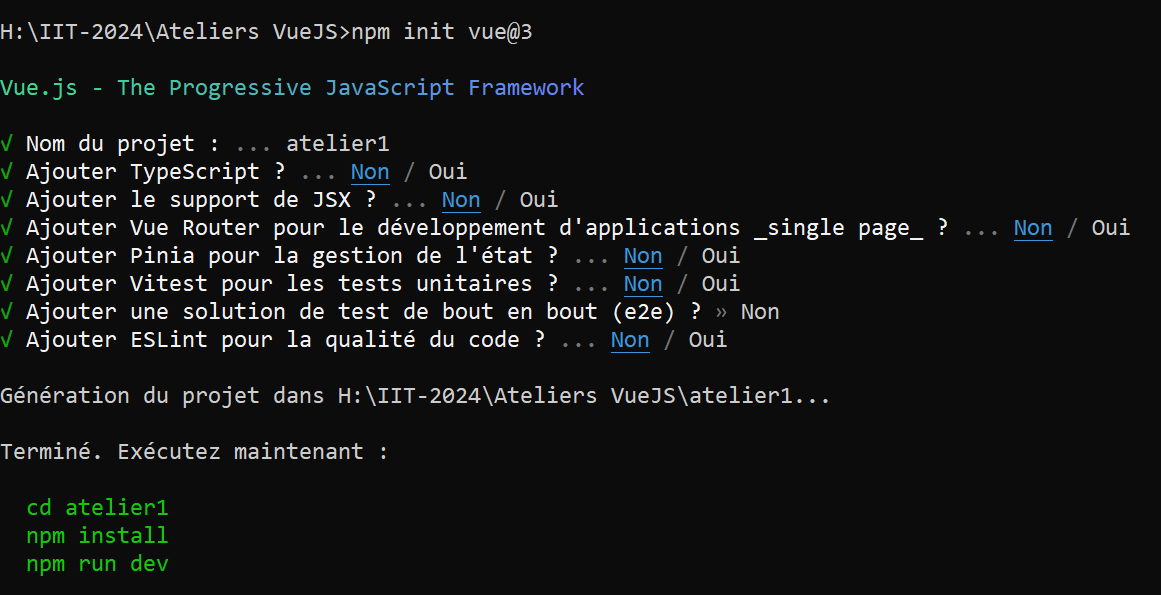
**Atelier 3 : Initiation à Vuejs : State et Props**

