

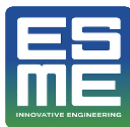
Projet Javascript

Site Web

Binôme: Cindy MUTHUKRISHNAN
Ishika HOSSAIN

Classe: A2MSI

Promotion: 2021-2022



RECIP'EASY

19/04/2022

Professeure référente: Zouhour GUIRAS

TABLE DES MATIERES

Introduction	3
I. Description du projet	4
II. Installation de notre site	5
III. Exigences fonctionnelles.....	8
A. Liste des fonctionnalités	8
B. Description des pages.....	8
Page d'Accueil	9
Barre de menu responsive	10
Page d'Inscription	10
Page de Connexion.....	11
Page de Recettes filtrées par pays	11
Page de cours de cuisine & Page de formulaire	13
Page de Contact.....	14
Page FAQ.....	15
IV. Exigences techniques	16
A. Réalisation du back-end du site.....	16
B. Fonctions JavaScript.....	17
i. Fonction inscription()	17
ii. Fonction lesson_register()	19
iii. Fonction sendMail()	19
iv. Fonction toggleMenu().....	21
v. Fonction getMealList()	22
V. La Méthode Agile	23
Conclusion	25

Introduction

« Vous rentrez chez vous après une longue journée de travail, et n'avez aucune idée de ce que vous pourrez cuisiner pour ce soir. L'heure tourne, et c'est la panique totale, vous ne savez ni quoi faire, ni quoi manger. C'est alors que vous vous souvenez qu'un super site existe avec des recettes simples et délicieuses, vous permettant de voyager à travers le monde durant un laps de temps.

Avec Recip'Easy, fini les tracas et place à la tranquillité : chercher des recettes du monde entier en un seul clic, et découvrez diverses recettes pour éviter la redondance des plats ! Vous pouvez même prendre des cours de cuisine pour augmenter vos compétences ! Simple, rapide et efficace, tels sont les trois mots qui décrivent la plate-forme de cuisine développée par deux étudiantes d'ESME Sudria. »

Dans le cadre de notre enseignement de JavaScript, nous avons eu à développer un site web en binôme. Afin d'unifier notre équipe autour de ce projet, nous avons choisi un thème qui nous plaisait à toutes les deux, et nous avons donc eu l'idée de créer un site de recettes de cuisine.

Ensemble, nous avons réfléchi à ce que notre site web pourrait apporter de plus par rapport à des sites de recettes classiques. Nous avons donc pris la décision de concevoir une plateforme ayant la particularité suivante : proposer des recettes variées du monde entier. Aussi nous souhaitions que nos internautes puissent s'inscrire directement sur notre plateforme à des cours de cuisine par niveau afin qu'ils puissent développer leurs compétences et apprendre de nouvelles techniques. C'est ainsi que le projet « Recip'Easy » est apparu.

Ce rapport a pour but d'expliquer comment ce projet a été réalisé dans sa globalité, avec une description, les étapes d'installations de notre site et les différentes exigences fonctionnelles et techniques.

I. Description du projet

Notre projet consiste à créer un site internet recensant des recettes de cuisine du monde entier. Nommé Recip'Easy, notre site permettra à nos clients de rechercher des recettes par catégorie de cuisine du monde (canadienne, italienne, chinoise, etc.).

Aussi, via notre plateforme, nos internautes auront la possibilité de s'inscrire en fonction de leur niveau à des cours de cuisines dispensés par plusieurs chefs. Les cours de cuisine sont répartis en 3 catégories : débutant, intermédiaire et avancé.

La réalisation du site web se décompose en deux grandes parties avec : une partie « cliente » (le visuel) où nous utilisons les langages html, css, JavaScript, et d'autre part, une partie serveur (le « backend ») où nous utilisons le serveur Node JS.

Notre groupe de projet se compose de deux étudiantes issues de l'école d'ingénieurs généralistes *ESME Sudria*, ayant choisi le Management des Systèmes d'Information comme majeure :

- MUTHUKRISHNAN Cindy
- HOSSAIN Ishika

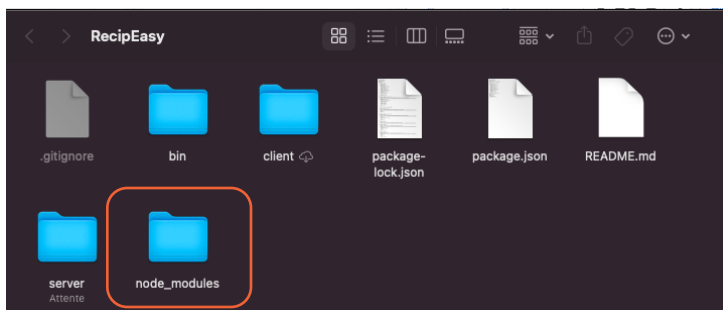
Nous avons choisi la méthodologie agile afin de gérer le projet dans de bonnes conditions. La méthode agile a permis à toute l'équipe d'avoir un visuel sur l'avancement du projet. Ainsi, toute l'équipe participe à l'ensemble des tâches à mettre en place, que ce soit sur le graphisme, ou encore, le code. Cela nous permettait alors d'acquérir et de développer des connaissances sur le développement en HTML, CSS et JavaScript mais aussi sur l'organisation d'un projet.

II. Installation de notre site

Pour accéder à notre site web, il vous faut dans un premier temps installer **PostgreSQL** afin d'importer notre base de données. Ensuite, il faut installer sur votre ordinateur **Node.js** qui sera notre serveur WEB.

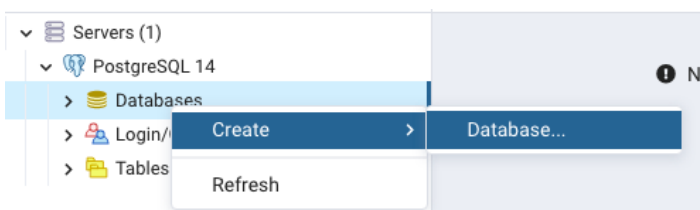
Après ces installations préalables, voici les étapes à suivre :

- 1) Télécharger l'archive c'est-à-dire le point ZIP qui contient notre projet WEB.
- 2) Extrayez l'archive dans un dossier quelconque ou sur votre bureau de votre ordinateur.
- 3) Ouvrir PowerShell pour Windows ou un Terminal pour linux ou mac.
- 4) Grâce au terminal, vous devez naviguer jusqu'au dossier créé.
- 5) Ensuite tapez la commande suivante : **npm install**
- 6) Après avoir réalisé ces étapes, un nouveau dossier node_modules apparaît dans notre dossier projet. Ce dossier contient les modules externes à notre code.

