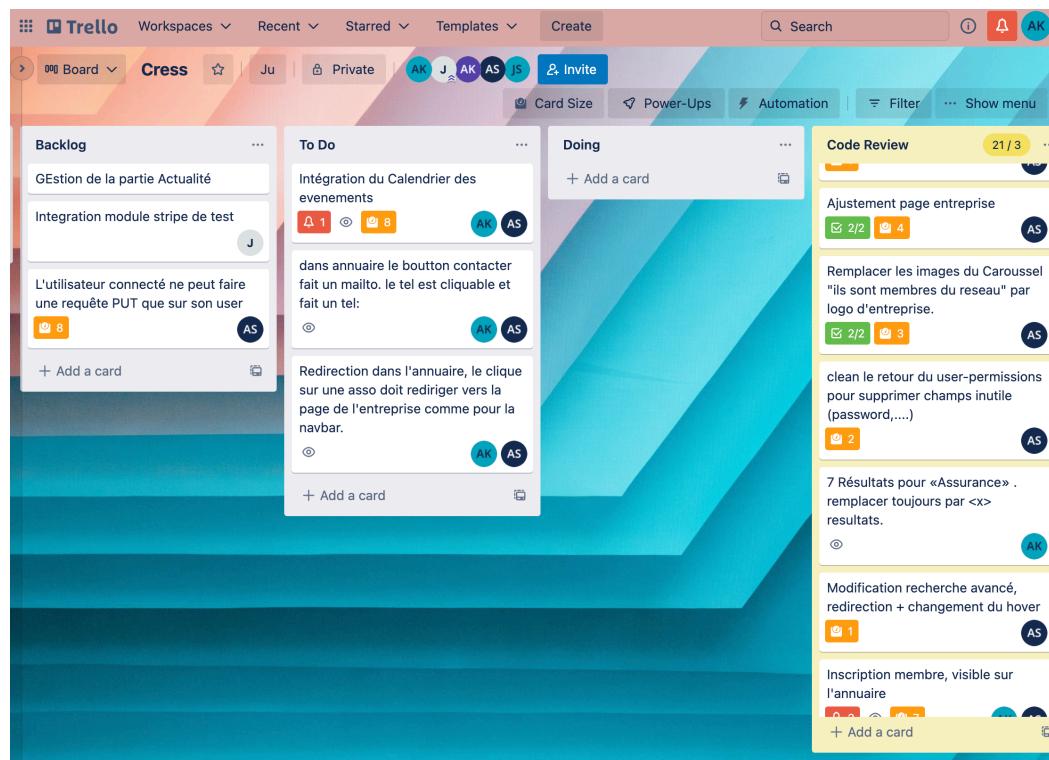


Review

DEMO : Présentation du projet et test de la fonctionnalité.

FEED-BACK: Marker le bug et les problèmes que je rencontre et retravailler dessus pour réaliser les buts du projet.

Ci-dessous la capture d'écran de mon trello à fin de mon travail



Compétences du référentiel couvertes par le projet

Pour l'activité type 2 « **Concevoir et développer la persistance des données en intégrant les recommandations de sécurité : »** :

Exemple n° 1

LE PROJET

Cress-Paca

La CRESS PACA souhaite faire évoluer son modèle économique. En effet, l'institution souhaite passer d'un modèle de subvention publique à un modèle où elle serait plus autonome : le modèle des trois-tiers. Ainsi, elle serait financée de la manière suivante :

- 1/3 de subventions,
- 1/3 cotisations des membres de la CRESS PACA,
- 1/3 ventes de produits et services.

L'outil ici développé, serait donc un service que proposerait la CRESS sous forme de plateforme numérique à toutes les structures ESS et non uniquement ses membres contre participation.

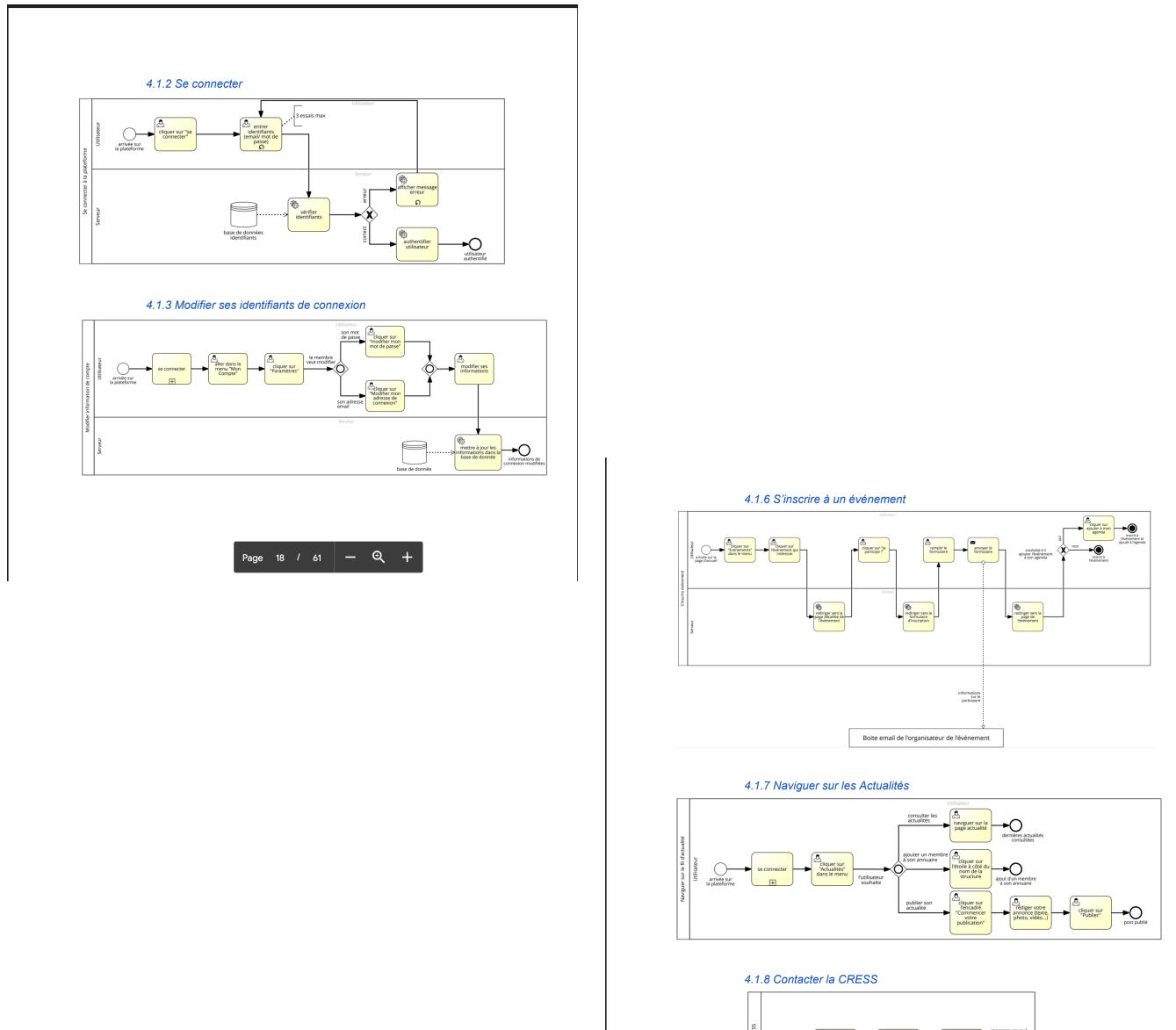
De même, la CRESS, avec cet outil, souhaite rendre le secteur de l'ESS plus compétitif sur le marché. Pour cela, une acculturation doit être menée. Cet outil serait une première étape pour atteindre cet objectif.

Aussi, il y a une forte volonté de digitaliser le réseau d'autant plus en cette période. Digitaliser le réseau pour gagner en visibilité mais surtout accélérer leur développement économique. Et bien évidemment, cet outil digital participerait au rayonnement de l'ESS dans la région. Il permet à la CRESS de mener à bien sa mission et atteindre ses 3 objectifs

Le projet exposé dans ce dossier a été développé pendant la période d'alternance du 01 janvier au 31 août 2022

Planification

Product Backlog

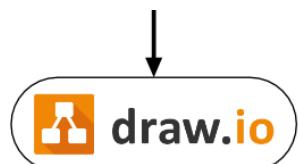


Outils :

VsCode



Draio



Api-plateforme



Git



Trello



Google Chart.



Google Meet



Swagger



Postman



Plesk



Outils utilisés

Satrapi : Exécuter la partie back-end.

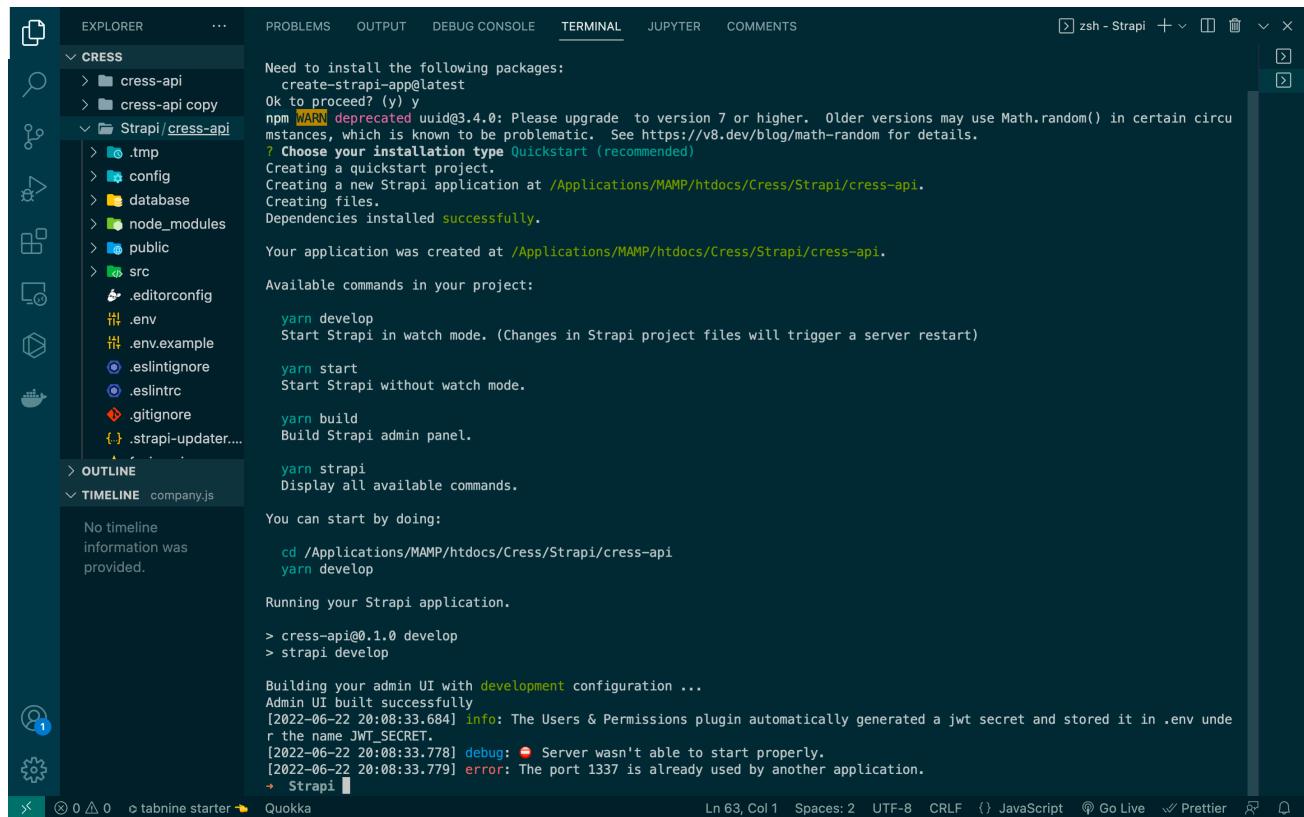
Figma : Création des maquettes.

Docker : Déployer les applications en utilisant plusieurs container avec une base de données.

Work Scheme

Ci-dessous la capture d'écran de l'initialisation du projet

Command : npx create-strapi-app@latest my-project



```
zsh - Strapi
Need to install the following packages:
  create-strapi-app@latest
Ok to proceed? (y) y
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
? Choose your installation type Quickstart (recommended)
Creating a quickstart project.
Creating a new Strapi application at /Applications/MAMP/htdocs/Cress/Strapi/cress-api.
Creating files.
Dependencies installed successfully.

Your application was created at /Applications/MAMP/htdocs/Cress/Strapi/cress-api.

Available commands in your project:

  yarn develop
    Start Strapi in watch mode. (Changes in Strapi project files will trigger a server restart)

  yarn start
    Start Strapi without watch mode.

  yarn build
    Build Strapi admin panel.

  yarn strapi
    Display all available commands.

You can start by doing:

  cd /Applications/MAMP/htdocs/Cress/Strapi/cress-api
  yarn develop

Running your Strapi application.

> cress-api@0.1.0 develop
> strapi develop

Building your admin UI with development configuration ...
Admin UI built successfully
[2022-06-22 20:08:33.684] info: The Users & Permissions plugin automatically generated a jwt secret and stored it in .env under the name JWT_SECRET.
[2022-06-22 20:08:33.778] debug: ✨ Server wasn't able to start properly.
[2022-06-22 20:08:33.779] error: The port 1337 is already used by another application.
→ Strapi
```

Itération

Ci-dessous la capture d'écran de la création des entity de cress-api qui vont automatiquement se mettre à jour dans les directory et créer le controller et le router.