1. **Créer un Nouveau Projet VUEJS version 3**

**npm init vue@3**



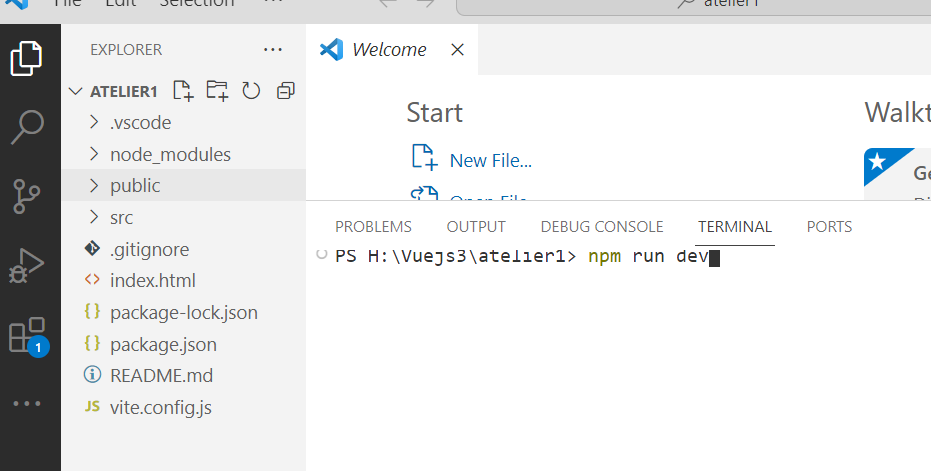
**cd atelier1**

**npm install**

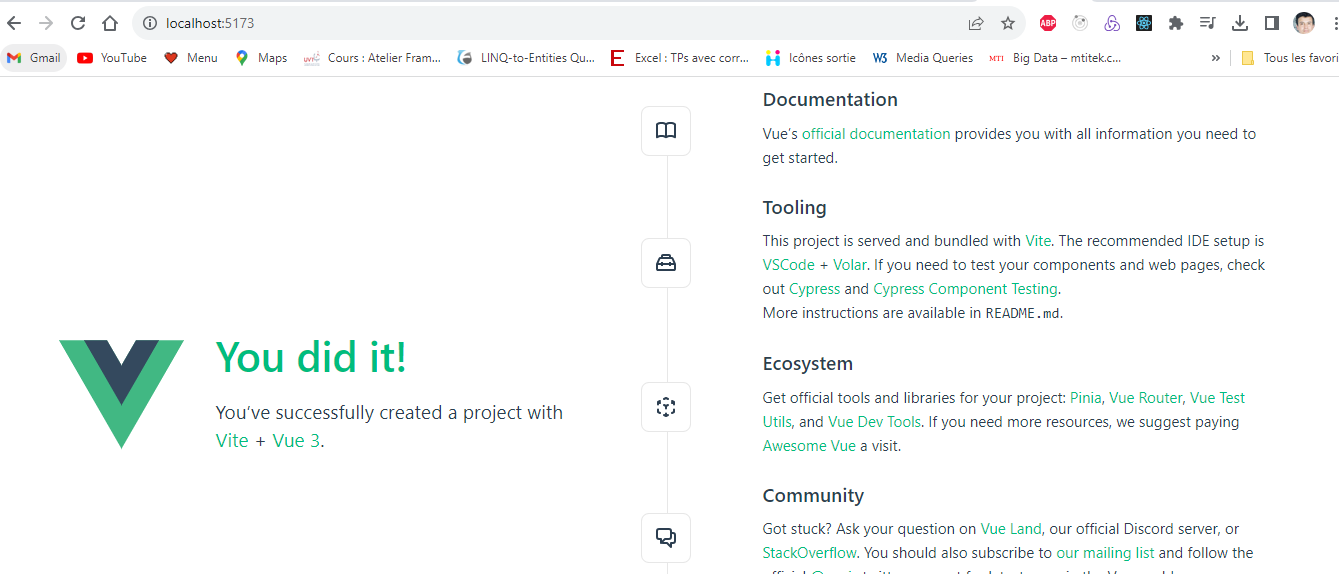
**Code .**

**Terminal🡪New Terminal**

**npm run dev**



**Résultat :**



1. **States**

**Sous le dossier src/components créer le composant Personne.vue**

<script setup>

import { ref } from 'vue';

const nom = ref('Mohamed')

const age=ref(22)

</script>

<template>

  <div>

    <h1>Welcome</h1>

    <h3>Je m'appelle {{ nom }} </h3>

    <div class="item"> j'ai {{age}} ans </div>

  </div>

</template>

<style scoped>

h3{

  font-weight: 500;

  font-size: 2.6rem;

  position: relative;

  top: -10px;

}

.item {

  margin-top: 2rem;

  font-size: 20px;

  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

  font-style: italic;

}

</style>

**Pour afficher le composant Personne, on doit y faire appel dans le composant APP.vue comme suit :**

**Composant App.vue**

<script setup>

import Personne from './components/Personne.vue';

</script>

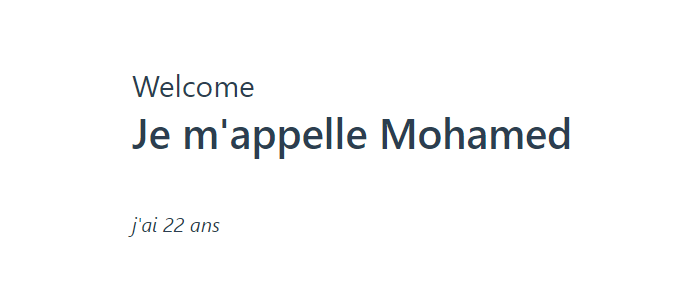
<template>

  <Personne/>

</template>

<style scoped>

</style>

**Résultat :** 

* **State : Affichage d’un objet**

**Sous le dossier src/components créer le composant Utilisateur.vue**

<script setup>

import { ref } from 'vue';

const user=ref({

  email:"IIT2024@gmail.com",

  adresse:"Campus Elons"

})

</script>

<template>

  <div>

    <h1>{{user.email}}</h1>

    <h3>Adresse: {{ user.adresse}} </h3>

  </div>

</template>

<style scoped>

</style>

Composant App.vue

import Personne from './components/Personne.vue';

import Utilisateur from './components/Utilisateur.vue'

</script>

<template>

  <Personne/>

  <Utilisateur/>

</template>

<style scoped>

</style>

* **State : Affichage du tableau d’objets**

Ajouter des snippets

Vue VSCode Snippets

Vbase-3-setup

Installer bootstrap

Accéder au site https://getbootstrap.com/

npm i bootstrap

Dans le fichier main.js:

import 'bootstrap/dist/css/bootstrap.css';