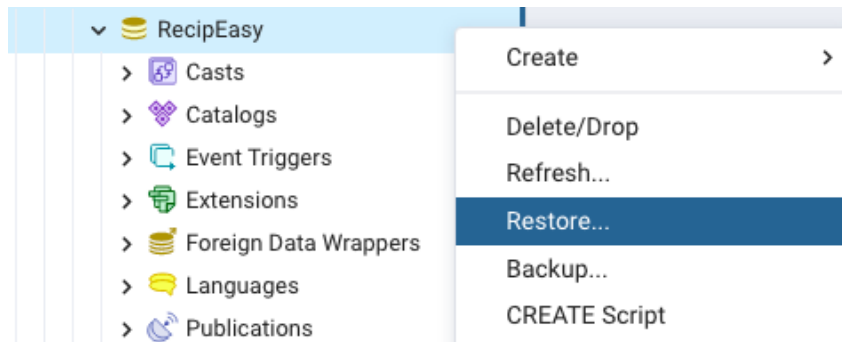


7) Importer la base de données sur PostgreSQL :

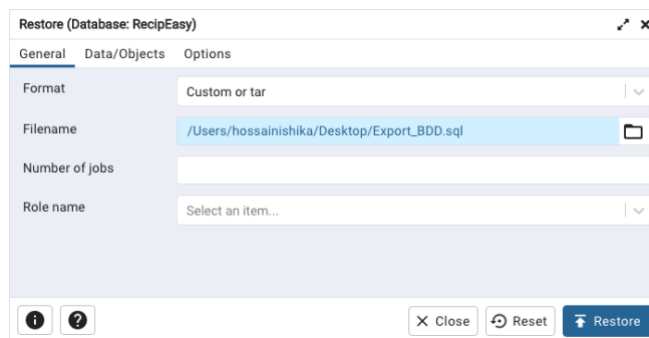
- Ouvrir pgAdmin et se connecter ;
- Créer une base de données vierge avec comme nom « **RecipEasy** » ;



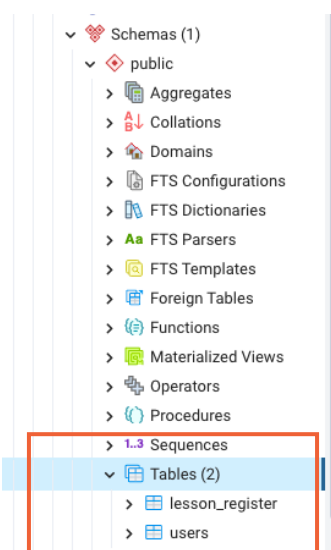
- Faire un clic droit sur la base de données et choisir « Restore »



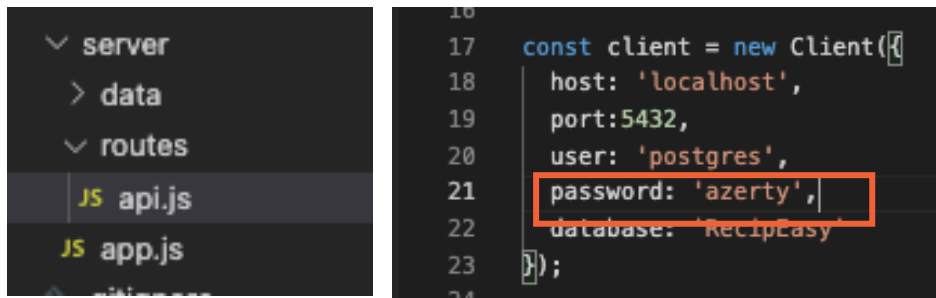
- À partir de l'onglet général, dans le champ « Filename », charger le fichier « **Export_BDD.sql** » (envoyé par email) qui contient tous les tableaux et données de notre base de données. Et cliquez sur Restore.



- Vous devriez maintenant voir les deux tables suivantes : « **users** » et « **lesson_register** » dans **RecipEasy > Schemas > Tables**.



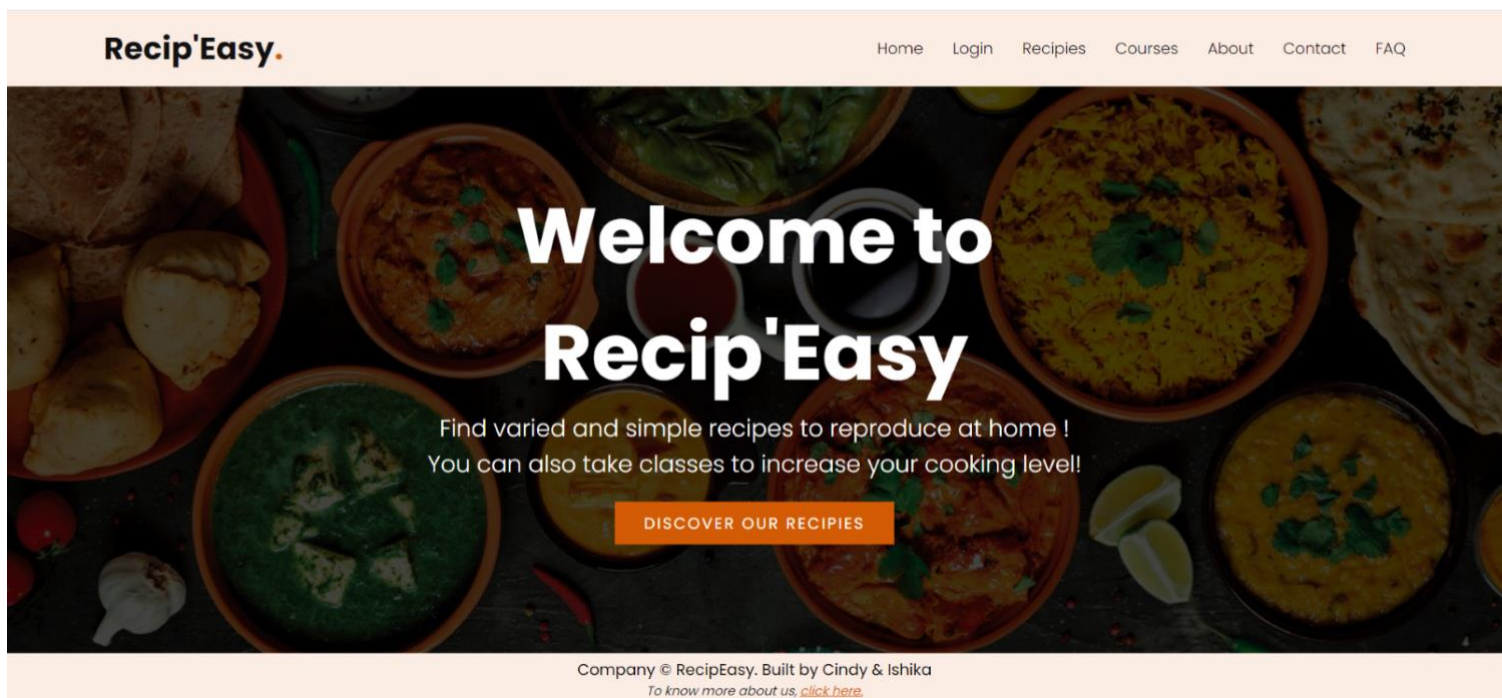
- Changez le mot de passe par le vôtre dans le fichier « **api.js** » de notre projet dans **server > routes > api.js**.



8) Maintenant, pour lancer notre serveur, retournez sur votre terminal à l'endroit du dossier et taper **npm start**.

9) Ensuite entrer l'URL suivante : <http://localhost:3000> sur votre navigateur web.

Si tout s'est bien passé, vous devriez tomber sur la page suivante :



III. Exigences fonctionnelles

A. Liste des fonctionnalités

L'idée principale était de créer un site simple d'utilisation permettant aux internautes de :

- S'inscrire sur le site en fournissant un nom d'utilisateur, un email et un mot de passe.
- Se connecter sur le site.
- Une fois connecté, visualiser la page de recherche de recette pour suivre des recettes de cuisine par rapport au pays.
- Cliquer sur une recette, voir les instructions et la vidéo YouTube associée.
- S'inscrire à des cours de cuisine en fonction de leur niveau. Ils peuvent également choisir la date et le créneau d'horaire souhaité en fonction des disponibilités.
- Contacter notre équipe en cas de besoin via le formulaire de contact.
- Prendre connaissance des questions les plus posées via la page FAQ.