The screenshot shows the Strapi Content-Type Builder interface at localhost:1337/admin/plugins/content-type-builder/content-types/api:address.address. The left sidebar shows the navigation menu with 'Content-Type Builder' selected. The main area displays the 'Address' collection type configuration. The table lists fields: 'number' (Number), 'street' (Text), 'complement' (Text), 'city' (Text), 'department' (Enumeration), and 'postal_code' (Text). A button at the bottom right allows adding another field to the collection type.

NAME	TYPE	Actions
123 number	Number	edit trash
Ab street	Text	edit trash
Ab complement	Text	edit trash
Ab city	Text	edit trash
Ab department	Enumeration	edit trash
Ab postal_code	Text	edit trash

Ci-dessous la capture d'écran de la création des relations entre les entity de cress-api qui vont automatiquement s'exécuter dans les directory et créer le controller, le content-type et le router.

The screenshot shows the Strapi Content-Type Builder interface at localhost:1337/admin/plugins/content-type-builder/content-types/api:event.event. A modal window titled "Edit User" is open, showing the relationship configuration between the "Event" and "User" entities. The "Event" entity has a field named "user" and the "User" entity has a field named "events". A blue line with arrows indicates a many-to-many relationship labeled "User has many Events". Other fields visible include "event_url" and "event.name". The "Edit" button is at the top right of the main content area.

The screenshot shows a code editor displaying the generated `company.js` controller file for the `cress-api` project. The code is written in JavaScript and uses Promises and async/await syntax. It includes logic for finding a company by ID, populating related entities like addresses, social network links, keywords, users, events, and roles, and sanitizing the output. The code editor interface includes a sidebar with file navigation, a left panel with "OUTLINE" and "TIMELINE" sections, and a bottom status bar showing file statistics and tool icons.

```

// Generated code from the Strapi Content-Type Builder
module.exports = createCoreController("api::company.company", ({ strapi }) => ({
  async findWithEvents(ctx) {
    const { id } = ctx.params;

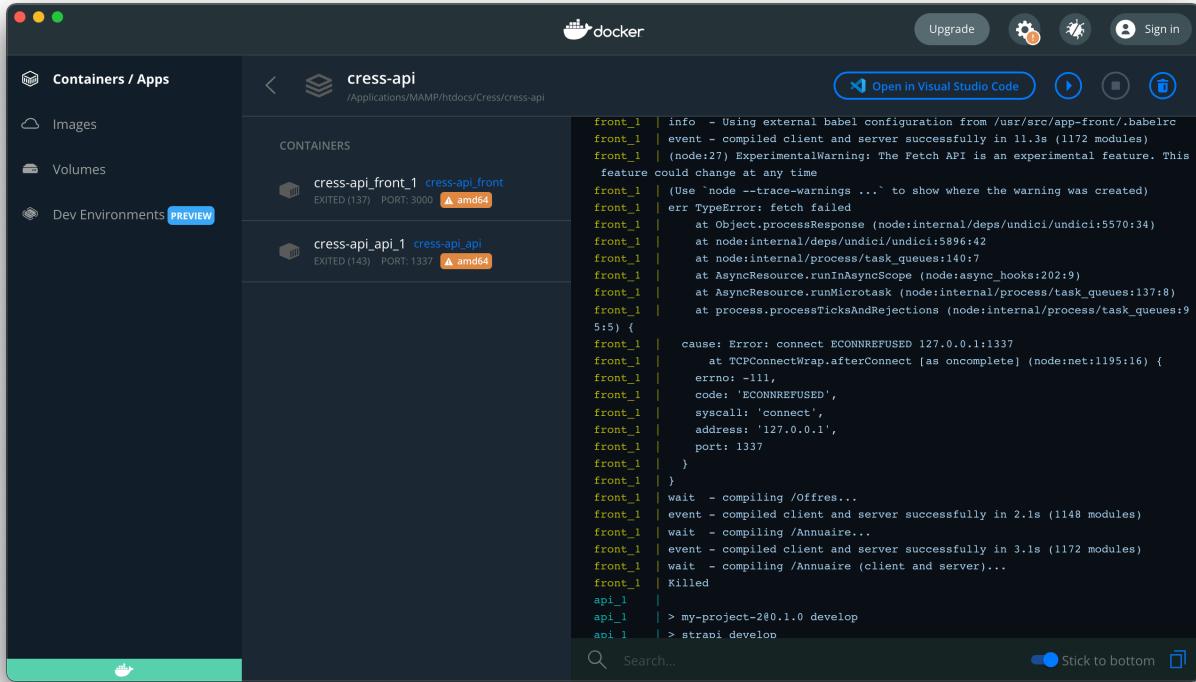
    const entity = await strapi.service("api::company.company").findOne(id, {
      populate: [
        "addresses",
        "social_network_link",
        "keywords",
        "users",
        "users.role",
        "events",
        "events.user",
        "events.tags",
        "users.events",
      ],
    });
    const sanitizedEntity = await this.sanitizeOutput(entity, ctx);
    sanitizedEntity.users = await populateUsers([
      "role",
    ], sanitizedEntity.users, this);
    sanitizedEntity.events = await Promise.all(
      sanitizedEntity.events.map(async (event) => ({
        id: event.id,
        ...event,
        user: await populateUsers(["role"], event.user),
      }))
    );
    return entityWithoutPw;
  }
})

```

Review

Livrable(<http://www.college-ibnkhaldoun.com/>)

Ci-dessous la capture d'écran du docker où on execute le test de déploiement de projet.



```
front_1 |   info - Using external babel configuration from /usr/src/app-front/.babelrc
front_1 |   event - compiled client and server successfully in 11.3s (1172 modules)
front_1 |   (node:27) ExperimentalWarning: The Fetch API is an experimental feature. This
feature could change at any time
front_1 |   (Use `node --trace-warnings ...` to show where the warning was created)
front_1 |   err TypeError: fetch failed
front_1 |       at Object.processResponse (node:internal/deps/undici/undici:5570:34)
front_1 |       at node:internal/deps/undici/undici:5896:42
front_1 |       at node:internal/process/task_queues:140:7
front_1 |       at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
front_1 |       at AsyncResource.runMicrotask (node:internal/process/task_queues:137:8)
front_1 |       at process.processTicksAndRejections (node:internal/process/task_queues:9
5:5) {
front_1 |     cause: Error: connect ECONNREFUSED 127.0.0.1:1337
front_1 |         at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1195:16) {
front_1 |           errno: -111,
front_1 |           code: 'ECONNREFUSED',
front_1 |           syscall: 'connect',
front_1 |           address: '127.0.0.1',
front_1 |           port: 1337
front_1 |         }
front_1 |       }
front_1 |       wait - compiling /Offres...
front_1 |       event - compiled client and server successfully in 2.1s (1148 modules)
front_1 |       wait - compiling /Annuaire...
front_1 |       event - compiled client and server successfully in 3.1s (1172 modules)
front_1 |       wait - compiling /Annuaire (client and server)...
front_1 |       Killed
api_1 |
api_1 | > my-project-2@0.1.0 develop
api_1 | > strapi develop
```

Ci-dessous la capture d'écran où j'ai fait une requête avec la méthode fetch de cress-api et où les réponses apparaissent filtrées.

```
134     business_sector,
135     structural_type,
136     addresses,
137     i_am,
138   },
139   pagination: {
140     start: 0,
141     limit: 10,
142   },
143 },
144 {
145   encodeValuesOnly: true, // prettify url
146 }
147 );
148
149 const FilterUrl = `http://localhost:1337/api/companies?populate=*&${query}`;
150
151 const response = await fetch(FilterUrl);
152 const res = await response.json();
153 console.log("res", res);
154
155 setAssurance([...res.data]);
156 };
157 // console.log("query", searchPoint(query));
158 useEffect(() => {
159   searchFilter();
160 }, [business, structural, department, i_am, search]);
161
162 console.log("i'am", i_am);
163
164 // function handling the change of an event and keeping the both the previous and the current event made
165 const handleBusiness = (e, data) => {
166   console.log("data", data);
167   if (e.target.checked) {
168     setBusiness((currentState) => [...currentState, data]);
169   } else {
170     setBusiness(business.filter((buss) => buss !== data));
171   }
172 }
```

Ci-dessous la capture d'écran d'une requête avec la méthode fetch de cress-api avec réponses filtrées et affichage des résultats en UI.

The screenshot shows a web browser displaying the website of the Chambre Régionale des entreprises de l'économie sociale et solidaire (Provence-Alpes-Côte d'Azur). The header includes the logo, the name of the chamber, and navigation links: Accueil, Notre offre, Annuaire, Evènements, Actualités, Contact, and Se connecter. A search bar at the top has the word "Assurance" typed into it, with a magnifying glass icon to its right. Below the search bar, a large image of a person's hands holding a smartphone is displayed. The main content area is titled "Annuaire". A search result summary "7 Résultats pour «Assurance»" is shown. On the left, there is a sidebar with a "Votre recherche avancé..." search bar and a "Secteur d'activité" section containing checkboxes for various sectors: Tourisme, Culture, Insertion, Sport, Médico-social, Habitat, Economie circulaire, and Environnement. The main results list shows two entries for MACIF, each with a logo, a brief description, a "Contacter" button with a phone number, and a "Delete" link. At the bottom of the results, there is a file download summary for "webdesign-cress-paca.pdf" with details: Added 14 Mar at 10:46 - 2.73 MB, MACIF Assurance en ligne, and a link to open the file in a new tab.

Chambre Régionale
des entreprises
de l'économie sociale
et solidaire
Provence-Alpes-Côte d'Azur

Accueil Notre offre Annuaire Evènements Actualités Contact

Se connecter

Assurance

Annuaire

7 Résultats pour «Assurance»

Votre recherche avancé...

Secteur d'activité

Tourisme
 Culture
 Insertion
 Sport
 Médico-social
 Habitat
 Economie circulaire
 Environnement

MACIF
Assurance en ligne
Essentiel pour moi : assurance auto, assurance habitation, assurance santé, crédit, assurance vie... 03 20 54 21 85

MACIF
Assurance en ligne
Essentiel pour moi : assurance auto, assurance habitation, assurance santé, crédit, assurance vie... 03 20 54 21 85

Added 14 Mar at 10:46 - 2.73 MB **MACIF**
Assurance en ligne
Essentiel pour moi : as-
z Open in new tab

Compétences du référentiel couvertes par le projet

Pour l'activité type 2 « **Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurités** » :

- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile

Exemple n° 2

Cadre du projet

Junior(api)-Application mobile

Le Projet

Développer une partie Back-end pour une application mobile qui stocke la tâche à rentrer dans une base de données et qui permet d'exécuter CRUD. Les tâches ont été réalisées en respectant le Object-Oriented Programming (OOP).

Le projet exposé dans ce dossier a été développé pendant la période du 01 mars au 30 mars 2021.

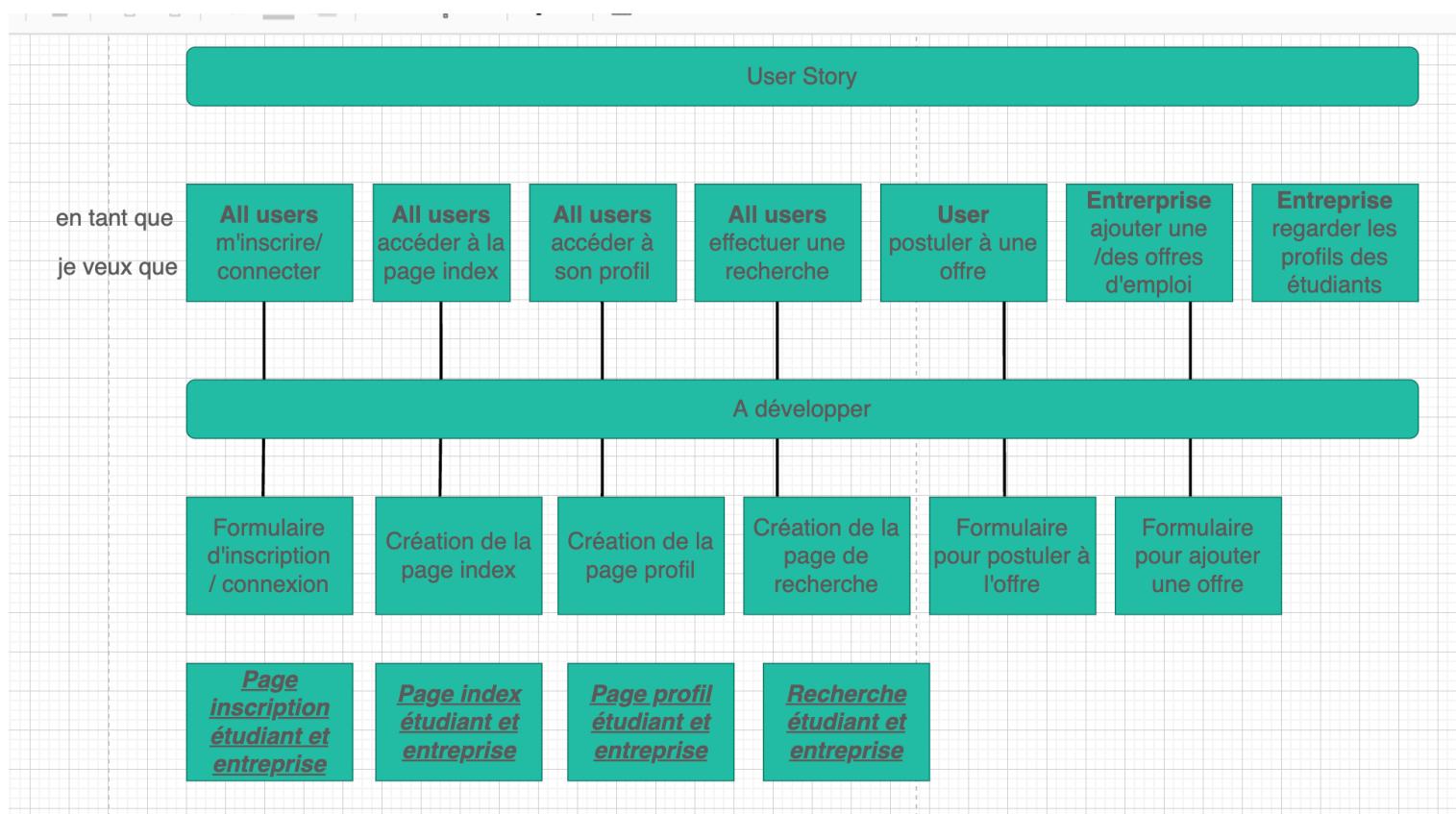
Planification

Ce projet a été réalisé en travail collaboratif en respectant la bonne pratique de développement d'un API rest. On arrive à diviser les tâches selon les compétences d'après le mode agile.