* Créer le composant Listlivres.vue dans le dossier src/components

<script setup>

import { ref } from 'vue';

const livres=ref([

{

  "isbn": "42368756",

  "titre": "ReactJS",

  "auteur":"Romain Rissouan",

  "annedition": 2021,

  "prix": 33,

  "couverture": "https://m.media-amazon.com/images/I/61vg+0-1yGL.\_SY466\_.jpg",

},

{

   "isbn": "123698745",

  "titre": "Les VPNS",

  "auteur":"Stephan@gmail.com",

  "annedition": 2020,

  "prix": 55,

  "couverture": "http://res.cloudinary.com/iset-sfax/image/upload/v1696761358/images/9782100491155\_internet\_w290.jpg.jpg",

},

{

  "isbn": "54894",

  "titre": "Techniques de Référencement Web",

  "auteur":"Patrick Wampe",

  "annedition": 2020,

  "prix": 55,

  "couverture": "https://blog.internet-formation.fr/wp-content/uploads/2018/04/1e-4e-couverture-livre-SEO.jpg",

},

{

   "isbn": "5464sf",

  "titre": "VueJS",

  "auteur":"Alexandra Martin",

  "annedition": 2020,

  "prix": 20,

  "couverture": "https://m.media-amazon.com/images/I/718QqTspltL.\_SY466\_.jpg",

},

{

   "isbn": "65465445",

  "titre": "Laravel",

  "auteur":"Eric Sarrion",

  "annedition": 2020,

  "prix": 50,

  "couverture": "https://images.booksense.com/images/132/218/9798554218132.jpg",

}

])

</script>

<template>

  <div>

    <h3>LISTE DES LIVRES DE LA BIBLIOTHEQUE</h3>

    <table class="table table-striped">

      <thead>

          <tr>

            <th>ISBN</th>

            <th>Image</th>

            <th>Titre Livre</th>

            <th>Auteur</th>

            <th>Année Edition</th>

            <th>Prix</th>

          </tr>

      </thead>

      <tbody>

       <tr v-for="liv in livres" :key="liv.isbn">

        <td>{{liv.isbn}}</td>

        <td><img :src="liv.couverture" :alt=liv.titre width="100" /></td>

        <td>{{liv.titre}}</td>

        <td>{{liv.auteur}}</td>

        <td>{{liv.annedition}}</td>

        <td>{{liv.prix}}</td>

      <td><button class="btn btn-danger">Supprimer</button></td>

       </tr>

      </tbody>

    </table>

  </div>

</template>

<style scoped>

</style>

* Composant App.vue

<script setup>

import Listlivres from "./components/Listlivres.vue"

</script>

<template>

  <Listlivres/>

</template>

<style scoped>

</style>



1. **Les props**

Créer un composant Affichelivres.vue sous le dossier components

<script setup>

**const props=defineProps(['livres'])**

</script>

<template>

  <div>

    <h3>LISTE DES LIVRES DE LA BIBLIOTHEQUE</h3>

    <table class="table table-striped">

      <thead>

          <tr>

            <th>ISBN</th>

            <th>Image</th>

            <th>Titre Livre</th>

            <th>Auteur</th>

            <th>Année Edition</th>

            <th>Prix</th>

          </tr>

      </thead>

      <tbody>

        <tr v-for="liv in livres" :key="liv.isbn">

                <td>{{liv.isbn}}</td>

                <td><img :src="liv.couverture" :alt=liv.titre width="100" /></td>

                <td>{{liv.titre}}</td>

                <td>{{liv.auteur}}</td>

                <td>{{liv.annedition}}</td>

                <td>{{liv.prix}}</td>

                <td>

                    <button class="btn btn-danger">Supprimer</button></td>

            </tr>

      </tbody>

    </table>

  </div>

</template>

<style scoped>

</style>

Créer le composant Listlivres.vue sous le dossier components

<script setup>

import { ref } from 'vue';

import Affichelivres from './Affichelivres.vue';

const livres=ref([

{

  "isbn": "42368756",

  "titre": "ReactJS",

  "auteur":"Romain Rissouan",

  "annedition": 2021,

  "prix": 33,

  "couverture": "https://m.media-amazon.com/images/I/61vg+0-1yGL.\_SY466\_.jpg",

},

{

   "isbn": "123698745",

  "titre": "Les VPNS",

  "auteur":"Stephan@gmail.com",

  "annedition": 2020,

  "prix": 55,

  "couverture": "http://res.cloudinary.com/iset-sfax/image/upload/v1696761358/images/9782100491155\_internet\_w290.jpg.jpg",

},

{

  "isbn": "54894",

  "titre": "Techniques de Référencement Web",

  "auteur":"Patrick Wampe",

  "annedition": 2020,

  "prix": 55,

  "couverture": "https://blog.internet-formation.fr/wp-content/uploads/2018/04/1e-4e-couverture-livre-SEO.jpg",