B. Description des pages

La conception de l'interface était l'un des éléments primordiaux. Nous voulions que l'expérience de l'utilisateur soit la plus ergonomique possible. Pour cela, une barre de menu est placée en haut de chacune des pages afin de faciliter les accès aux différentes fonctionnalités du site.

Chaque page possède également un bas de page permettant d'accéder à la page « À propos ».

Cette partie visuelle a été conçu au moyen de Vue.js. Le Vue.js est un Framework léger et rapide. Il permet de détecter plus rapidement des erreurs et donc nous permet d'optimiser au maximum notre site. En effet, VueJS utilise une approche orientée composants (des fichiers. Vue). Chaque composant est un bloc autonome

avec sa propre template, sa propre logique et son propre style. Les composants sont découpés en trois parties :

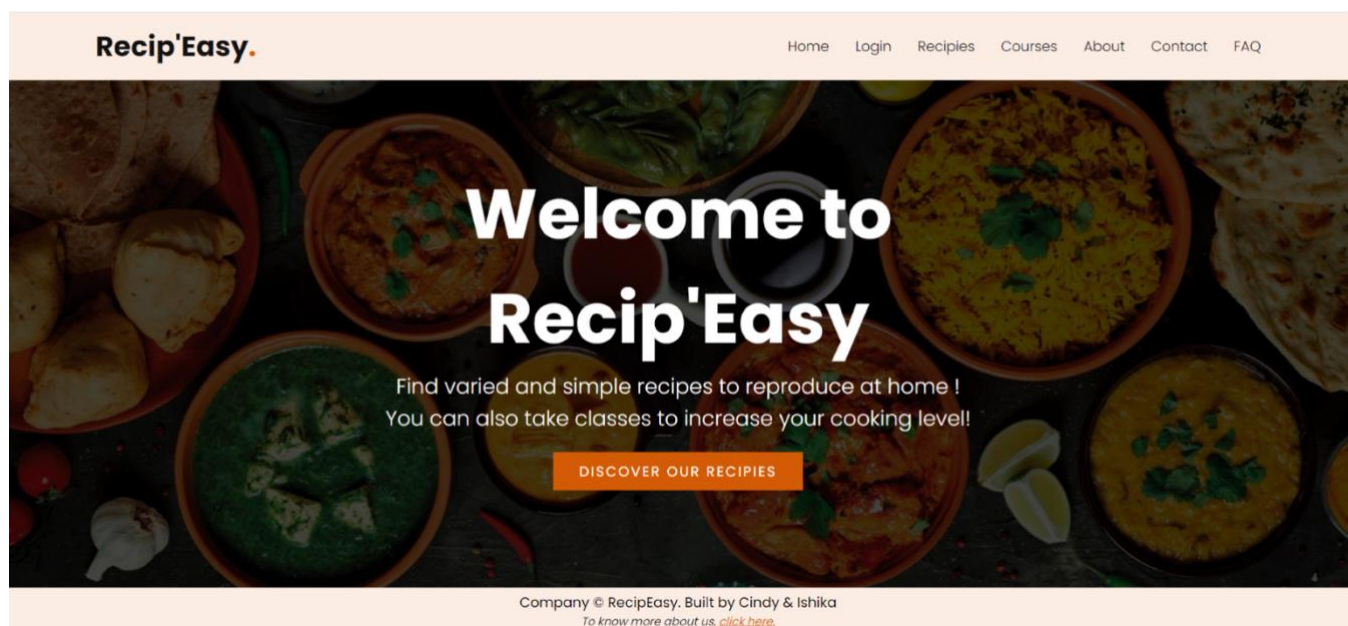
- **Le template** : C'est une balise dans laquelle on implémente la structure de notre page grâce au HTML.
- **Le script** : Une balise dans lequel on va implémenter des codes JavaScript, qui permet d'avoir des interactions avec l'utilisateur.
- **Le style scoped** : Cette balise permet d'apporter au composant une certaine esthétique grâce au CSS.

Page d'Accueil

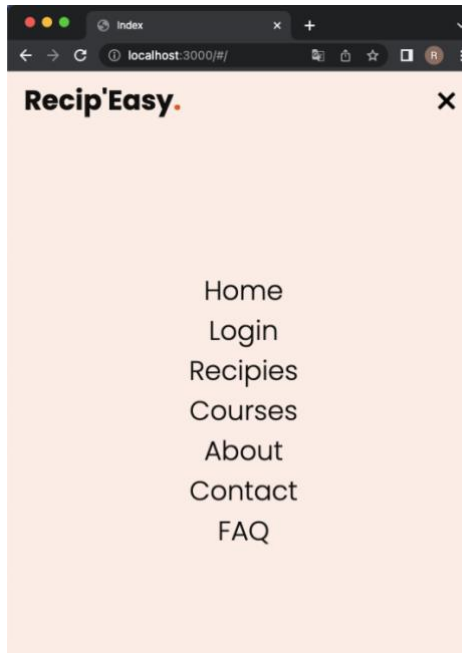
La page d'accueil présente de manière synthétique le principe de notre site. Elle permet d'avoir une idée générale sur le site et de naviguer vers les autres pages du site avec la barre de navigation.

Si l'utilisateur est connecté, il pourra naviguer entre toutes les pages sans problème. Si ce n'est pas le cas et qu'il appuie sur « **Recipies** » ou « **Courses** », il sera directement redirigé vers la page de connexion puisque ces deux pages requièrent la connexion de l'utilisateur.

Pareillement pour le bouton « **Discover our recipies** », si l'utilisateur est connecté et qu'il appuie sur ce bouton, il sera redirigé vers la page de recettes, sinon, vers la page de connexion.



Barre de menu responsive



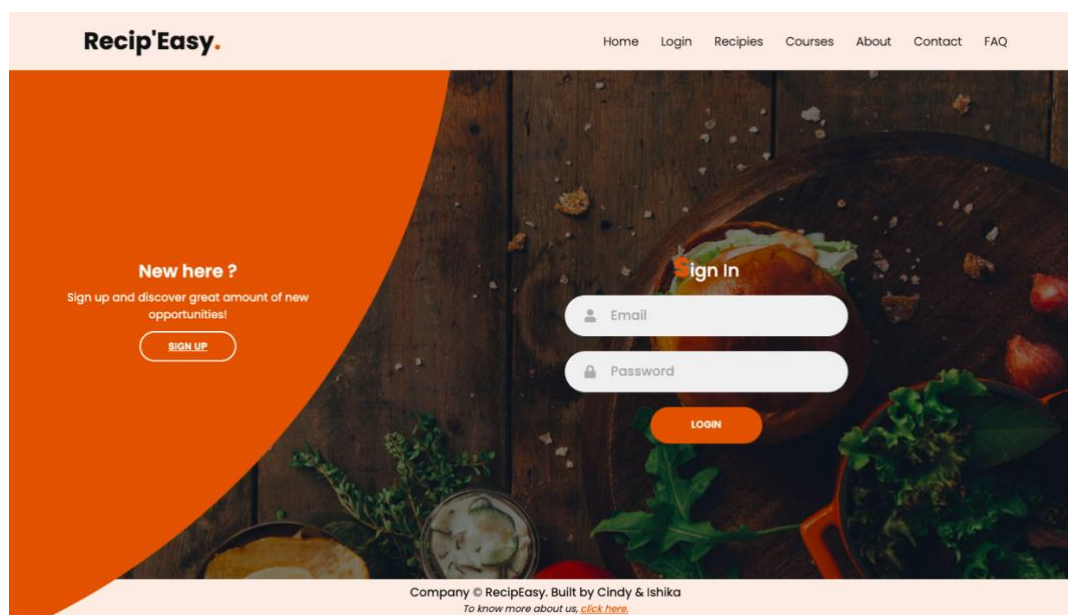
Page d'Inscription

Toute personne souhaitant accéder aux recettes de notre site devra s'inscrire. Pour effectuer son inscription plusieurs champs sont à remplir : nom d'utilisateur, email, mot de passe. Si l'utilisateur appuie sur « SIGN IN », il sera redirigé vers la page de connexion.