NOTE : Le tâche à donner en estimation de point sur chaque tâche

On a utilisé le framework symfony et Api-plateforme pour réaliser ce projet.

Product Backlog



Outils Utiliser

Draw oj : Organiser le user story et faire les prototypes de donnée en mode object-oriented Programming (oop) en création de mode MCD, MLD, MPD.

Trello : Organiser et faire une estimation de compétences pour chaque tâche avec points de complexité.

Api-Plateform : C'est un framework web utilisé pour générer des API REST et GraphQL, se basant sur le patron de conception MVC. La partie serveur du framework est écrite en PHP et basée sur le framework Symfony.

Git : Collaboration de projet.

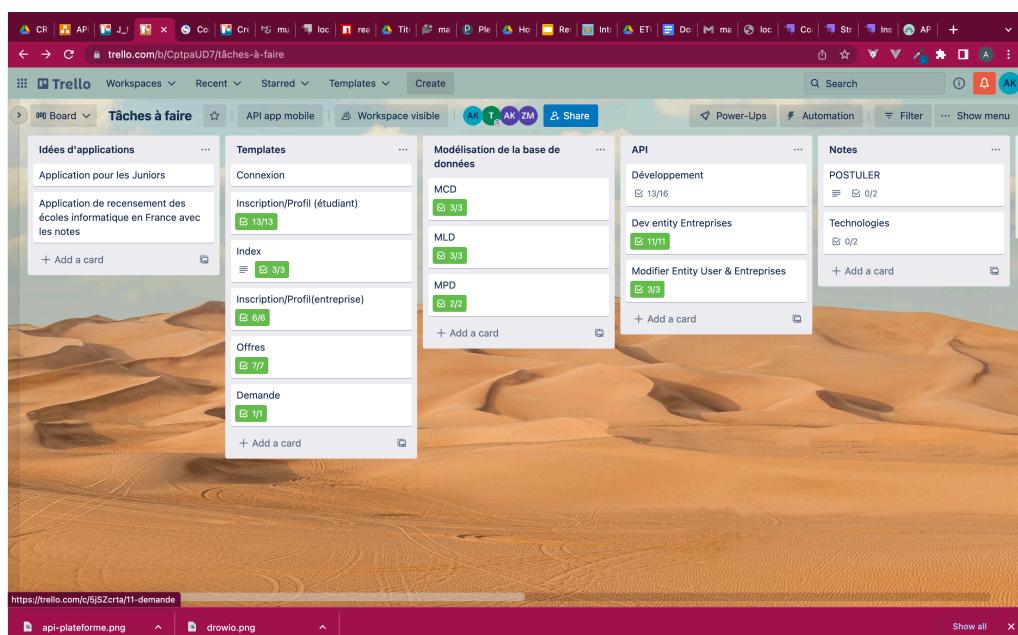
Swagger : C'est un langage de description d'interfaces permettant de décrire des API RESTful exprimées à l'aide de JSON.

Postman : C'est une application permettant de tester des API.

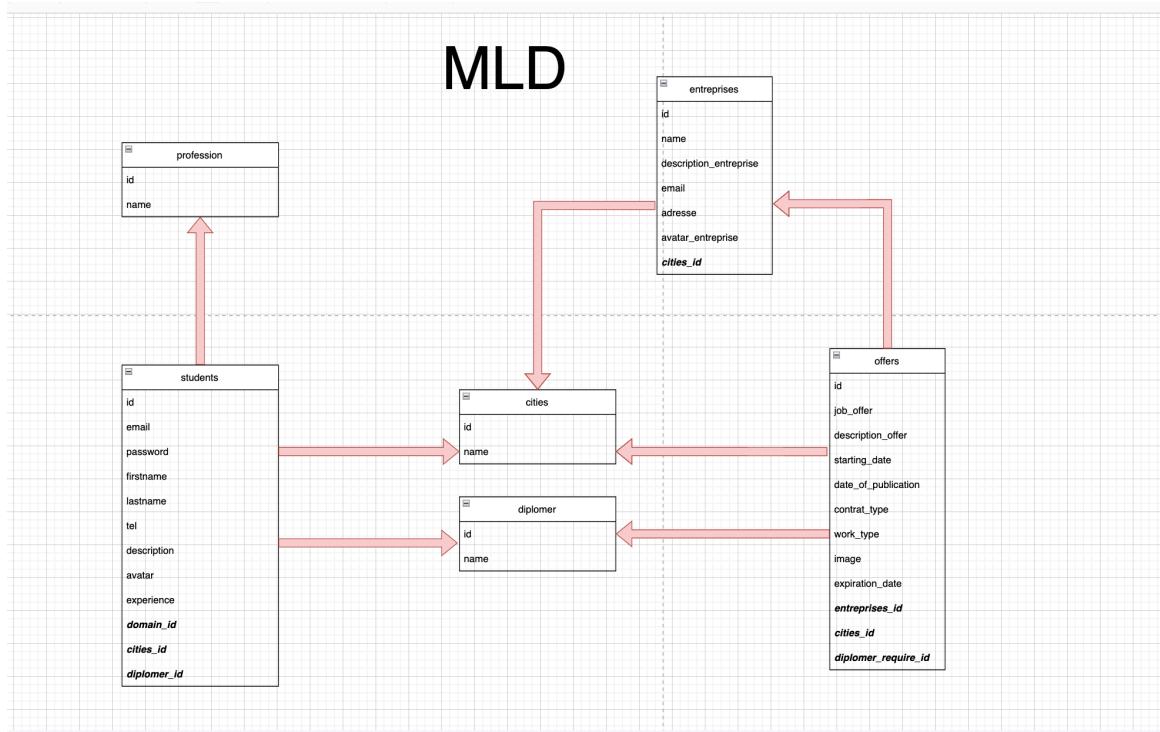
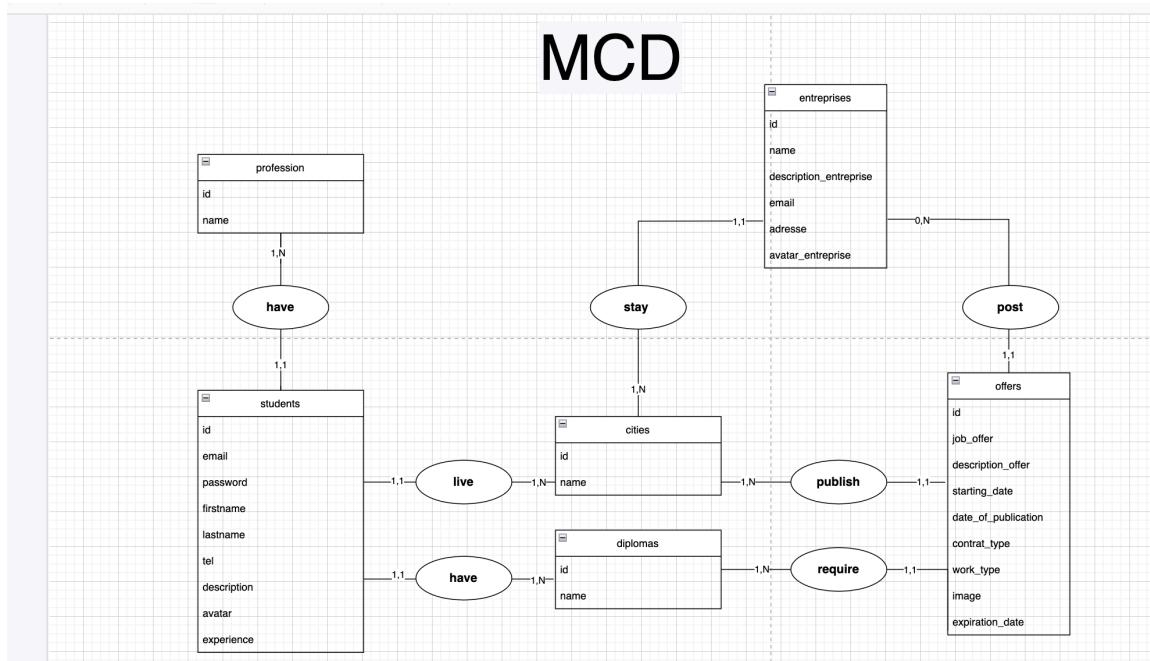
Plesk : Déploiement de Api.

Work Scheme

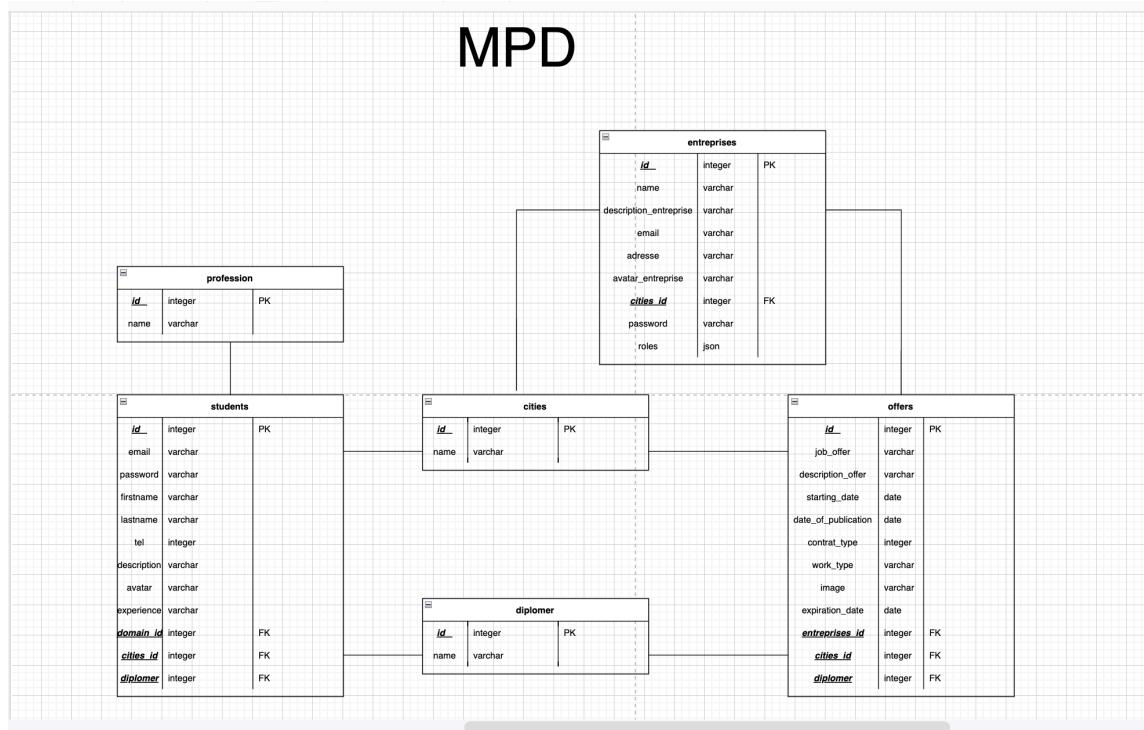
Ci-dessous la capture d'écran du trello d'organisation de mon travail par rapport à l'importance des tâches.



Ci-dessous la capture d'écran de mon drawio avec les conception de nos données en model MCD, MLD, MPD.



MPD



Itération

Installation de framework symfony avec api-platform avec utilisation de la documentation (<https://api-platform.com>) avec les commandes suivantes :

Symfony new Junior-api

Symfony composer require api

Symfony console doctrine:database:create

Symfony console doctrine:schema:create

Symfony serve

. Création des documents swagger avec les routers

. Création de entity et de la relation entre les entity

bin/console make:entity

php bin/console make:migration

php bin/console doctrine:migrations:migrate

avec SQL(junior-api)

. Création de router pour afficher les pages

. Documentation de l'api avec swagger

. Faire l'authentification avec JWT

. Fait les faire le test CRUD pour toute avec postman ???

. Faire les tests (Test Unitaire)

. Déploiements de Api

Ci-dessous la capture d'écran de la création de l'entity de l'api junior (entity user).