},

{

   "isbn": "5464sf",

  "titre": "VueJS",

  "auteur":"Alexandra Martin",

  "annedition": 2020,

  "prix": 20,

  "couverture": "https://m.media-amazon.com/images/I/718QqTspltL.\_SY466\_.jpg",

},

{

   "isbn": "65465445",

  "titre": "Laravel",

  "auteur":"Eric Sarrion",

  "annedition": 2020,

  "prix": 50,

  "couverture": "https://images.booksense.com/images/132/218/9798554218132.jpg",

}

])

</script>

<template>

  <div>

**<Affichelivres :livres="livres"/>**

  </div>

</template>

<style scoped>

</style>

Composant App.vue

<template>

  <div>

    <Listlivres/>

  </div>

</template>

<script setup>

import Listlivres from './components/Listlivres.vue';

</script>

<style lang="scss" scoped>

</style>

1. Bindings et événements

Créer un composant appelé Inscription.vue sous le dossier components

<template>

    <div>

<h3>Inscription au site</h3>

<div class="mb-3">

  <label for="Email" class="form-label">Email address</label>

  <input type="email" class="form-control" id="email"  v-model="email" placeholder="name@example.com">

</div>

<div class="mb-3">

  <label for="Nom" class="form-label">Nom : </label>

  <input type="text" class="form-control" id="nom"  v-model="nom" placeholder="Votre nom">

</div>

<div class="mb-3">

  <label for="Prenom" class="form-label">Prénom : </label>

  <input type="text" class="form-control" id="prenom"  v-model="prenom" placeholder="Votre prénom">

</div>

<div class="mb-3">

  <label for="Password" class="form-label">Password : </label>

  <input type="text" class="form-control" id="password"  v-model="password" placeholder="Password">

</div>

    </div>

</template>

<script setup>

import { ref } from 'vue';

const email = ref('');

const nom=ref('')

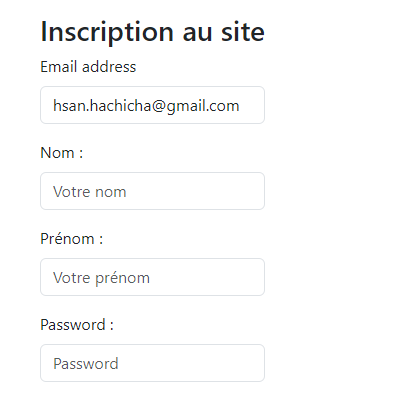
const prenom=ref ('')

const password=ref('')

</script>

<style lang="scss" scoped>

</style>



Ajouter un bouton de confirmation

<template>

    <div>

<h3>Inscription au site</h3>

<div class="mb-3">

  <label for="Email" class="form-label">Email address</label>

  <input type="email" class="form-control" id="email"  v-model="email" placeholder="name@example.com">

</div>

<div class="mb-3">

  <label for="Nom" class="form-label">Nom : </label>

  <input type="text" class="form-control" id="nom"  v-model="nom" placeholder="Votre nom">

</div>

<div class="mb-3">

  <label for="Prenom" class="form-label">Prénom : </label>

  <input type="text" class="form-control" id="prenom"  v-model="prenom" placeholder="Votre prénom">

</div>

<div class="mb-3">

  <label for="Password" class="form-label">Password : </label>

  <input type="text" class="form-control" id="password"  v-model="password" placeholder="Password">

</div>

<button type="button" @click="handleconfirm" class="btn btn-success">Afficher</button>

    </div>

</template>

<script setup>

import { ref } from 'vue';

const email = ref('');

const nom=ref('')

const prenom=ref ('')

const password=ref('')

const handleconfirm=()=>{

    alert("Inscription effectuée avec succés, votre nom est : " + nom.value + " Prénom est :" + prenom.value + " Email est :" + email.value )