A screenshot of the Recip'Easy sign-up page. The page has a dark background with a wooden surface and food. On the left, there is a sign-in section with the text 'One of us ?' and 'Time to feel like home... Connect quickly!' with a 'SIGN IN' button. On the right, there is a sign-up section with the text 'Sign Up' and four input fields: Username, Email, Password, and Confirmation password, followed by a 'SIGN UP' button. The footer contains the text 'Company © RecipEasy. Built by Cindy & Ishika' and a link 'To know more about us, click here.'

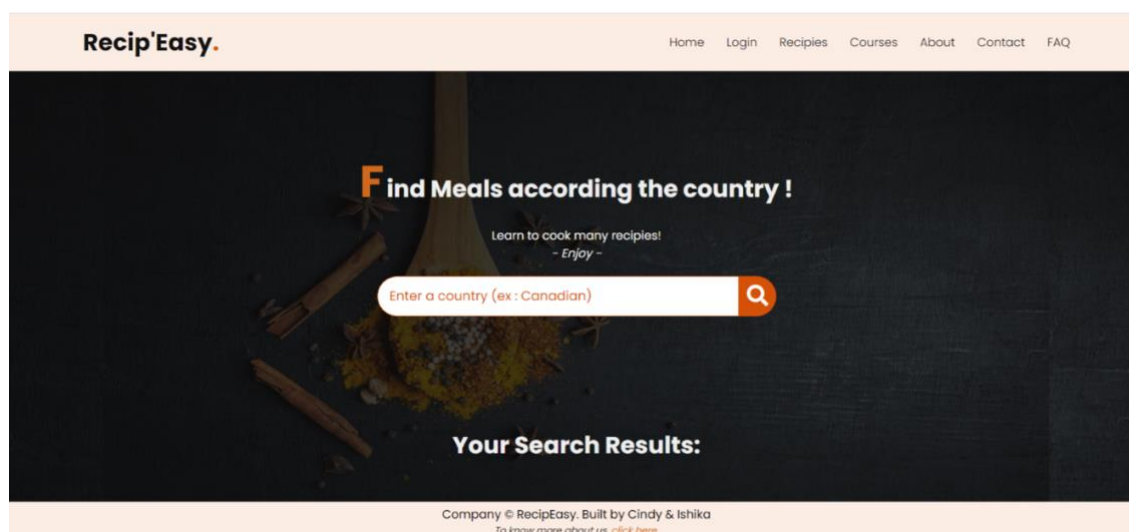
Page de Connexion

Une fois que l'internaute s'est inscrit sur notre site, il pourra se connecter via la page de connexion en inscrivant les champs d'email et de mot de passe. Si l'utilisateur appuie sur « SIGN UP », il sera redirigé vers la page d'inscription.



Page de Recettes filtrées par pays

Cette page permet d'effectuer une recherche de recette par type de cuisine du monde. L'internaute pourra donc lancer la recherche de son choix et découvrir les recettes proposées par notre site. En cliquant sur une recette, il aura la liste des instructions à suivre avec également le lien vers la vidéo de cuisine de la recette en question.






Recip'Easy.

[Home](#)[Login](#)[Recipes](#)[Courses](#)[About](#)[Contact](#)[FAQ](#)

french


Your Search Results:




Company © RecipEasy. Built by Cindy & Ishika
To know more about us, [click here](#).

Recip'Easy.


[Home](#)[Login](#)[Recipes](#)[Courses](#)[About](#)[Contact](#)[FAQ](#)



Coq au vin
Get Recipe



Duck Confit
Get Recipe



Fennel Dauphinoise
Get Recipe

Company © RecipEasy. Built by Cindy & Ishika
To know more about us, [click here](#).

Recip'Easy.

[Home](#)[Login](#)[Recipes](#)[Courses](#)[About](#)[Contact](#)[FAQ](#)

Coq au vin


Duck Confit

Fennel Dauphinoise

Fennel Dauphinoise

Side

Instructions:
Heat oven to 180C/160C fan/gas 4. Put potatoes, fennel, and garlic in a medium non-stick pan. Pour in milk and double cream, season well and simmer gently, covered, for 10 mins, stirring halfway through, until potatoes are just tender. Divide the mixture between 2 small (about 150ml) buttered ramekins and scatter with Parmesan. Bake for 40 mins until the potatoes are golden and tender when pierced with a knife. Snip the reserved fennel fronds over before serving.



Watch Video

Company © RecipEasy. Built by Cindy & Ishika
To know more about us, [click here](#).

Page de cours de cuisine & Page de formulaire

Sur cette page on retrouve les 3 types de cours de cuisine proposés sur notre plateforme. On y trouve une description détaillée de chaque cours pour que la personne puisse choisir le cours le plus adapté à son niveau.

On peut aussi accéder à un formulaire qui permettra à l'internaute d'effectuer son inscription. Il aura des champs spécifiques à remplir et fonction du niveau choisi, il aura des créneaux d'horaires et des chefs spécifiques de proposés. Les menus déroulant des horaires et des chefs s'adaptent par rapport au niveau choisi.

The screenshot shows the Recip'Easy website with a navigation bar (Home, Login, Recipies, Courses, About, Contact, FAQ) and a main banner with the text "Take Cooking sessions with Recip'Easy ! No matter your level, Recip'Easy is here to support you. So choose classes according your level !". Below the banner are three course cards:

- Beginner:** You have no cooking skills? No problem, Paula and Chris are here to teach you the basics !
 - 1 Hour
 - 10h-11h / 11h-12h
 - Completely Free !
 - Chiefs : Paula, Chris[CLICK HERE !](#)
- Intermediate:** You have the basics in cooking, but want to learn more? Take an intermediate class and watch your skills soar!
 - 1 Hour
 - 14h-15h / 15h-16h
 - Completely Free !
 - Chiefs : Emelia, Marcus[CLICK HERE !](#)
- Expert:** You already have a very good knowledge of cooking but would like to learn from our chefs? Take classes and become an expert !
 - 1 Hour
 - 16h-17h / 17h-18h
 - Completely Free !
 - Chiefs : Marie, Lucas[CLICK HERE !](#)

Company © RecipEasy. Built by Cindy & Ishika
To know more about us, [click here](#).