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows the project structure under "JUNIOR". The "src" directory contains "Entity" which has "User.php".
- Context.js:** A file in the background.
- User.php:** The code for the User entity. It includes annotations like `@ORM\Column(type: 'integer')` for the `$id` field, `@Vich\UploadableField(mapping="user_picture", fileNameProperty="photoFile")` for the `$photoFile` field, and `#Groups(["item"])` for the `$email` field. It also includes validation logic for the email field.
- TERMINAL:** Shows a PHPUnit test run with 1 test and 1 assertion passed.
- OUTPUT:** Shows the command line output of the test run.
- DEBUG CONSOLE:** Shows memory usage and other debug information.
- PROBLEMS:** Shows no problems.
- Find the documentation at https://symfony.com/doc/current/testing.html#unit-tests**
- PHPUnit 9.5.13 by Sebastian Bergmann and contributors.**
- Testing:** Shows the test results: 1 / 1 (100%) OK (1 test, 1 assertion)
- Time:** 00:00.011, Memory: 10.00 MB
- Ln 1, Col 1 Spaces: 4 UTF-8 LF PHP Go Live Prettier**

Ci-dessous la capture d'écran de ma base de données (MSQL) avec toutes mes tables.

The screenshot shows the phpMyAdmin interface for the "JuniorVersion01" database. The left sidebar lists the following tables:

- cities
- diplomas
- doctrine_migration_versions
- entreprises
- offers
- profession
- refresh_tokens
- user

The main area displays the structure of the "user" table:

Table	Action	Rows	Type	Collation	Size	Overhead
cities	Browse Structure Search Insert Empty Drop	48	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
diplomas	Browse Structure Search Insert Empty Drop	38	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
doctrine_migration_versions	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_unicode_ci	16.0 Kib	-
entreprises	Browse Structure Search Insert Empty Drop	52	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
offers	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	64.0 Kib	-
profession	Browse Structure Search Insert Empty Drop	38	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
refresh_tokens	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
user	Browse Structure Search Insert Empty Drop	56	InnoDB	utf8mb4_unicode_ci	80.0 Kib	-

Below the table structure, there is a "Create table" form with fields for "Name:" and "Number of columns:" set to 4, and a "Go" button.

Ci-dessous la capture d'écran de notre création de router et la documentation de l'api avec swagger.

The screenshot shows the API Platform interface for the 'Profession' resource. It lists various HTTP methods and their corresponding endpoints:

- PUT** /api/offers/{id} Replaces the Offers resource.
- DELETE** /api/offers/{id} Removes the Offers resource.
- PATCH** /api/offers/{id} Updates the Offers resource.

Profession

- GET** /api/professions Retrieves the collection of Profession resources.
- POST** /api/professions Creates a Profession resource.
- GET** /api/professions/{id} Retrieves a Profession resource.
- PUT** /api/professions/{id} Replaces the Profession resource.
- DELETE** /api/professions/{id} Removes the Profession resource.
- PATCH** /api/professions/{id} Updates the Profession resource.

Token

- POST** /authentication_token Get JWT token to login.

At the bottom, there is a navigation bar with icons for Plesk, Postman, Swagger, API Plateforme, and Drowio, along with a 'Show all' link.

The screenshot shows the API Platform interface for the 'User' resource. It lists various HTTP methods and their corresponding endpoints:

- POST** /api/entreprises/{id} Creates a Entreprises resource.
- DELETE** /api/entreprises/{id} Removes the Entreprises resource.
- PATCH** /api/entreprises/{id} Updates the Entreprises resource.
- GET** /api/my Retrieves the collection of Entreprises resources.
- POST** /api/register_company Creates a Entreprises resource.

User

- GET** /api/me Retrieves the collection of User resources.
- POST** /api/register_user Creates a User resource.
- GET** /api/users/{id} Retrieves a Offers resource.
- POST** /api/users/{id} Creates a User resource.
- DELETE** /api/users/{id} Removes the User resource.
- PATCH** /api/users/{id} Updates the User resource.

At the bottom, there is a navigation bar with icons for Plesk, Postman, Swagger, API Plateforme, and Drowio, along with a 'Show all' link.

Ci-dessous la capture d'écran de notre mise en oeuvre de JWT authentification pour l'api.

```

JUNIOR_APP_V1/.env
API_Junior > .env # DATABASE_URL="mysql://toreapat:041219Tp&@127.0.0.1:3306/torea-patissier_junior?serverVersion=5.5"
24
25 ###> doctrine/doctrine-bundle ##
26 # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/
27 # configuration.html#connecting-using-a-uri
28 #
29 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
30
31
32 # DATABASE_URL="sqlite:///kernel.project_dir/var/data.db"
33 DATABASE_URL="mysql://root:root@127.0.0.1:8889/JuniorVersion01?serverVersion=5.7"
34
35
36 #DATABASE_URL="postgresql://symfony:ChangeMe@127.0.0.1:5432/app?serverVersion=13&charset=utf8"
37 ####< doctrine/doctrine-bundle ##
38
39 ####> nelmio/cors-bundle ##
40 CORS_ALLOW_ORIGIN="https://(localhost|127\\.0\\.0\\.1)(:[0-9]+)?"
41 ####< nelmio/cors-bundle ##
42
43 ####> lexik/jwt-authentication-bundle ##
44 JWT_SECRET_KEY=%kernel.project_dir%/config/jwt/private.pem
45 JWT_PUBLIC_KEY=%kernel.project_dir%/config/jwt/public.pem
46 JWT_PASSPHRASE=admin
47 JWT_TTL=3600
48 ####< lexik/jwt-authentication-bundle ##

```

Ci-dessous la capture d'écran de entity user avec l'exécution de l'authentification avec JWT authentication (api-platefome) pour hâcher le mode passe.

	Edit	Copy	Delete	id	city_id	profession_id	diploma_id	updated_at	email	roles	password
<input type="checkbox"/>	Edit	Copy	Delete	7	NULL	NULL	NULL	NULL	Let@gmail.com		\$2y\$13\$q0yJF/hjatPyggILbVaBuc3dDxVzD.EVdwVS1BY
<input type="checkbox"/>	Edit	Copy	Delete	12	2	2	2	2022-06-15 08:22:20	rahman@gmail.com	[{"ROLE_USER"]	\$2y\$13\$6W1dRwlFs.a.howrU91YC.oKxYUGkRouK.cOwPv
<input type="checkbox"/>	Edit	Copy	Delete	63	7	7	7	NULL	kiki@gmail.fr		\$2y\$13\$7flIP15l1JrCtywTMDD.nsQzcrM/ZvhTwnoSpMek
<input type="checkbox"/>	Edit	Copy	Delete	68	12	12	12	2022-06-15 12:42:11	mamam@gmail.com		\$2y\$13\$F68sSFrIBMkNnjXdrJ.mnMrRS/hsPv4neB4R
<input type="checkbox"/>	Edit	Copy	Delete	73	20	15	20	2022-06-15 12:50:48	kakak@gmail.fr		\$2y\$13\$NstuEC3/gH00yAOCoPpJ.upld0Myh3.trVuyHcx7G
<input type="checkbox"/>	Edit	Copy	Delete	76	22	22	22	2022-06-15 13:11:50	ugh@gmail.fr		\$2y\$13\$TdL3rUKLD5Gj3JBVrl/bQ./7TxIMDTs9msclJff3GW
<input type="checkbox"/>	Edit	Copy	Delete	101	23	23	23	2022-06-16 07:20:39	work@gmail.com	[{"ROLE_USER"]	\$2y\$13\$LGHULI9cn4xCACgFuug.Xqu.wDsseqpSXE32B49C
<input type="checkbox"/>	Edit	Copy	Delete	102	NULL	NULL	NULL	NULL	see@gmail.com		\$2y\$13\$sgdGrkzwcdOajhusJm0NGwEeEpibO4CukUlxpOf
<input type="checkbox"/>	Edit	Copy	Delete	103	NULL	NULL	NULL	NULL	meseee@gmail.com		\$2y\$13\$wq2N8ICWab4kAwFjluccVu312qoYngQxNoYKBkJ
<input type="checkbox"/>	Edit	Copy	Delete	104	NULL	NULL	NULL	NULL	myfr@gmail.com		\$2y\$13\$cX6wwWbSr9mtT9AL8Quq7udy2kRRXkS30thw
<input type="checkbox"/>	Edit	Copy	Delete	106	NULL	NULL	NULL	NULL	alexi@gmail.com		\$2y\$13\$OXCYJRMILBb32lBDjjo.BCGJjezfbQ.YA11.hx..
<input type="checkbox"/>	Edit	Copy	Delete	107	NULL	NULL	NULL	NULL	emai@gmail.com		\$2y\$13\$v7Nhd.4gxPKYWhjEMnsUOYiqRpnczplc9iH8C
<input type="checkbox"/>	Edit	Copy	Delete	108	NULL	NULL	NULL	NULL	szcpkczck@gmail.fr		\$2y\$13\$adRFhZG.OB61Tojn06aTek0t5y3GB4bV6lRFcxHq
<input type="checkbox"/>	Edit	Copy	Delete	109	NULL	NULL	NULL	NULL	theo@hernandez.com		\$2y\$13\$y5iuxpnHrW8Su4FwGoDs.UqGLNQqrFZuCzujK9
<input type="checkbox"/>	Edit	Copy	Delete	110	NULL	NULL	NULL	NULL	ihefefgef@gmail.lrf		\$2y\$13\$10lFxFxbjZMYYCM5mEA.vDy.hGhbq3gttlhcaqaF
<input type="checkbox"/>	Edit	Copy	Delete	111	NULL	NULL	NULL	NULL	inth@gmail.fr		\$2y\$13\$vgKomh4nzddus4AjB8r4qO9SyWwUdM90LukA15s
<input type="checkbox"/>	Edit	Copy	Delete	112	NULL	NULL	NULL	NULL	lookme@gmail.fr		\$2y\$13\$.Oosir1kZbKVu8CgXNW1X.zbuji3QnzzOLpk78gmr
<input type="checkbox"/>	Edit	Copy	Delete	113	NULL	NULL	NULL	NULL	ahcen@example.com	{"ROLE_USER"]	\$2y\$13\$8xj/N8v0e1gVQN69.kmReO1mA1vnybybbIZOVInqs
<input type="checkbox"/>	Edit	Copy	Delete	114	NULL	NULL	NULL	NULL	see@gmail.fr		\$2y\$13\$Kmp0ObBelnCqMfWGVfleF5dMwtbSKJhrF9W
<input type="checkbox"/>	Edit	Copy	Delete	115	NULL	NULL	NULL	NULL	shiva@gmail.com		\$2y\$13\$zqXu4PmfHneumeZlRvJU1.LDLejyk1rdRyuJrV2l
<input type="checkbox"/>	Edit	Copy	Delete	116	NULL	NULL	NULL	NULL	djdzd@gmail.com		\$2y\$13\$KieWG786/yzTaYhsG86veGvx5wUrJEtAigAieWwr
<input type="checkbox"/>	Edit	Copy	Delete	117	25	25	25	2022-06-16 09:52:46	bjdzdbjz@gmail.fr		\$2y\$13\$AD/t4s4PaGkL2iSb8yQWROkR3RJLesHaxqMLth3

Ci-dessous la capture d'écran des résultats des tests test de l'api (junior-api) sur postman.

```

POST https://127.0.0.1:8000/authentication_token
{
  "email": "work@gmail.fr",
  "password": "password"
}
{
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Ni9...",
  "refreshToken": "7bb28b756e26dbe262277a...2966cBeb0"
}
    
```

Ci-dessous la capture d'écran du Test Unitaire sur symfony avec api-platform.

```

Context.js .phpunit.result.cache
1 {"version":1,"defects":[],"tests":{"App\Tests\SpamCheckerTest::testSomething":0.003}}


Package operations: 0 installs, 0 updates, 1 removal
- Removing symfony/test-pack (1.0.10)
Generating optimized autoloader files
composer/package-versions-deprecated: Generating version class...
composer/package-versions-deprecated: ...done generating version class
14 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

Run composer recipes at any time to see the status of your Symfony recipes.

Info from https://repo.packagist.org: #StandWithUkraine
Executing script cache:clear [OK]
Executing script assets:install public [OK]

→ API_junior git:(AhceneAPIS) × symfony console make:test TestCase SpamCheckerTest
created: tests/SpamCheckerTest.php

Success!
    
```

Review

Plesk est une interface de gestion de serveur payante, mais j'ai pu y avoir un accès gratuit en tant qu'étudiant à La Plateforme. J'ai donc décidé d'y déployer mon API dans le cadre de mon projet d'application mobile.

Pour cela j'ai procédé comme suit :

- J'ai ajouté mon dépôt git sur plesk.
- J'ai choisi le mode de déploiement automatique, cela signifie que toutes les modifications qui ont lieu sur mon dépôt git sont déployées automatiquement.
- Pour la connexion au dépôt git distant j'ai créé une clé publique SSH.

Code location

 **Remote repository**
Your code is hosted online (a cloud service like GitHub, GitLab, or Bitbucket, or your own server).
Plesk will pull code from there.

 **Local repository**
Your code is on the computer to which Plesk wouldn't be able to connect. For example it's your laptop or some server unavailable outside of your LAN.
You will push code to the server with Plesk yourself.