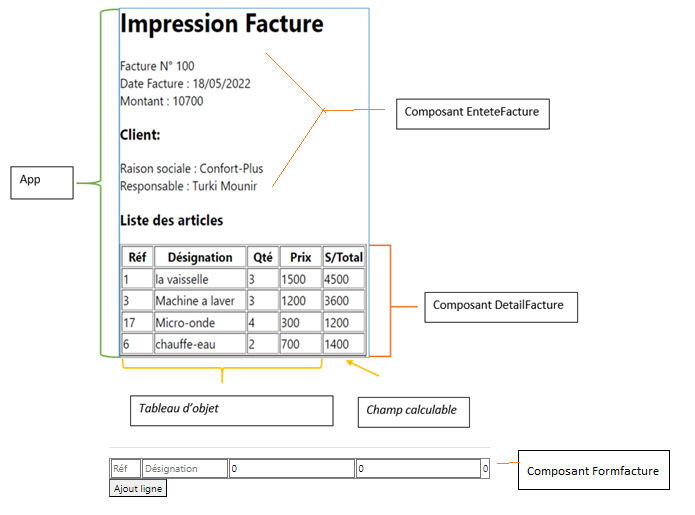
}

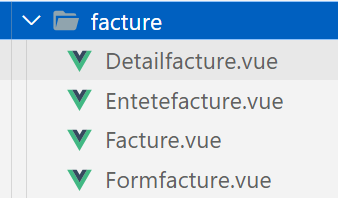
</script>

<style lang="scss" scoped>

</style>

Les CRUDs avec Vue JS





Composant Facture.vue

<template>

    <div>

<Entetefacture :fact="facture"/>

<Detailfacture :articles="articles"/>

<Formfacture :ajoutLigne="ajoutLigne"/>

    </div>

</template>

<script setup>

import { ref } from "vue";

import Entetefacture from "./Entetefacture.vue";

import Formfacture from "./Formfacture.vue";

import Detailfacture from "./Detailfacture.vue";

const facture =ref({

                numfact:"100",

                datefact:"18/05/2022",

                raisoc:"Confort-Plus",

                responsable:"Turki Mounir"

            });

            const articles=ref([

                {reference:"1",designation:"la vaisselle",qte:3,prix:1500},

                {reference:"3",designation:"Machine a laver",qte:3,prix:1200},

                {reference:"17",designation:"Micro-onde",qte:4,prix:300}

            ]);

    const ajoutLigne =(ref,des,qt,pr)=>{

   //on construit la nouvelle ligne

         let ligne={reference:ref,designation:des,qte:qt,prix:pr}

//on change le state d&#39;articles en lui ajoutant la ligne

        articles.value.push(ligne)

 }

</script>

<style lang="scss" scoped>

</style>

Composant Entetefacture.vue

<template>

    <div>

        <div>

        <h1>Impression Facture</h1>

        Facture N° {{fact.numfact}}<br/>

        Date Facture : {{fact.datefact}}<br/>

        </div>

        <div>

            <h3>Client:</h3>

            Raison sociale : {{fact.raisoc}}<br/>

            Responsable : {{fact.responsable}}

        </div>

     </div>

</template>

<script setup>

const props=defineProps(['fact'])

</script>

<style lang="scss" scoped>

</style>

Composant Detailfacture.vue

<template>

              <div>

           <h3>Liste des articles</h3>

            <table class="table table-striped">

            <thead>

        <tr>

          <th width="30px">Référence</th>

          <th width="40px">Désignation</th>

          <th width="25px">Qté</th>

          <th width="35px">Prix</th>

          <th width="35px">Montant</th>

        </tr>

        </thead>

            <tbody>

            <tr v-for ="art in articles" :key="art.reference">

            <td width="40px">{{art.reference}}</td>

             <td width="125px"> {{art.designation}}</td>

             <td width="40px"> {{art.qte}}</td>

             <td width="55px"> {{art.prix}}</td>

             <td width="55px"> {{art.prix\*art.qte}}</td>

            </tr>

      </tbody></table>

        </div>

</template>

<script setup>

const props = defineProps(['articles'])

</script>

<style lang="scss" scoped>

</style>

Composant Formfacture.vue

<template>

    <div>

           <table border="1">

    <tbody>

  <tr>

   <td> <input type="text" placeholder="Réf"

    name="reference"

    v-model="reference"

    size="1"/>

    </td>

   <td>

    <input type="text" placeholder="Désignation"

   name="des"

   v-model="des"

   size="12"/> </td>

   <td>

    <input type="number"

   name="qte"

   v-model="qte"

   size="2"/> </td>

   <td>

    <input type="number" name="pr"

   v-model="prix"/>

   </td>

    <td width="55px"> {{prix\*qte}}</td>

   </tr>

   </tbody>

  </table>

    <button @click="validerLigne">

       Ajout ligne

    </button>

  </div>

</template>

<script setup>

import { ref } from 'vue';

const props = defineProps(['ajoutLigne'])

        const reference=ref()

        const des=ref()

        const qte=ref(0)

        const prix=ref(0)

        const validerLigne=()=>{

          props.ajoutLigne(reference.value,des.value,qte.value,prix.value);

          reference.value=''

          des.value=""

          qte.value=0

          prix.value=0

        }

</script>

<style lang="scss" scoped>

</style>

Composant App.vue

<script setup>

import Facture from "./components/facture/Facture.vue";