En cliquant sur le bouton « **Click Here !** », on arrive directement sur le formulaire suivant :

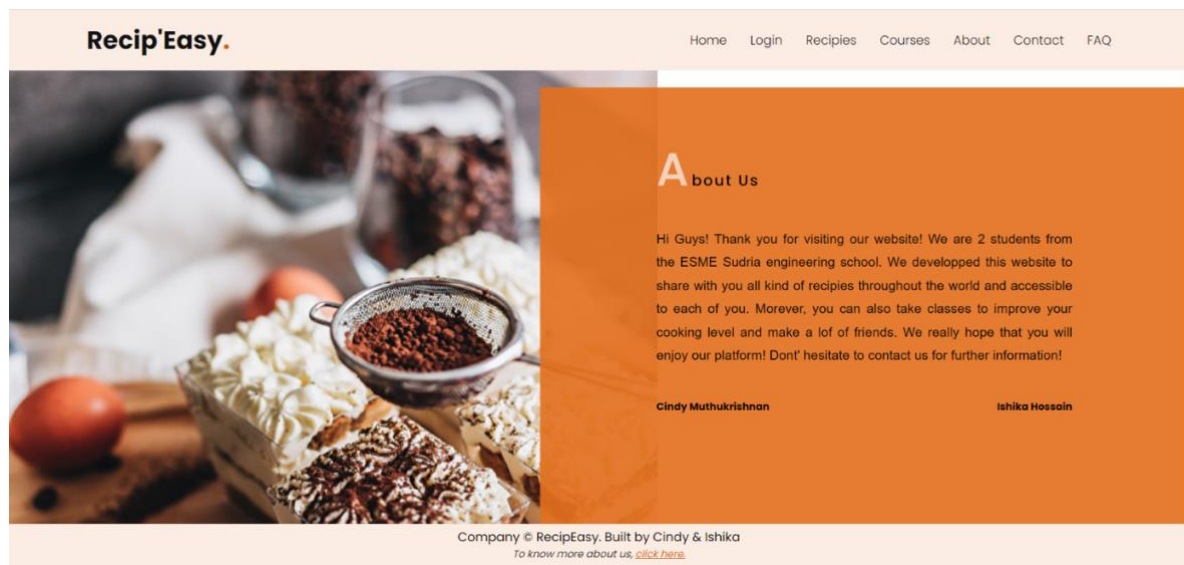
The screenshot shows the Recip'Easy website with a navigation bar (Home, Login, Recipies, Courses, About, Contact, FAQ) and a main banner with the text "Take Cooking sessions with Recip'Easy ! No matter your level, Recip'Easy is here to support you. So choose classes according your level !". Below the banner are three course cards:

- Beginner:** You have no cooking skills? No problem, Paula and Chris are here to teach you the basics !
 - 1 Hour
 - 10h-11h / 11h-12h
 - Completely Free !
 - Chiefs : Paula, Chris[CLICK HERE !](#)
- Intermediate:** You have the basics in cooking, but want to learn more? Take an intermediate class and watch your skills soar!
 - 1 Hour
 - 14h-15h / 15h-16h
 - Completely Free !
 - Chiefs : Emelia, Marcus[CLICK HERE !](#)
- Expert:** You already have a very good knowledge of cooking but would like to learn from our chefs? Take classes and become an expert !
 - 1 Hour
 - 16h-17h / 17h-18h
 - Completely Free !
 - Chiefs : Marie, Lucas[CLICK HERE !](#)

Company © RecipEasy. Built by Cindy & Ishika
To know more about us, [click here](#).

Page « À propos »

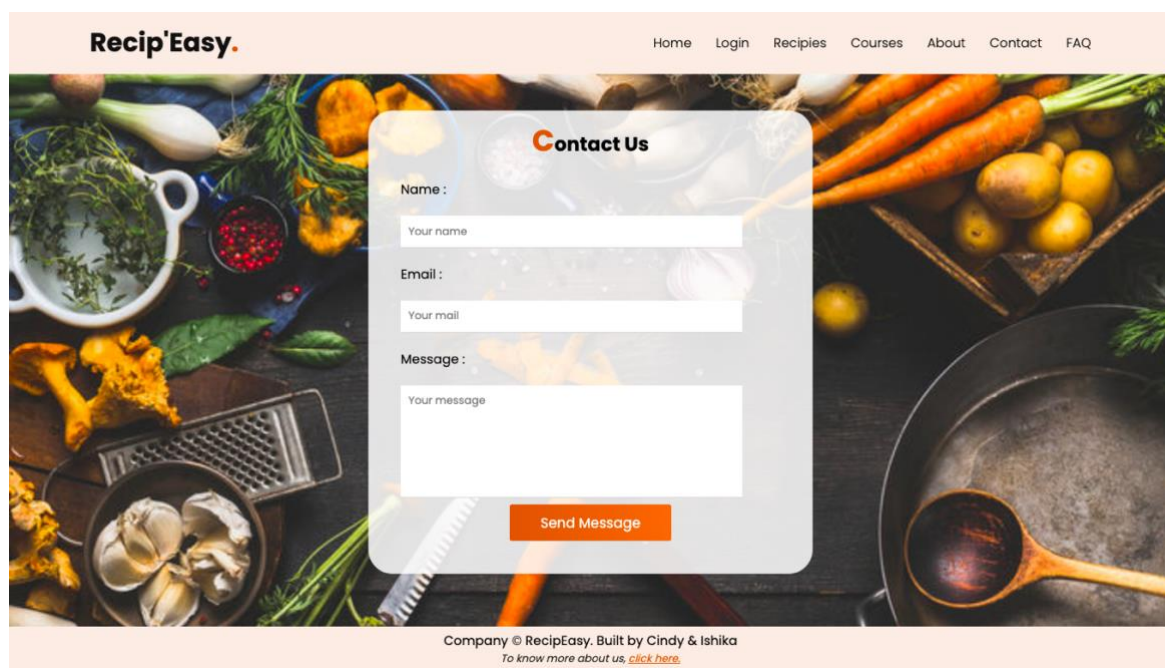
Cette page comporte une présentation générale du principe de notre site. Les internautes peuvent ainsi mieux comprendre qui nous sommes.



Page de Contact

Cette page permet à n'importe quelle personne de nous contacter directement en cas de besoin. Ils doivent remplir trois champs : nom, email et le contenu du message.

Grâce à l'utilisation d'EmailJS, nous recevons par la suite le message sur notre adresse mail créé pour le projet « recipeasy.project@gmail.com ».

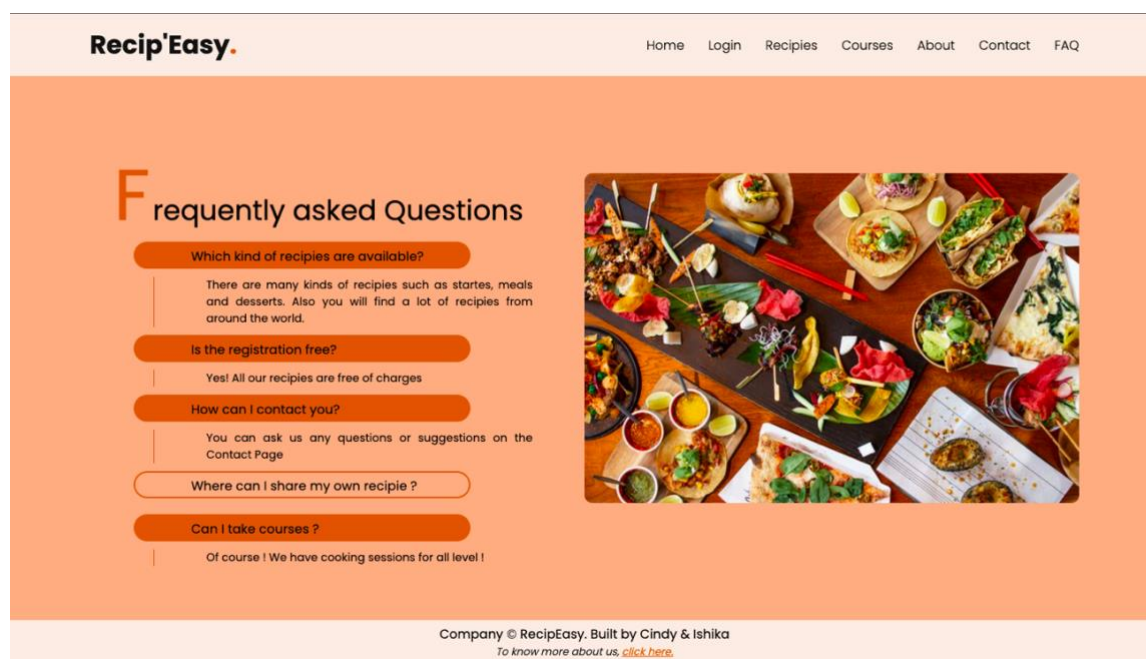


Voici un exemple de mail reçu :



Page FAQ

Cette page recense les questions les plus posées par nos internautes. Ils peuvent ainsi consulter et prendre connaissance des réponses que nous avons apporté. Comme vous pouvez le voir, il suffit de cliquer sur une question pour avoir la réponse qui s'affiche juste en dessous dynamiquement.