Repository URL *

Both HTTP(S) and SSH protocols are supported

SSH public key

```
ssh-rsa
AAAAAB3NzaC1yc2EAAAQABAAQAC3uOq8WvMPDckExgfCwmbEwy
Can56qcM4US9Tl+bWj1lnnQ8NreDKUzRKh3HRMkl7ZIS9+ccGFBK3wXn
Dhm2CEw9G8NyHVi+KUpSseCpKcqFDns3NNoZT2Etjo4jfgNuPODEINpp
```

This is the public part of the SSH key used for authorization in the remote repository. It must be added to the remote service.

Repository name *

Specify a name that is unique within a domain.

Deployment settings

Deployment mode *

Automatic Manual Disabled

Files will be deployed to the production site as soon as they are available in the Plesk repository.

Server path *

/htdocs 

Directory on the server where files will be deployed.

Enable additional deployment actions

Specify shell commands to be run every time upon deployment.

api.torea-patissier.students-laplateforme.io • Active

Website at api.torea-patissier.students-laplateforme.io/public/ IP address: 93.90.200.103 System user: torea-patissier

[Hosting Settings](#) [Open in web](#) [Preview](#) [Description](#) [Move domain](#)

 File Manager  Mail  Databases [torea-patissier_API](#) [Open](#)

 API_junior.git Branch: toreaV2; Automatic deploy to [/api.torea-patissier.students-laplateforme.io](http://api.torea-patissier.students-laplateforme.io) [Pull Updates](#)

[Show Less](#)

 Web Hosting Access	 FTP Access	 Hosting Settings
 PHP Composer	 SSL/TLS Certificates Security can be improved	 Git Enabled
 PHP Settings Version 8.0.20	 Applications	 Node.js
 File Manager	 Web Statistics SSL/TLS	 DNS Settings
 Mail Settings	 Password-Protected Directories	 Website Copying
 Logs	 Web Users	 Remove Subdomain

Ci-dessous la capture d'écran de notre déploiement de l'api junior sur plesk

- Le nom de la Database
- Déploiement automatique
- Le nom de la branch sur laquelle l'API est déployé (toreaV2)

The screenshot shows a deployment interface for an API. At the top, there's a header with a cloud icon and the text "API_junior.git". Below it, there are two input fields: "URL" containing "git@github.com:torea-patissier/API_junior.git" and "Branch" set to "toreaV2". A "Pull now" button is located below these fields. Under "Latest commits", there are two entries: one from "2022-06-21 17:14" labeled "BUG" and another from "2022-06-21 16:31" labeled "me". There's also a "show more" link. In the "Deployment" section, it says "toreaV2 branch automatically to /api.torea-patissier.students-laplateforme.io" and has a "Deploy now" button. At the bottom right, there are three small icons: a trash can, a gear, and a circular arrow.

Compétences du référentiel couvertes par le projet

Pour l'activité type 3 “ **Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité** “

Exemple n°1

Cadre du projet

Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.

- Concevoir une application
- Développer des composants métier
- Construire une application organisée en couches
- Développer une application mobile
- Préparer et exécuter les plans de tests d'une application
- Préparer et exécuter le déploiement d'une application

Planification

Ce projet a été réalisé en travail collaboratif en respectant la bonne pratique de développement d'une application mobile en front-end avec react-native. On s'est divisé les tâches par rapport aux compétences d'après le mode agile.

NOTE : Estimation en points de la complexité de chaque tâche.

On a utilisé le framework javascript avec expo pour réaliser ce projet.

Cahier de charge

Cahier de Charge

JUNIOR APPLICATION

IDEA

As we know there are various mobile applications that may perform the same function as Junior. As developers it always good to create something that answers the need of the client and the world at large. Junior App came just as what we thought as the little missing piss in job application. Its always heard for a junior developer to find there first job(stage, Alternance, CDD, CDI) in other to gain and improve the skills in web and application development. We this the idea of creation and application that serve the need of junior developer in helping us find our first experience in a company or a project. The aim is to provide job with all the junior developers within 0 to 3 years of professional experience

User Story

The User Story Map diagram illustrates the flow of requirements. At the top, a 'User Story' box contains user needs: 'All users accéder à la page index', 'All users accéder à son profil', 'All users effectuer une recherche', 'User postuler à une offre', 'Entreprise accéder une offre', and 'Entreprise inscrire à un profil des étudiants'. Below this, a 'A développer' box lists developer tasks: 'Formulaire d'inscription / connexion', 'Création de la page index', 'Création de la page profil', 'Création de la page de recherche', 'Formulaire pour postuler à l'offre', and 'Formulaire pour inscrire une offre'. At the bottom, a 'Page inscription étudiant et entreprise' box contains specific pages: 'Page index étudiant et entreprise', 'Page profil étudiant et entreprise', and 'Recherche étudiant et entreprise'.

PROJECT

The aim of Junior is to help reduce the gap and doubt between junior developer that just want to find their first experience to put their skill to test and improve as better developer and Company Recruiters, and Start-ups. Junior is bring you a meeting place for both the companies and the junior developers to have a direct contact. As the junior developer may have access to job opportunity that link to their level of skill and contact the company direct. Company that have posted a job offer can also view a junior develop profile and can contact him/her direct from the application.

Application Junior version 1(demo)

The application will be having 12 screen :

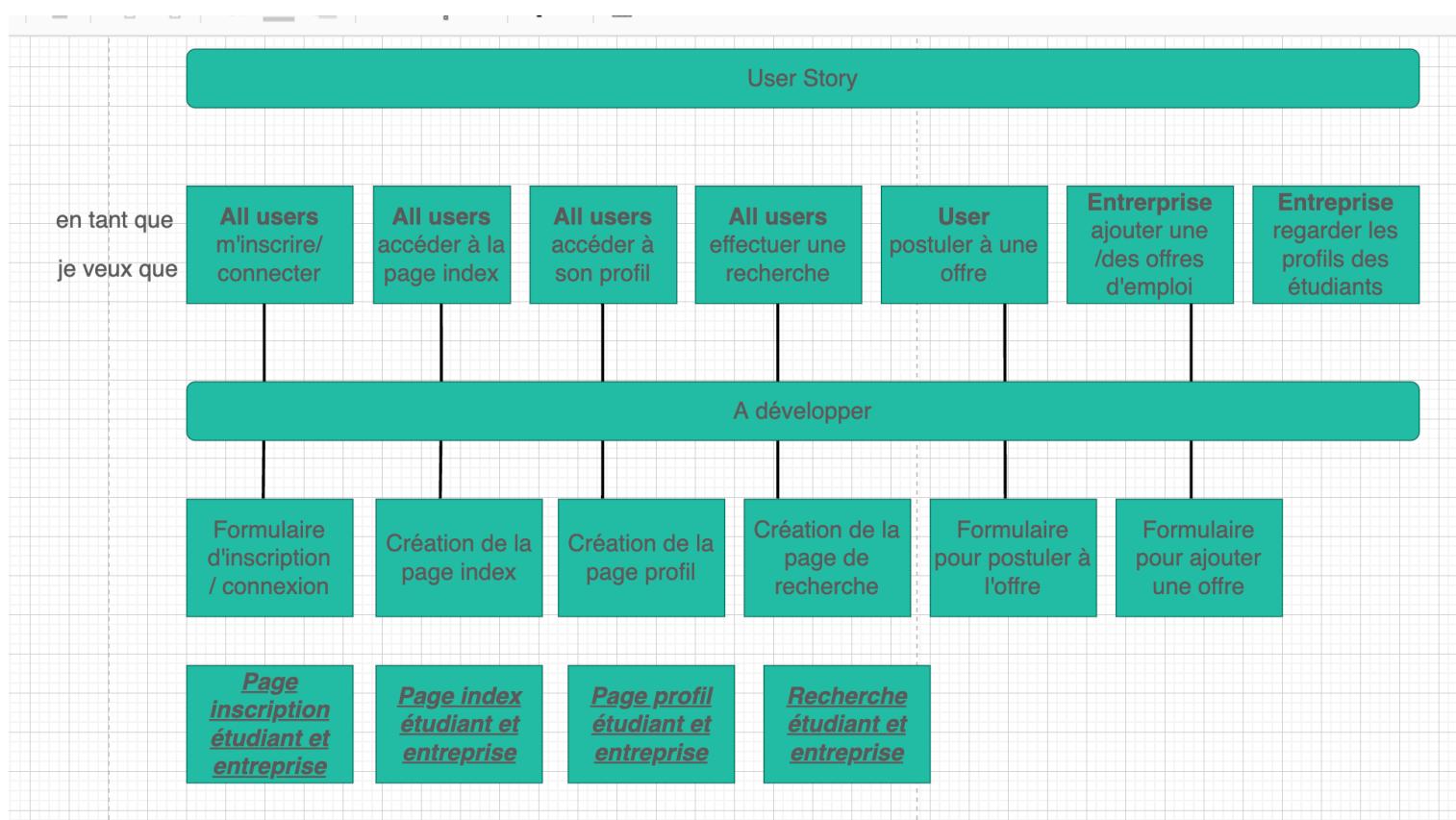
- Splash screen
- Login screen
- Welcome screen
- Registration screen junior
- Registration screen Company
- Create profile screen junior
- Create profile screen Company
- Dashboard
- Profile screen
- Details screen
- Apply screen

Splash screen : this is the entering screen when you first try to use the app. It's contains the logo of the application and also a welcoming message to Welcome you to the application.

Login screen : The application have only one login screen for both users. This screen is authenticated and the conditions set should be met in order to have access to the app. It's also contain a message if

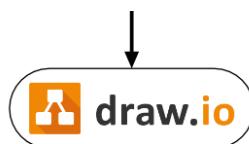
Product Backlog

Ci-dessous la capture d'écran de mon site avec la page qui permet de modifier la tâche.



Outils :

Drawio



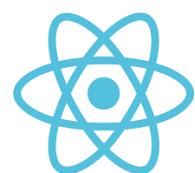
Trello



Figma



React-native.



Expo



Git



Xcode



Outils utilisés

Draw oi : Organiser le user story par rapport à la priorité et à l'ordre d'importance.

Trello : Organiser et faire une estimation de compétences pour chaque tâche avec points relatifs à la complexité.

Figma : Création de notre maquette (wireframe, maquette et prototyping).

React-native : C'est un framework d'applications mobiles open source créé par Facebook. Il est utilisé pour développer des applications pour Android, iOS et UWP en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes,

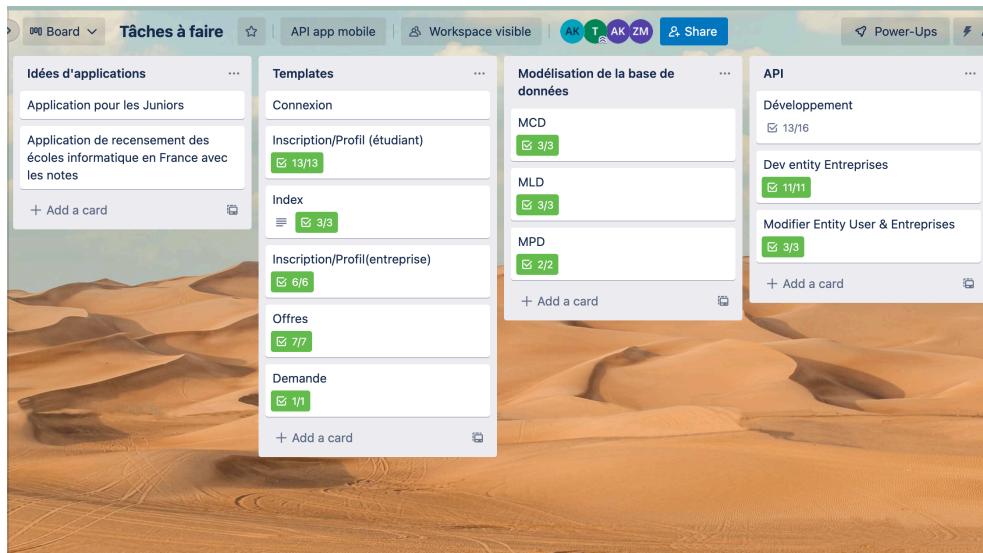
Expo : C'est un logiciel de développement plateform pour développer une application universelle qui fonctionne sur Ios et Andriod en utilisant javascript framework.

Git : Plateform de collaboration.