</script>

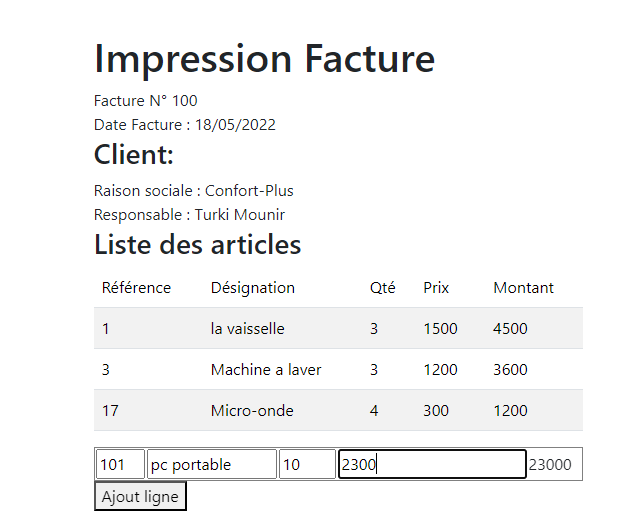
<template>

  <Facture/>

</template>

<style scoped>

</style>



Supprimer un article

Composant Detailfacture.vue

<template>

    <div>

 <h3>Liste des articles</h3>

  <table class="table table-striped">

  <thead>

<tr>

<th width="30px">Référence</th>

<th width="40px">Désignation</th>

<th width="25px">Qté</th>

<th width="35px">Prix</th>

<th width="35px">Montant</th>

<th width="40px">Actions</th>

</tr>

</thead>

  <tbody>

  <tr v-for ="**(art, index)** in articles" :key="art.reference">

  <td width="40px">{{art.reference}}</td>

   <td width="125px"> {{art.designation}}</td>

   <td width="40px"> {{art.qte}}</td>

   <td width="55px"> {{art.prix}}</td>

   <td width="55px"> {{art.prix\*art.qte}}</td>

   <td width="40px">

            <button @click="deleteArticle(index)" class="btn btn-danger btn-sm">Supprimer</button>

  </td>

  </tr>

</tbody></table>

</div>

</template>

<script setup>

const props = defineProps(['articles']);

const deleteArticle = (index) => {

  // on utilise l'index de l'article à supprimer

  if (index >= 0 && index < props.articles.length) {

    props.articles.splice(index, 1);

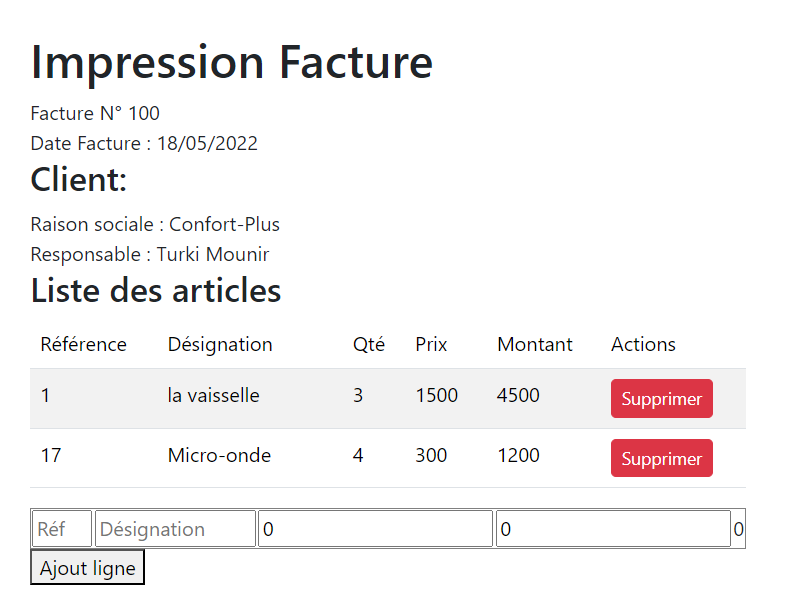
  }

};

</script>

<style lang="scss" scoped>

</style>



Modifier un article

Composant Detailfacture.vue

Nous allons ajouter un test editMode à chaque article(quantité et prix) pour suivre son état d'édition. Lorsqu’on clique sur le bouton "Modifier", toggleEditMode est appelé pour basculer entre l'affichagesimple des valeurs actuelles et des champs de saisie (input). Lorsqu’on clique sur "Enregistrer", la méthode updateArticle est appelée pour mettre à jour l'article avec les nouvelles valeurs et désactiver le mode édition.