IV. Exigences techniques

A. Réalisation du back-end du site

Pour ce projet, nous avons développé un serveur Web avec une API (Application Programming Interface) qui communique avec une base de données de type SQL. Pour cela, nous avons codé la partie serveur avec Node.js, la partie client en Vue.js et la base de données que nous avons utilisé est PostgreSQL.

Le serveur permet de faire le lien entre la partie cliente et la partie serveur. Le client fait donc une requête au côté serveur pour effectuer une action sur les ressources. Les objectifs du serveur sont donc de fournir les bonnes données aux bons utilisateurs au bon moment, mais également de stocker des informations et assurer la sécurité des données avec un système d'authentification.

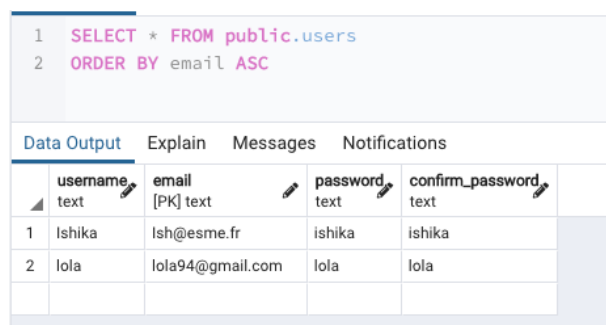
Pour avoir accès aux recettes du monde entier, nous avons utilisé une API externe (provenant du site <https://www.themealdb.com/api.php>) dans laquelle est stockée des milliers de recettes. Ainsi, lorsque l'utilisateur va effectuer une recherche de recettes par pays, le serveur va envoyer une requête à l'API externe et va récupérer les données associées à la recherche de l'utilisateur. Ensuite ces données vont être affichées sur l'écran de l'utilisateur.

En plus de faire des requêtes auprès d'une API externe le serveur doit interagir avec une base de données. Cette base de données permet de stocker les informations des utilisateurs. Elle est utilisée lors de l'inscription et lors de la connexion. Lorsqu'un utilisateur s'inscrit, ses données sont ajoutées à la base de données, et lorsqu'un utilisateur se connecte alors le serveur vérifie que son e-mail et son mot de passe correspondent aux informations de la base de données. Ainsi, le serveur a une importance majeure car il permet de faire le lien entre le client et la base de données.

B. Fonctions JavaScript

i. Fonction inscription()

Lorsqu'un utilisateur s'inscrit sur notre site, ses données sont ajoutées à la base de données et plus spécifiquement, dans la table « users ». C'est cette table qui stocke toutes les données des utilisateurs lors de leur inscription.



The screenshot shows a SQL query editor with the following query:

```
1 SELECT * FROM public.users
2 ORDER BY email ASC
```

Below the query, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing a table with the following data:

	username text	email [PK] text	password text	confirm_password text
1	Ishika	Ish@esme.fr	ishika	ishika
2	lola	lola94@gmail.com	lola	lola

Table « users » de la base de données

Pour ce faire, nous récupérons dans un premier temps les données remplies par l'utilisateur avec le fichier « SignUp.vue ».

```
<script>
module.exports = {
  ...
  data(){
    return{
      users:{
        username : '',
        email : '',
        password : '',
        confirm_password : '',
      }
    }
  },
  methods : {
    inscription(){
      this.$emit("inscription", this.users);
    }
  }
}
</script>
```

Ensuite, nous créons une requête « POST » qui permet justement de stocker les données rentrées par l'utilisateur dans cette table.

```

router.post('/signup', async (req,res) => {
  Username = req.body.username;
  Email = req.body.email;
  Mdp = req.body.password;
  Mdp_conf = req.body.confirm_password;

  if(Username == "" || Email== "" || Mdp== "" || Mdp_conf== ""){
    res.json("Bad request - please enter your information");
    console.log("Bad request - please enter your information");
  }

  let alreadyexists = false;
  const op = await client.query({
    text : "SELECT * FROM users"
  })

  let i = 0;
  while(i<op.rows.length && alreadyexists!=true){
    if(op.rows[i].email == Email){
      alreadyexists = true;
    }
    i++;
  }

  if(alreadyexists == false) {
    const op2 = await client.query({
      text : 'INSERT INTO users(username, email ,password, confirm_password) VALUES ($1,$2,$3,$4)',
      values : [Username, Email, Mdp, Mdp_conf]
    })
    res.json("Success");
  }
  else{
    res.json("This email adress already exists");
  }
})

```

Ainsi, on récupère les noms, email et mot de passe fournis par l'utilisateur, et on effectue les vérifications préalables : est-ce que tous les champs sont bien remplis ? Si ce n'est pas le cas, renvoyer une erreur.

Ensuite, on regarde le contenu de la table « users », et on vérifie si l'email donné par l'utilisateur est déjà présent dans la table. Si ce n'est pas le cas, alors l'inscription est bien possible, et on utilise requête « insert » en insérant les valeurs de chaque variable dans la base de données.

Et enfin, nous utilisons la fonction asynchrone « inscription() » pour que l'utilisateur soit redirigé vers la page de connexion une fois qu'il est inscrit (avec le routeur.push).

```

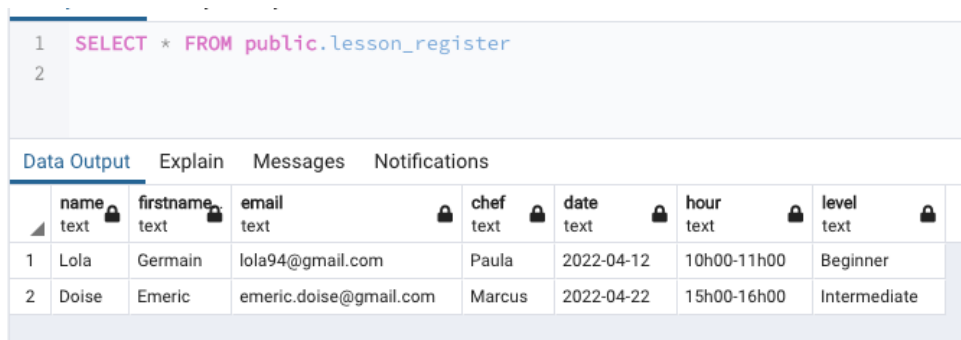
async inscription(users){
  const res = await axios.post('/api/signup', users);
  this.$router.push('/login')
},

```

ii. Fonction `lesson_register()`

Comme pour l'inscription, lorsqu'un utilisateur s'inscrit sur notre site afin de prendre un cours de cuisine, ses données sont ajoutées à la base de données et plus spécifiquement, dans la table « `lesson_register` ». C'est cette table qui stocke toutes les données des utilisateurs lors de leur inscription à un cours.