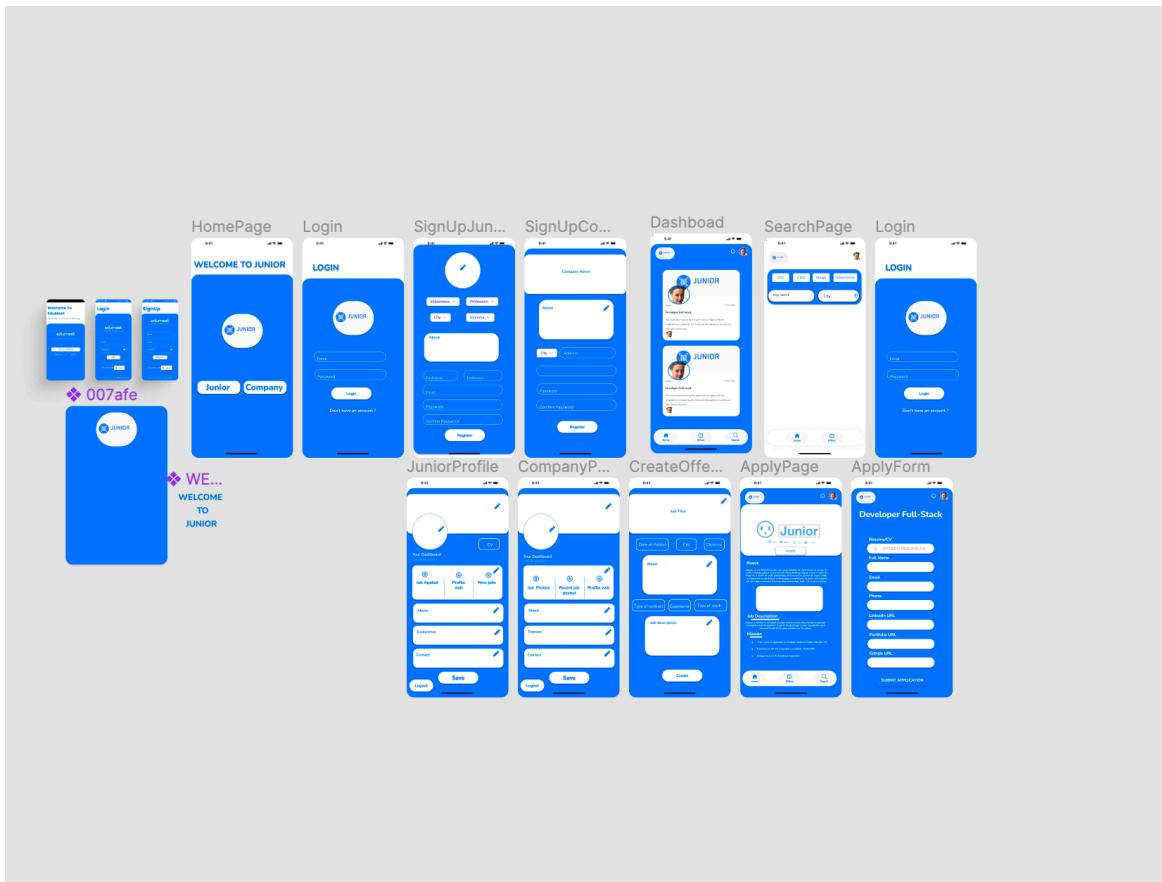
Xcode : Inclut tout ce dont les développeurs ont besoin pour créer de superbes applications pour Mac, iPhone, iPad, Apple TV et Apple Watch. Xcode fournit aux développeurs un flux de travail unifié pour la conception, le codage, les tests et le débogage de l'interface utilisateur. L'IDE Xcode combiné au langage de programmation Swift rend le développement d'applications facile et amusant.

Work Scheme

Ci-dessous la capture d'écran de mon trello qui organise mon travail par rapport à l'importance des tâches à réaliser (Junior App).

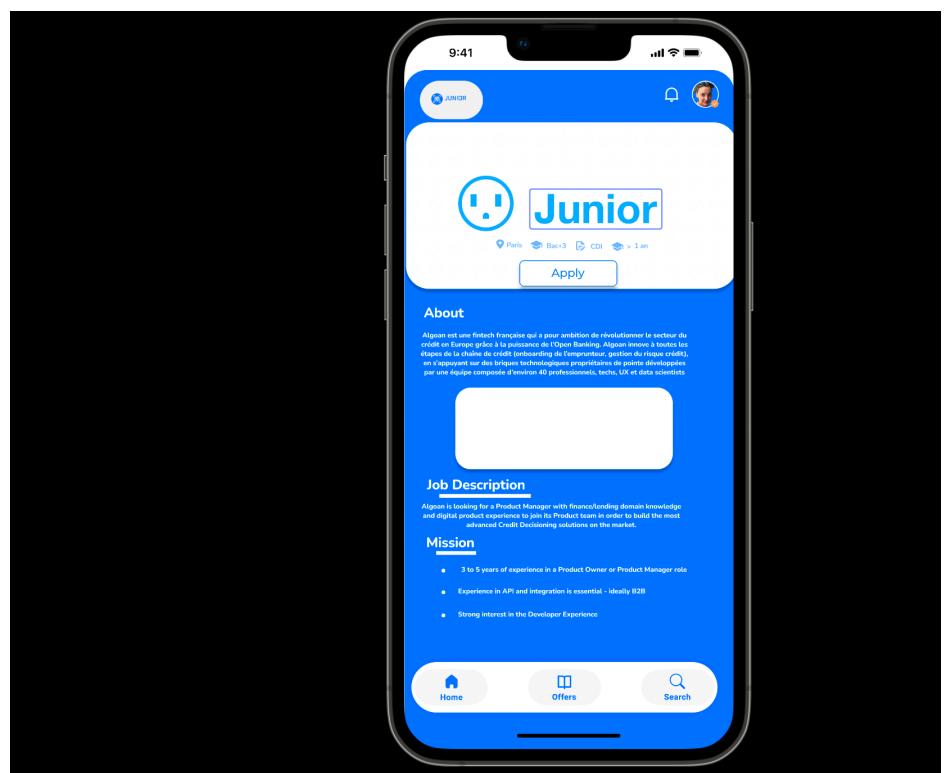
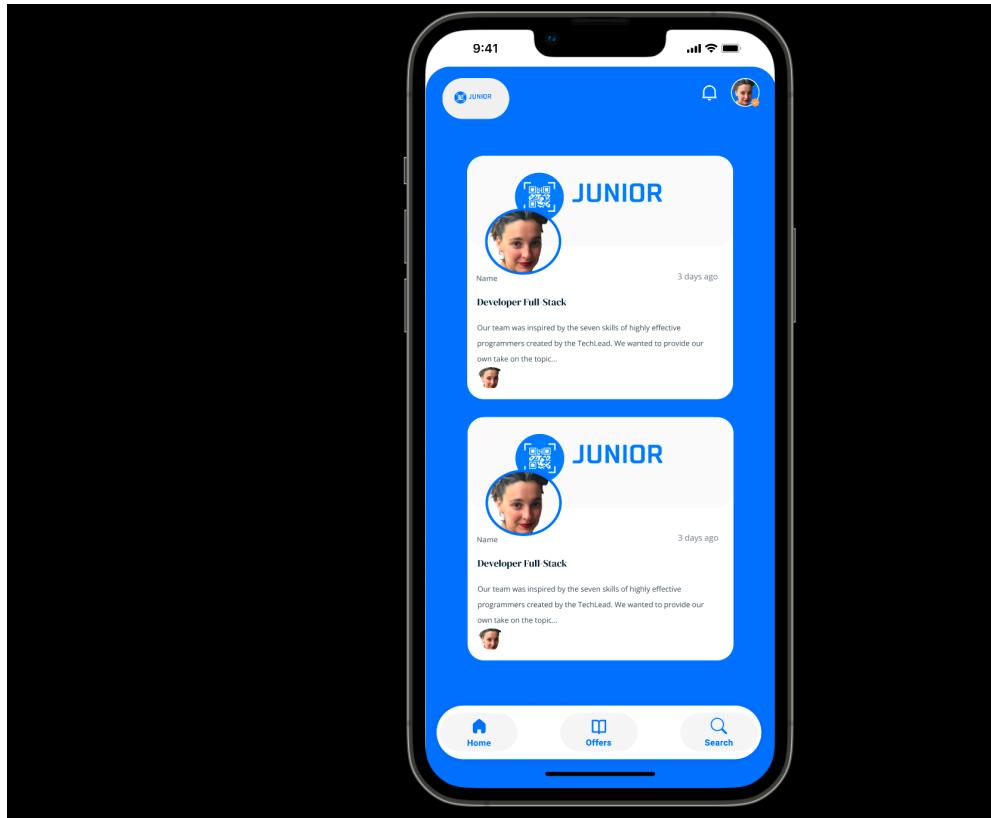


Ci-dessous la capture d'écran de mon figma qui montre mon travail sur la maquette (Junior App).



Ci-dessous la capture d'écran de ma maquette en prototype.

(<https://www.figma.com/proto/gZuM6gWaq8SUatqkEXvIF8/API?page-id=20%3A4269&node-id=32%3A6839&viewport=-289%2C181%2C0.17&scaling=scale-down&starting-point-node-id=32%3A6839&showproto-sidebar=1>)



Itération

Installation du framework react-native avec expo avec utilisation de la documentation à partir des (<https://docs.expo.dev/get-started/create-a-new-app/>) commandes suivantes :

expo init JuniorApp

cd JuniorApp

Expo start

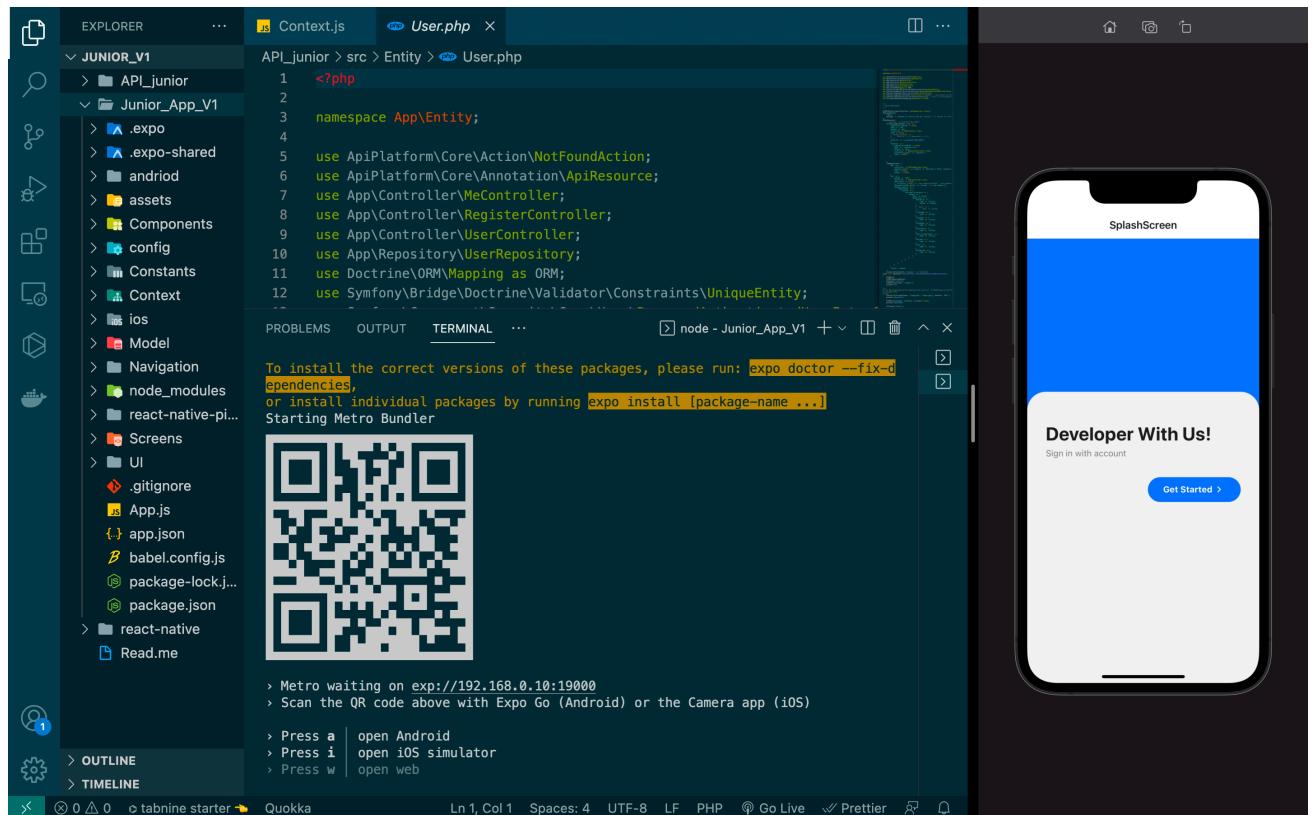
- . Création de composant, screen avec react-native
- . Création de navigation en utilisant react-navigation
- . Faire un appel de l'api pour connecter le front et le back de notre application (juniorApp)
- . Utiliser Xcode pour tester, debugger l'application (juniorApp)
- . Création de l'authentification avec useContext
- . Stockage des informations avec AsyncStorage
- . Tester l'application (juniorApp)

Ci-dessous la capture d'écran du lancement de juniorApp avec expo cli.