<template>

    <div>

 <h3>Liste des articles</h3>

  <table class="table table-striped">

  <thead>

<tr>

<th width="30px">Référence</th>

<th width="40px">Désignation</th>

<th width="25px">Qté</th>

<th width="35px">Prix</th>

<th width="35px">Montant</th>

<th width="100px">Actions</th>

</tr>

</thead>

<tbody>

        <tr v-for="(art, index) in articles" :key="art.reference">

          <td width="40px">{{ art.reference }}</td>

          <td width="125px">{{ art.designation }}</td>

          <td width="40px">

**<span v-if="!art.editMode">{{ art.qte }}</span>**

            <input

              v-else

              type="number"

              v-model="art.qte"

              @change="updateArticle(index)"

            />

          </td>

          <td width="55px">

**<span v-if="!art.editMode">{{ art.prix }}</span>**

            <input

              v-else

              type="number"

              v-model="art.prix"

              @change="updateArticle(index)"

            />

         </td>

          <td width="55px">{{ art.prix \* art.qte }}</td>

          <td width="100px">

            <button

                @click="art.editMode ? enregistrer(art) : toggleEditMode(art)"

                class="btn btn-primary btn-sm"

                >

                {{ art.editMode ? "Enregistrer" : "Modifier" }}

                </button>

                <button

              @click="deleteArticle(index)"

              class="btn btn-danger btn-sm"

            >

              Supprimer

            </button>

          </td>

        </tr>

      </tbody>

    </table>

</div>

</template>

<script setup>

const props = defineProps(['articles']);

const deleteArticle = (index) => {

  // on utilise l'index de l'article à supprimer

  if (index >= 0 && index < props.articles.length) {

    props.articles.splice(index, 1);

  }

};

const toggleEditMode = (article) => {

  article.editMode = !article.editMode;

};

const updateArticle = (index) => {

  if (index >= 0 && index < props.articles.length) {

    const article = props.articles[index];

    article.montant = article.prix \* article.qte;

  }

};

// Initialise l'état "editMode" pour chaque article

props.articles.forEach((article) => {

  article.editMode = false;

});

const enregistrer =(article)=>{

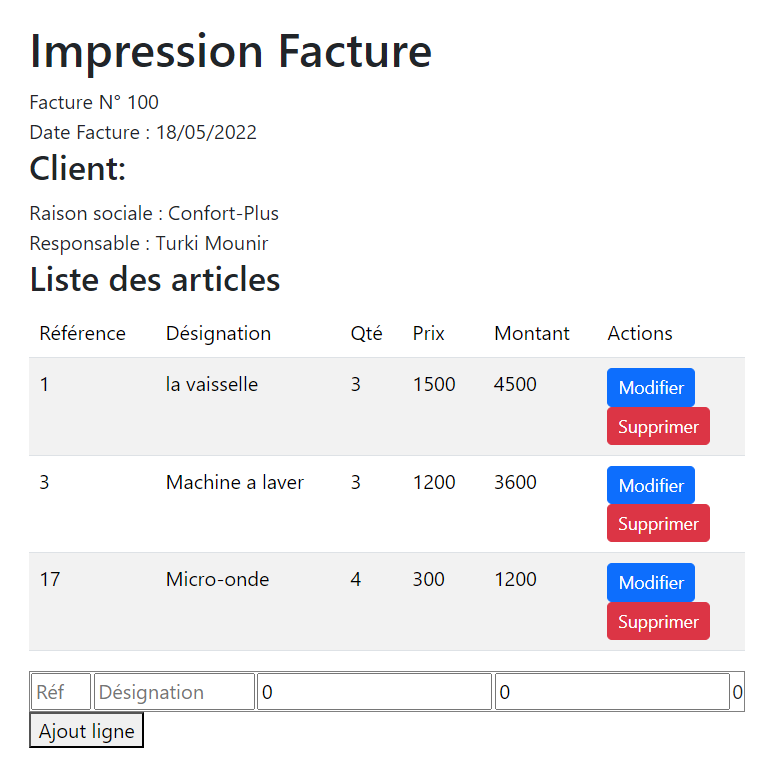
    toggleEditMode(article); // Désactive le mode édition après la mise à jour