La technique est exactement identique avec la récupération des données, la requête « `POST` » et la fonction asynchrone avec une redirection vers la page de présentation des cours.



	name text	firstname text	email text	chef text	date text	hour text	level text
1	Lola	Germain	lola94@gmail.com	Paula	2022-04-12	10h00-11h00	Beginner
2	Doise	Emeric	emeric.doise@gmail.com	Marcus	2022-04-22	15h00-16h00	Intermediate

Table « `lesson_register` » de la base de données

iii. Fonction `sendMail()`

Pour que les utilisateurs puissent nous contacter à travers le formulaire de la page « `Contact` », nous avons créé la fonction « `sendMail(params)` » dans le fichier **`scriptcontact.js`** qui utilise `EmailJS`.

`EmailJS` est un logiciel de communication qui permet d'envoyer des e-mails directement depuis JavaScript. Nous avons ainsi créé un modèle d'e-mail que nous recevrons à chaque fois qu'un utilisateur essaiera de nous contacter avec un contenu dynamique (paramètres récupérés du formulaire). Ainsi, dès que l'utilisateur appuie sur le bouton d'envoi du formulaire, il y a un déclenchement d'e-mail.

Voici le template d'e-mail que nous avons conçu avec les paramètres dynamiquement spécifiés (prénom de l'utilisateur, adresse mail et message) :

My Default Template

[Playground](#)
[Test It](#)

[Content](#)
[Auto-Reply](#)
[Attachments](#)
[Contacts](#)
[Settings](#)

Subject *

New message from {{from_email}}

Content *

B I U sans-serif 12pt
 A

Hello RecipEasy,
 You got a new message from {{from_name}} ({{from_email}}) :
 {{message}}
 Best wishes,
 EmailJS team

To Email *

recipeasy.project@gmail.com

From Name

{{from_name}}

From Email *

☐ Use Default Email Address

 {{from_email}}

Reply To

Bcc

```
function sendMail(params){
  var tempParams = {
    from_name:document.getElementById("fromName").value,
    from_email:document.getElementById("fromEmail").value,
    message:document.getElementById("msg").value,
  };


  /* Send email with emailJS */
  emailjs.send('service_u1cv2rl','template_muvgl6e',tempParams,'GrkSeRXgiVa62PC1P')
    .then(function(res){
      console.log("success", res.status);
      location.reload();
    })
}
```

Ainsi comme vous pouvez le voir, on récupère dans un premier temps toutes les valeurs que l'utilisateur spécifie lorsqu'il remplit le formulaire grâce à leur Id (nom, email et message).

Ensuite, le déclenchement d'un e-mail s'effectue à l'aide de « emailjs.send » en spécifiant comme paramètre le numéro de notre service (Gmail), le numéro du template, les paramètres récupérées du formulaire (qu'on a appelé tempParams) et enfin la clé publique de l'API.

Email Services


[Add New Service](#)



Gmail
Service ID: service_u1cv2rl

Email Templates

[Create New Template](#)



My Default Template
Template ID: template_muvgl6e

API keys

Public Key

GrkSeRXgiVa62PC1P

iv. Fonction toggleMenu()

Pour que la barre de menu soit responsive et s'adapte lorsque la page se rétrécit, nous avons utilisé la fonction toggleMenu().

Ainsi, lorsque la page se réduit à une certaine taille, la fonction toggle permet de sélectionner la barre de navigation (avec querySelector) et basculer les noms de classe, pour que celles-ci soient désormais «.menuToggle.active» et «.navigation.active».

```
<!-- Fonction for responsive toolbar -->
<script type="text/javascript">
  function toggleMenu() {
    const menuToggle = document.querySelector('.menuToggle');
    const navigation = document.querySelector('.navigation');
    menuToggle.classList.toggle('active');
    navigation.classList.toggle('active');
  }
</script>
```

```
<div class="menuToggle" onclick="toggleMenu();"></div>

<!-- Toolbar access if user is connected -->
<ul class="navigation" v-if="connecte">
  <li><router-link to="/"><a href="#">Home</a></router-link></li>
  <li><router-link to="/login"><a href="#login">Login</a></router-link></li>
  <li><router-link to="/recipies"><a href="#recipies">Recipies</a></router-link></li>
  <li><router-link to="/lessons"><a href="#courses">Courses</a></router-link></li>
  <li><router-link to="/about"><a href="#about">About</a></router-link></li>
  <li><router-link to="/contact"><a href="#contact">Contact</a></router-link></li>
  <li><router-link to="/faq"><a href="#faq">FAQ</a></router-link></li>
</ul>

<!-- Toolbar access if user is not connected -->
<ul class="navigation" v-else>
  <li><router-link to="/"><a href="#">Home</a></router-link></li>
  <li><router-link to="/login"><a href="#login">Login</a></router-link></li>
  <li><router-link to="/login"><a href="#recipies">Recipies</a></router-link></li>
  <li><router-link to="/login"><a href="#courses">Courses</a></router-link></li>
  <li><router-link to="/about"><a href="#about">About</a></router-link></li>
  <li><router-link to="/contact"><a href="#contact">Contact</a></router-link></li>
  <li><router-link to="/faq"><a href="#faq">FAQ</a></router-link></li>
</ul>
</div>
```

Justement, dans le fichier css, nous avons justement défini ces classes avec une certaine représentation.



v. Fonction getMealList()

Afin d'afficher les recettes en fonction du type de recherche effectuée, nous avons utilisé la fonction `getMealList()`. Tout d'abord, nous récupérons le type de spécialité culinaire du monde entrée dans la barre de recherche par l'internaute via la fonction « `document.getElementById('search-input')` ».

Ensuite, nous utilisons la fonction `fetch` qui permet de récupérer les recettes de l'API que nous utilisons en fonction de la recherche effectuée sur notre site :

[https://www.themealdb.com/api/json/v1/1/filter.php?i=\\${searchInputTxt}](https://www.themealdb.com/api/json/v1/1/filter.php?i=${searchInputTxt})

Ainsi nous affichons :

- Le nom de la recette via le : « `${meal.strMeal}` ».
- Une caractéristique spécifique de la recette (exemple : un ingredient en particulier) via le : « `${meal.idMeal}` ».
- L'image associée à la recette grâce au : « `${meal.strMealThumb}` ».

Si la recherche effectuée par l'internaute n'est pas reconnue car non contenue dans notre API, alors un message sera affiché pour indiquer qu'aucune recette n'a pu être associée à la recherche. Chaque nouvelle recherche non reconnue, sera ajoutée à la liste des recherches non trouvées.