The screenshot shows a terminal window with the following output:

```
There is a new version of expo-cli available (5.4.11).  
You are currently using expo-cli 5.4.7  
Install expo-cli globally using the package manager of your choice;  
for example: `npm install -g expo-cli` to get the latest version  
  
Starting project at /Applications/MAMP/htdocs/juniorApp/React_native/Junior_v1/react-native/junior  
Developer tools running on http://localhost:19002  
Starting Metro Bundler  
  
QR code:  
  
  
> Metro waiting on exp://192.168.0.10:19000  
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)  
> Press a | open Android  
> Press i | open iOS simulator  
> Press w | open web  
> Press r | reload app  
> Press m | toggle menu  
> Press d | show developer tools  
> shift+d | toggle auto opening developer tools on startup (disabled)  
> Press ? | show all commands  
  
Logs for your project will appear below. Press Ctrl+C to exit.  
Started Metro Bundler  
iOS Bundling complete 8121ms  
iOS Running app on iPhone 13 mini
```

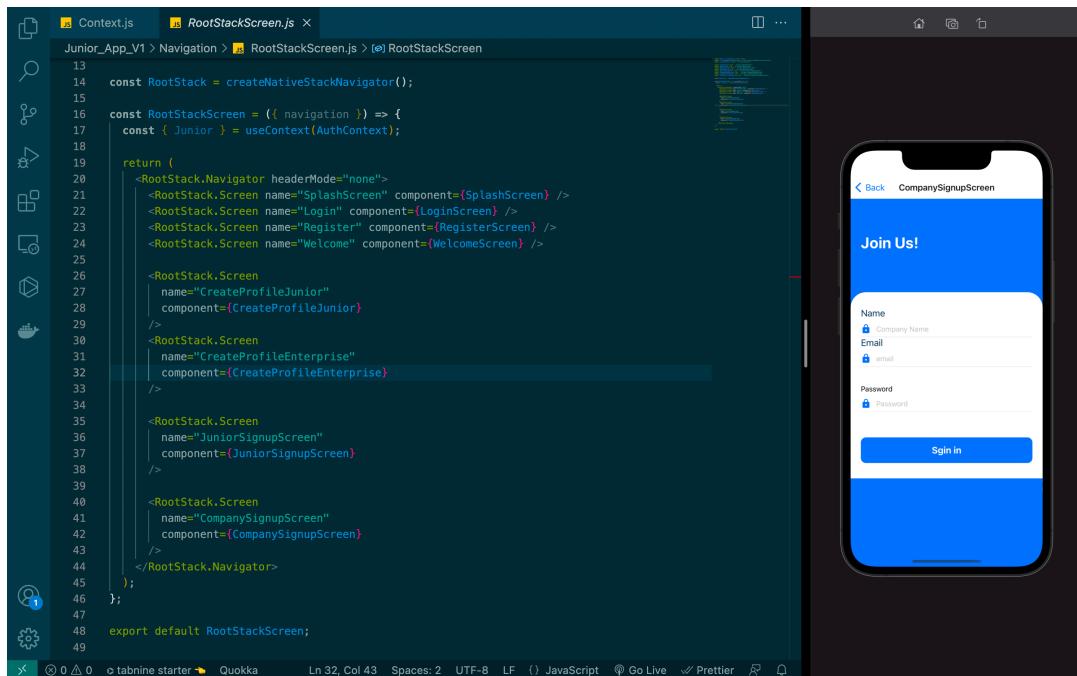
Ci-dessous la capture d'écran de test de notre application (juniorApp) en utilisant Xcode pour lancer l'application sur iPhone.



Ci-dessous la capture d'écran d'exécution de la react-navigation pour définir mes stack screen et utiliser les différentes navigation (Stack, Tab et Drawer navigation).

npm install @react-navigation/native

expo install react-native-screens react-native-safe-area-context



The screenshot shows a code editor with two tabs: Context.js and RootStackScreen.js. The code in RootStackScreen.js defines a stack navigator with several screens: SplashScreen, Login, Register, Welcome, CreateProfileJunior, CreateProfileEnterprise, JuniorSignupScreen, and CompanySignupScreen. To the right of the editor is a mobile application running on an iPhone X simulator. The app's splash screen displays the text "Join Us!" and a form for Name, Email, and Password, with a "Sgin in" button.

```
const RootStack = createNativeStackNavigator();

const RootStackScreen = ({ navigation }) => {
  const { Junior } = useContext(AuthContext);

  return (
    <RootStack.Navigator headerMode="none">
      <RootStack.Screen name="SplashScreen" component={SplashScreen} />
      <RootStack.Screen name="Login" component={LoginScreen} />
      <RootStack.Screen name="Register" component={RegisterScreen} />
      <RootStack.Screen name="Welcome" component={WelcomeScreen} />

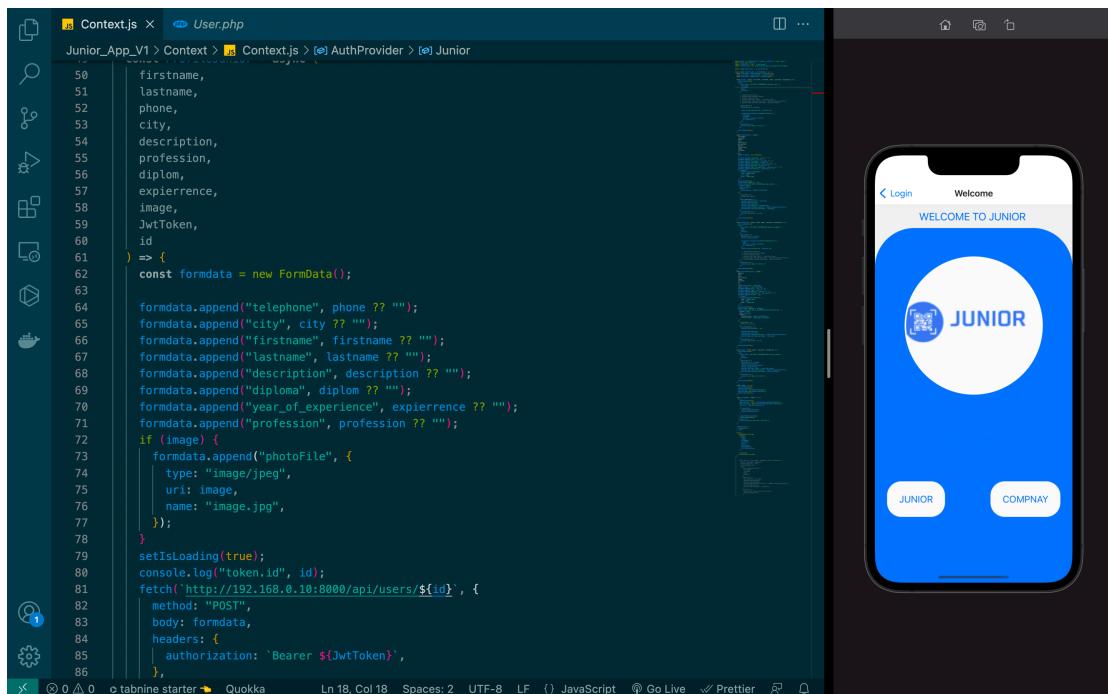
      <RootStack.Screen
        name="CreateProfileJunior"
        component={CreateProfileJunior}
      />
      <RootStack.Screen
        name="CreateProfileEnterprise"
        component={CreateProfileEnterprise}
      />

      <RootStack.Screen
        name="JuniorSignupScreen"
        component={JuniorSignupScreen}
      />

      <RootStack.Screen
        name="CompanySignupScreen"
        component={CompanySignupScreen}
      />
    </RootStack.Navigator>
  );
};

export default RootStackScreen;
```

Ci-dessous la capture d'écran d'exécution de useConext qui inclut toutes mes fonctions qui permettent d'appeler de l'api et de passer les variables sur tous les composants et l'écran.



The screenshot shows a code editor with two tabs: Context.js and User.php. The code in Context.js uses the useConext hook to manage user data like firstname, lastname, phone, city, profession, diplome, expirrience, image, JwtToken, and id. It also handles file uploads for the image field. To the right of the editor is a mobile application running on an iPhone X simulator. The app's splash screen displays the text "WELCOME TO JUNIOR" and features a QR code, the word "JUNIOR", and buttons for "JUNIOR" and "COMPANY".

```
const useConext = (initialState) => {
  const [state, setState] = useState(initialState);

  const handleLogout = () => {
    console.log("Logout");
    setState({ ...initialState });
  };

  const handleLogin = (data) => {
    console.log("Login", data);
    setState({ ...data });
  };

  const handleSignup = (data) => {
    console.log("Signup", data);
    setState({ ...data });
  };

  const handleForgotPassword = (data) => {
    console.log("Forgot Password", data);
    setState({ ...data });
  };

  const handleResetPassword = (data) => {
    console.log("Reset Password", data);
    setState({ ...data });
  };

  const handleGetUser = () => {
    console.log("Get User");
    setState({ ...initialState });
  };

  const handleUpdateUser = (data) => {
    console.log("Update User", data);
    setState({ ...data });
  };

  const handleDeleteUser = () => {
    console.log("Delete User");
    setState({ ...initialState });
  };

  const handleUploadImage = (image) => {
    const formData = new FormData();
    formData.append("telephone", phone ?? "");
    formData.append("city", city ?? "");
    formData.append("firstname", firstname ?? "");
    formData.append("lastname", lastname ?? "");
    formData.append("description", description ?? "");
    formData.append("diploma", diplome ?? "");
    formData.append("year_of_experience", expirrience ?? "");
    formData.append("profession", profession ?? "");

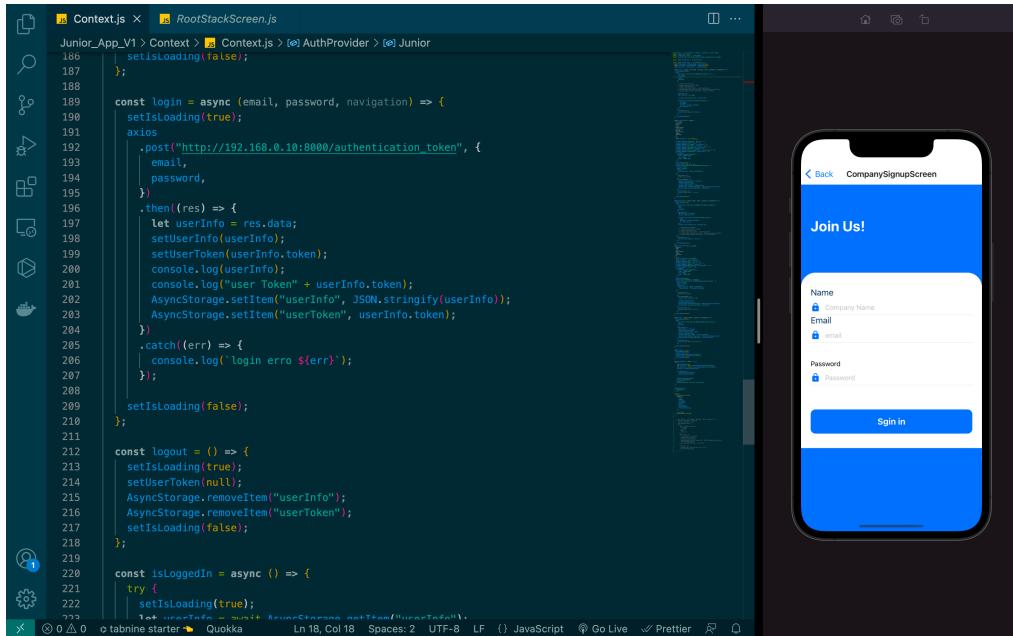
    if (image) {
      formData.append("photoFile", {
        type: "image/jpeg",
        uri: image,
        name: "image.jpg",
      });
    }

    setLoading(true);
    console.log("token.id", id);
    fetch(`http://192.168.0.10:8000/api/users/${id}`, {
      method: "POST",
      body: formData,
      headers: {
        authorization: `Bearer ${JwtToken}`,
      },
    })
      .then((res) => res.json())
      .then((data) => {
        console.log("Data", data);
        setState({ ...data });
      })
      .catch((err) => {
        console.error("Error", err);
      });
  };
};

export default useConext;
```

Ci-dessous la capture d'écran d'exécution de AsyncStorage pour stocker mes données et rafraîchir chaque itération avec l'aide de useEffect.