Nous avons pu effectuer ces fonctionnalités grâce aux fonctions :

“`mealList.classList.remove('notFound')`” et “`mealList.classList.add('notFound')`”.

```
// get meal list that matches with the ingredients
function getMealList(){
  let searchInputTxt = document.getElementById('search-input').value.trim();
  fetch(`https://www.themealdb.com/api/json/v1/1/filter.php?i=${searchInputTxt}`)
    .then(response => response.json())
    .then(data => {
      let html = "";
      if(data.meals){
        data.meals.forEach(meal => {
          html += `
            <div class = "meal-item" data-id = "${meal.idMeal}">
              <div class = "meal-img">
                <img src = "${meal.strMealThumb}" alt = "food">
              </div>
              <div class = "meal-name">
                <h3>${meal.strMeal}</h3>
                <a href = "#" class = "recipe-btn">Get Recipe</a>
              </div>
            </div>
          `;
        });
        mealList.classList.remove('notFound');
      } else{
        html = "Sorry, we didn't find any meal!";
        mealList.classList.add('notFound');
      }

      mealList.innerHTML = html;
    });
}
```

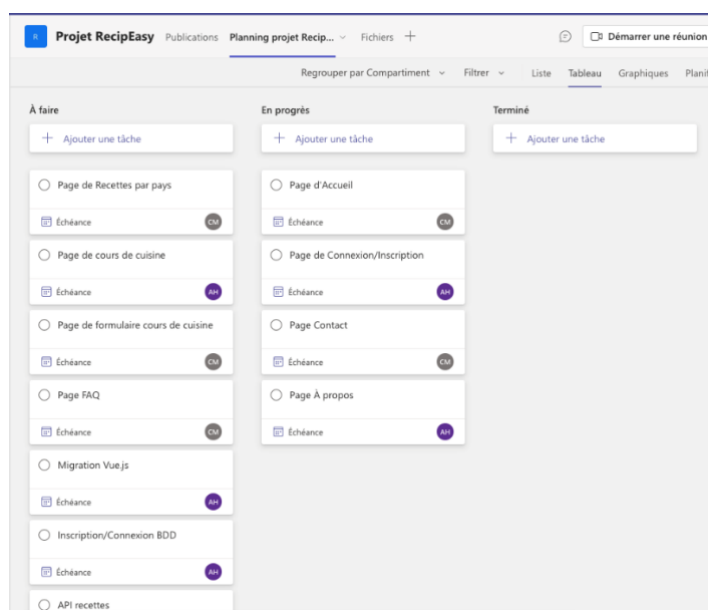
V. La Méthode Agile

Pour mener à bien le projet, nous avons opté pour la Méthodologie Agile. Cette méthodologie basée sur une idée simple nous permet de se fixer des objectifs à court terme. En effet, en général, nous savons que planifier la totalité de notre projet dans les moindres détails avant de le développer peut non seulement nous faire perdre un temps considérable, mais également être contre-productif.

De plus, en utilisant la méthodologie Agile en entreprise, nous nous sommes rendues compte que cette méthode était parfaite pour les projets informatiques : le projet est divisé en plusieurs sous-projets, et une fois l'objectif actuel atteint, on passe au suivant, et ce jusqu'à l'accomplissement de l'objectif final. Cette approche beaucoup plus flexible nous poussait donc à utiliser cette méthode pour notre projet Web.

Ainsi, nous avons tout de suite pu avoir un semblant de planning du projet afin d'avoir un visuel sur l'avancement du projet pour pouvoir le rendre à temps. Pour ce faire, nous avons créé un « Task » sur Teams qui ressemble au tableau Jira que nous utilisons généralement en entreprise avec trois colonnes essentielles : À faire, en progrès et terminé.

Voici le planning en début de projet :

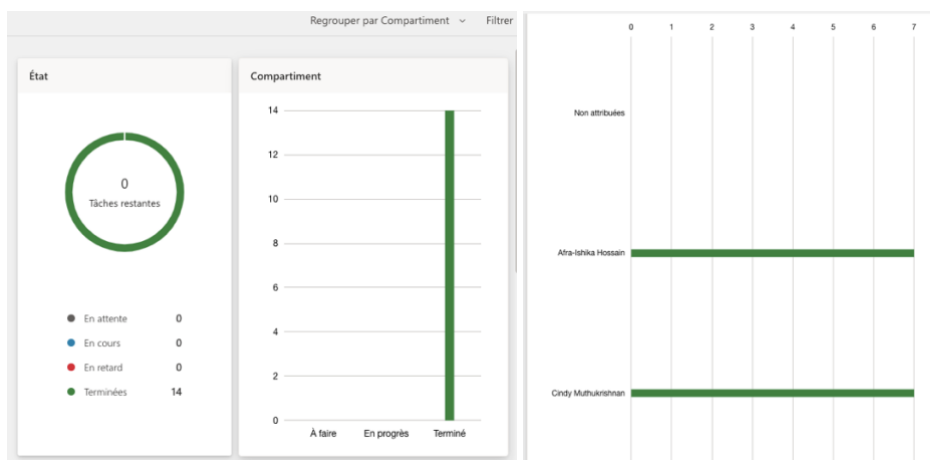


Projet RecipEasy Publications Planning projet Recip... Fichiers + Démarrer une réunion				
Actives Filtre Liste Tableau Graphiques Planification				
Titre de la tâche	Attribué à	Priorité	Délai	Compartimen
<input type="radio"/> Nouvelle tâche				À faire
<input type="radio"/> Animation réponse FAQ	Cindy Muthukrishnan			À faire
<input type="radio"/> Enregistrement cours BDD	Afra-Ishika Hossain			À faire
<input type="radio"/> Envoie email contact	Afra-Ishika Hossain			À faire
<input type="radio"/> API recettes	Cindy Muthukrishnan			À faire
<input type="radio"/> Inscription/Connexion BDD	Afra-Ishika Hossain			À faire
<input type="radio"/> Migration Vue.js	Afra-Ishika Hossain			À faire
<input type="radio"/> Page FAQ	Cindy Muthukrishnan			À faire
<input type="radio"/> Page de formulaire cours de cuisine	Cindy Muthukrishnan			À faire
<input type="radio"/> Page de cours de cuisine	Afra-Ishika Hossain			À faire
<input type="radio"/> Page de Recettes par pays	Cindy Muthukrishnan			À faire
<input type="radio"/> Page Contact	Cindy Muthukrishnan			En progrès
<input type="radio"/> Page À propos	Afra-Ishika Hossain			En progrès
<input type="radio"/> Page de Connexion/Inscription	Afra-Ishika Hossain			En progrès
<input type="radio"/> Page d'Accueil	Cindy Muthukrishnan			En progrès

Comme vous pouvez le voir, chaque tâche est attribuée à une personne. Toute l'équipe participe à l'ensemble des tâches à mettre en place, que ce soit sur le graphisme ou encore le code.

La personne est donc libre de bouger ses tâches entre les trois colonnes par rapport à son avancement.

De même, la section « Graphique » nous permet d'avoir un meilleur aperçu de l'avancement du projet. Ainsi à la fin du projet, voici les graphes :



Nous apercevons bien que toutes les tâches sont terminées, et l'attribution du nombre de tâches de chacune.

De plus, nous n'avions pas jugé cela nécessaire d'effectuer des « Daily Meeting » puisque nous ne sommes que deux dans le projet et que nous nous voyions tous les jours en classe. Cependant, nous avons effectué des réunions chaque week-end afin de faire un point sur les sujets en cours et les difficultés ressenties par chacune.

Conclusion

Pour conclure, nous avons réellement apprécié créer notre site web de recettes de cuisine. Ce projet nous a permis de développer une plateforme beaucoup plus approfondie en utilisant d'autres langages de programmations par rapport à l'an dernier. Nous avons appris à utiliser le Framework Vue.js pour le côté client et Node.js pour le côté serveur, ce qui s'avérait être un véritable challenge pour notre part. Par ailleurs, nous avons mis en place la méthode agile qui nous a permis de mener à bien notre projet. Aussi, nous avons pu respecter l'ensemble du cahier des charges que nous avons établi au commencement et livrer notre site web dans les temps impartis.

Concernant les axes d'amélioration, nous pensons que créer un forum de discussion entre les utilisateurs du site serait une fonctionnalité utile. Cela leur permettrait d'échanger des astuces entre eux mais aussi de partager leurs savoir-faire culinaires et leurs retours d'expériences des cours de cuisine.