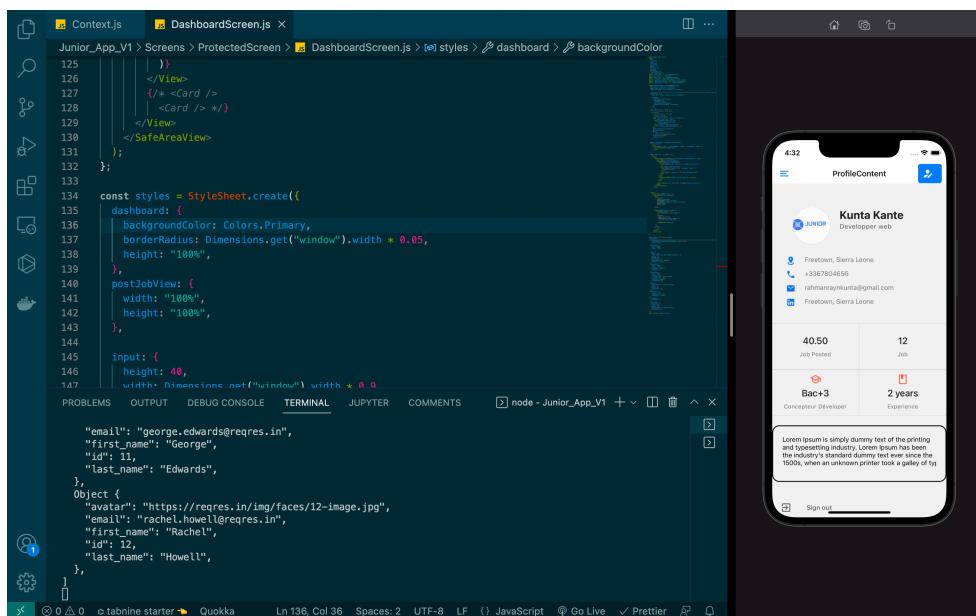
expo install @react-native-async-storage/async-storage



```
Context.js
186     setLoading(false);
187   };
188 
189   const login = async (email, password, navigation) => {
190     setLoading(true);
191     axios
192       .post("http://192.168.0.10:8000/authentication_token", {
193         email,
194         password,
195       })
196       .then((res) => {
197         let userInfo = res.data;
198         setUserInfo(userInfo);
199         setUserInfo(userInfo.token);
200         console.log(userInfo);
201         console.log("User Token" + userInfo.token);
202         AsyncStorage.setItem("userInfo", JSON.stringify(userInfo));
203         AsyncStorage.setItem("userToken", userInfo.token);
204       })
205       .catch((err) => {
206         console.log("Login error " + err);
207       });
208   };
209 
210   const logout = () => {
211     setLoading(true);
212     setUserInfo(null);
213     AsyncStorage.removeItem("userInfo");
214     AsyncStorage.removeItem("userToken");
215     setLoading(false);
216   };
217 
218   const isLoggedIn = async () => {
219     try {
220       setLoading(true);
221       let userInfo = await AsyncStorage.getItem("userInfo");
222     } catch (error) {
223       console.log(error);
224     }
225   };
226 }

RootStackScreen.js
125     </View>
126     <Card />
127     <Card />
128   </View>
129 </SafeAreaView>
130 </View>
131 </SafeAreaProvider>
132 </View>
133 </SafeAreaProvider>
134 const styles = StyleSheet.create({
135   dashboard: {
136     backgroundColor: Colors.Primary,
137     borderRadius: Dimensions.get("window").width * 0.05,
138     height: "100%",
139   },
140   postJobView: {
141     width: "100%",
142     height: "100%",
143   },
144   input: {
145     height: 40,
146     width: Dimensions.get("window").width * 0.8,
147   }
148 });

DashboardScreen.js
125     <Image alt="Profile picture of Kunta Kante" />
126     <Text>Kunta Kante</Text>
127     <Text>Developer web</Text>
128     <Text>Freetown, Sierra Leone</Text>
129     <Text>+3867804656</Text>
130     <Text>rahmanakunta@gmail.com</Text>
131     <Text>Freetown, Sierra Leone</Text>
132     <Text>40.50</Text>
133     <Text>12</Text>
134     <Text>Job Posted</Text>
135     <Text>Bac+3</Text>
136     <Text>2 years</Text>
137     <Text>Experience</Text>
138     <Text>Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and mixed it up to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.</Text>
139     <Text>Sign out</Text>
```



```
DashboardScreen.js
125     <Image alt="Profile picture of Kunta Kante" />
126     <Text>Kunta Kante</Text>
127     <Text>Developer web</Text>
128     <Text>Freetown, Sierra Leone</Text>
129     <Text>+3867804656</Text>
130     <Text>rahmanakunta@gmail.com</Text>
131     <Text>Freetown, Sierra Leone</Text>
132     <Text>40.50</Text>
133     <Text>12</Text>
134     <Text>Job Posted</Text>
135     <Text>Bac+3</Text>
136     <Text>2 years</Text>
137     <Text>Experience</Text>
138     <Text>Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and mixed it up to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.</Text>
139     <Text>Sign out</Text>
```

Review



A screenshot of a mobile application interface. The screen shows a list of profiles. Each profile card contains a small circular user icon, a name (e.g., Michael, Lindsay), and a large rectangular image placeholder. At the bottom of the screen, there is a navigation bar with icons for home, search, notifications, and account.

The left side of the image shows the code editor interface of a development environment. It displays two files: `Context.js` and `DashboardScreen.js`. The `DashboardScreen.js` file is open, showing code related to styling and component structure. The code includes imports for `Colors.Primary`, `Dimensions`, and `StyleSheet.create`. It defines styles for a `dashboard` section, which includes a background color, border radius, width, height, and a `postJobView` component. It also defines styles for an `input` element with a height of 40 pixels and a width of 90% of the window width.



A screenshot of a mobile application interface showing a profile creation screen. The screen has a title "CreateProfileJunior" at the top. Below the title, there is a large image placeholder. The form consists of several input fields: a text field for "John Doe", a text field for "0987656", a text field for "Marseille", a dropdown menu for "Kantu", and another dropdown menu for "exams". At the bottom of the form, there is a "Create" button. A message at the bottom right says "Let's all try and have some unity between ourself".

The left side of the image shows the code editor interface of a development environment. It displays two files: `Context.js` and `RootStackScreen.js`. The `RootStackScreen.js` file is open, showing code related to fetching data from a server. The code includes imports for `Image` and `FormData`. It defines a function that sets a loading state to true, logs the form data, and then performs a fetch request to `http://10.0.5.191:8000/api/entreprises/${id}` using a POST method with the form data as the body. The headers include an authorization token and a content type of `multipart/form-data`.

Réalisation

MVC

L'environnement logiciel utilisé et le design pattern MVC (Modèle Vue Controller). Ce pattern décrit une manière d'architecturer et d'organiser son code. C'est une méthode appliquant le principe de séparation des responsabilités, notamment le traitement de l'information et de sa mise en forme, pour guider l'équipe de développement sur la fonction associé à la création des fichiers. Il permet de séparer la logique du code en trois parties :

Modèle : Partie qui gère la manipulation de données du site. Son rôle est de récupérer

- ✓ les informations brutes dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le controller. Il permet de faire les requêtes SQL à la base de données

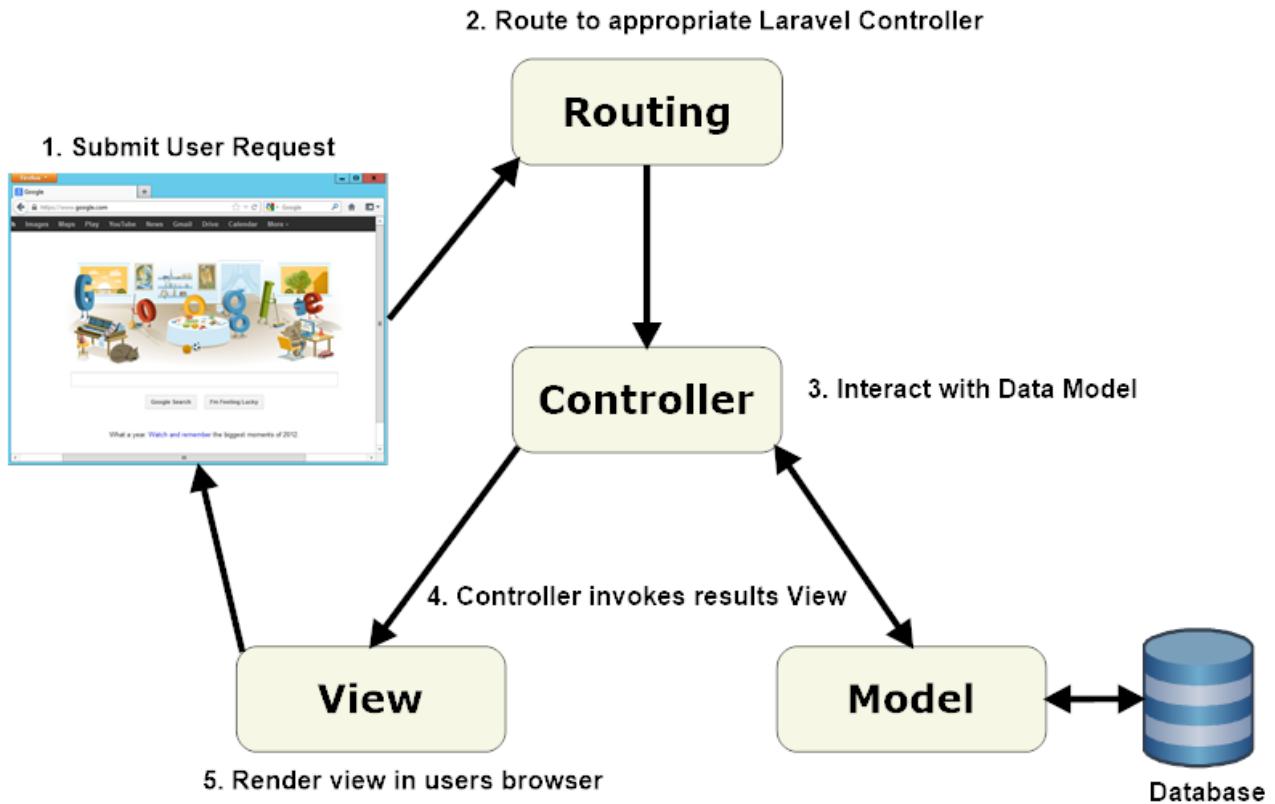
Vue : Cette partie se concentre sur l'affichage. Elle récupère les

- ✓ variables pour savoir ce qu'elle doit afficher. On y retrouver essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simple, pour afficher par exemple une liste de message.

Contrôleur: Cette partie gère la logique du code qui prend les

- ✓ décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le controller va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le controller contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

Ci-dessous le schéma d'un modèle MVC



MCD et MLD

Le MCD (Modèle Conceptuel de Données) ne peut pas être implanté dans une base de données sans modification.

Il est obligatoire de transformer ce modèle. On dit qu'on effectue un passage du modèle conceptuel de données vers le modèle logique de données. MLD

Le MLD pourra être implanté dans une base de données relationnelle.

Security(react-native)

La sécurité est souvent négligée lors de la création d'applications. Il est vrai qu'il est impossible de créer un logiciel complètement impénétrable - nous n'avons pas encore inventé un verrou complètement impénétrable (les coffres-forts des banques, après tout, sont toujours cambriolés). Cependant, la probabilité d'être victime d'une attaque malveillante ou d'être exposé à une faille de sécurité est inversement proportionnelle à l'effort que vous êtes prêt à déployer pour protéger votre application contre une telle éventualité. Bien qu'un cadenas ordinaire soit crochetable, il est toujours beaucoup plus difficile à franchir qu'un crochet d'armoire !

Storing Sensitive Info

Ne stockez jamais de clés API sensibles dans le code de votre application. Tout ce qui est inclus dans votre code peut être consulté en texte brut par toute personne inspectant l'ensemble d'applications. Des outils tels que react-native-dotenv et react-native-config sont parfaits pour ajouter des variables spécifiques à l'environnement telles que les points de terminaison d'API, mais ils ne doivent pas être confondus avec les variables d'environnement côté serveur, qui peuvent souvent contenir des secrets et des clés d'API.

Async Storage

Async Storage est un module géré par la communauté pour React Native qui fournit un magasin clé-valeur asynchrone et non chiffré. Le stockage asynchrone n'est pas partagé entre les applications : chaque application possède son propre environnement sandbox et n'a pas accès aux données des autres applications.

Network Security

Vos API doivent toujours utiliser le cryptage SSL. Le cryptage SSL protège contre la lecture des données demandées en texte brut entre le moment où elles quittent le serveur et avant qu'elles n'atteignent le client. Vous saurez que le point de terminaison est sécurisé, car il commence par https:// au lieu de http://.

Security(Symfony-Api-platefom)

composer require symfony/security-bundle

The User(L'utilisateur)

Toute section sécurisée de votre application nécessite un certain concept d'utilisateur. Le fournisseur d'utilisateurs charge les utilisateurs à partir de n'importe quel stockage (par exemple, la base de données) sur la base d'un "identifiant d'utilisateur" (par exemple, l'adresse e-mail de l'utilisateur) ;

Les autorisations dans Symfony sont toujours liées à un objet utilisateur. Si vous avez besoin de sécuriser (des parties de) votre application, vous devez créer une classe d'utilisateurs. Il s'agit d'une classe qui implémente `UserInterface`. Il s'agit souvent d'une entité Doctrine, mais vous pouvez également utiliser une classe d'utilisateurs dédiée à la sécurité.

Registering the User: Hashing Password

De nombreuses applications nécessitent qu'un utilisateur se connecte avec un mot de passe. Pour ces applications, le `SecurityBundle` fournit des fonctionnalités de hachage et de vérification de mot de passe

The Firewall & Authenticating Users (firewalls)

Le pare-feu est au cœur de la sécurisation de votre application. Chaque demande dans le pare-feu est vérifiée si elle nécessite un utilisateur authentifié. Le pare-feu se charge également d'authentifier cet utilisateur (par exemple à l'aide d'un formulaire de connexion) ;

La section pare-feu de `config/packages/security.yaml` est la section la plus importante. Un "pare-feu" est votre système d'authentification : le pare-feu définit quelles parties de votre application sont sécurisées et comment vos utilisateurs pourront s'authentifier (par exemple, formulaire de connexion, jeton API, etc.).

Access Control (Authorization) (access_control)

À l'aide du contrôle d'accès et du vérificateur d'autorisation, vous contrôlez les autorisations requises pour effectuer une action spécifique ou visiter une URL spécifique.

RGPD

Les entreprises de l’Union européenne doivent respecter le RGPD à l’égard de toutes les personnes concernées, quelle que soit leur localisation géographique, tandis que les entreprises en dehors de l’Union européenne doivent respecter le RGPD à l’égard des personnes concernées vivant au sein de l’Union européenne.

Qui est responsable de la collecte de consentements valides ?

Le propriétaire du site Internet ou le délégué à la protection des données (DPD) de l’entreprise est tenu de se conformer aux réglementations du RGPD relatives au traitement des données obtenues par le biais de cookies. Même si les cookies ne sont pas détenus par l’entreprise et sont des cookies tiers. Par exemple : les cookies de Google Analytics, Facebook Pixel, YouTube ou Addthis.

Quelle est votre tâche ?

Avant tout, vous devez disposer d’une bannière contextuelle de cookies valide sur votre site Internet, qui :

1°- Informe les visiteurs de l’utilisation de cookies par le site Internet

2°- Collecte les consentements des visiteurs (et les stocke pendant 5 ans, comme l’exige la loi)

3°- Donne aux visiteurs la possibilité de refuser les cookies (refus du suivi)

Merci