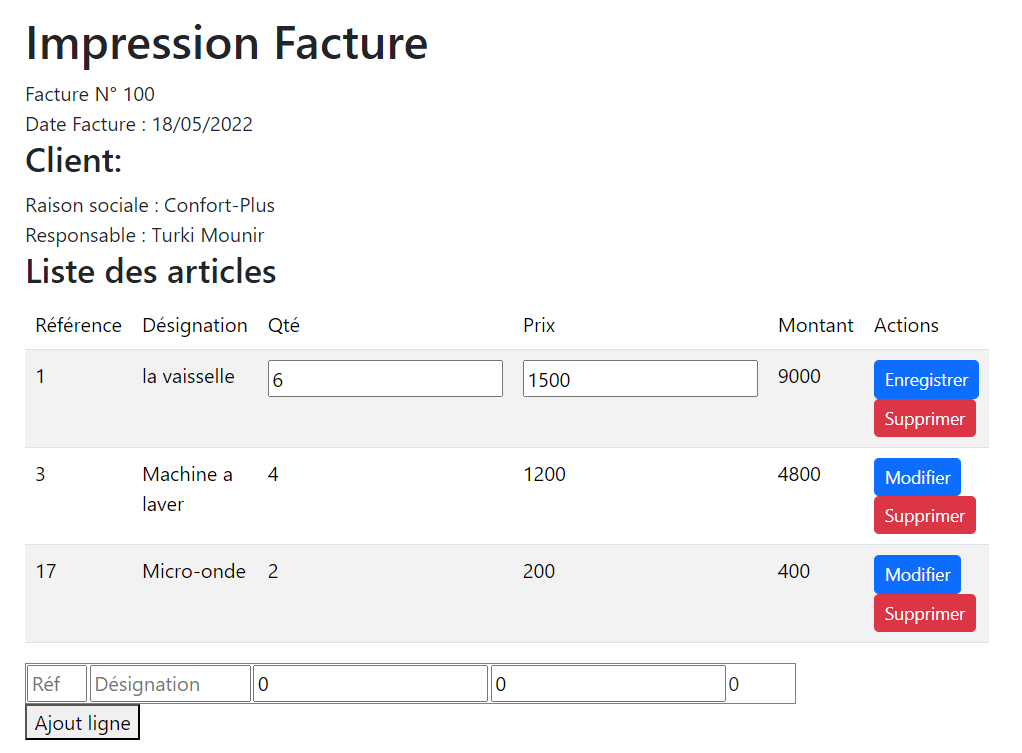
}

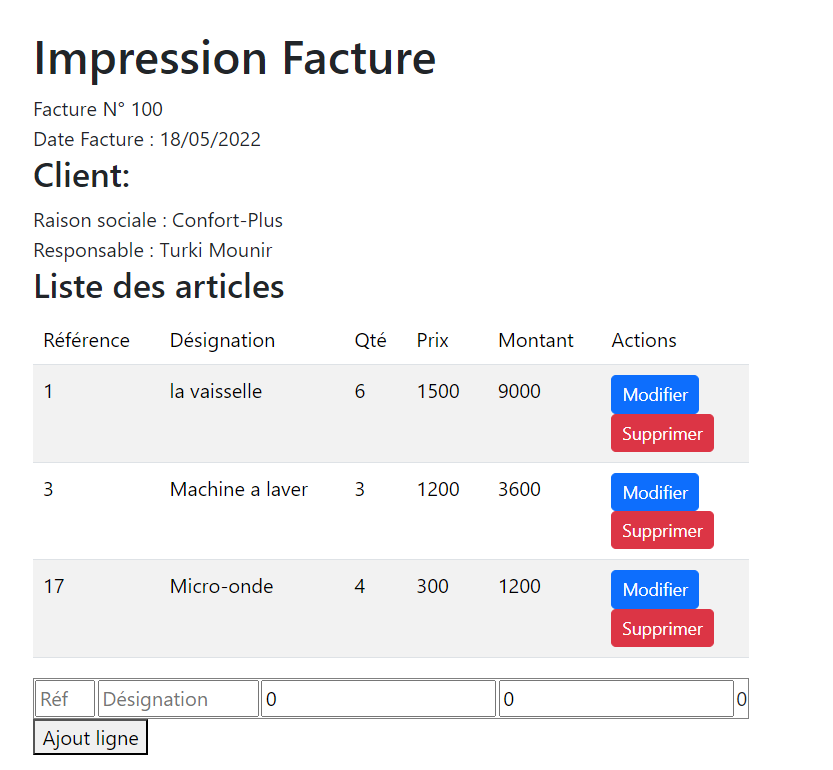
</script>

<style lang="scss" scoped>

</style>





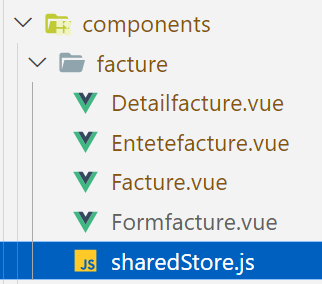


Calcul Montant Total

Dans chacun des composants on va importer le module sharedStore et on va utiliser le tableau articles depuis ce module qu’on va créer.

Créez un module sharedStore contenant le tableau articles et le gestionnaire de modification.

On va y créer une fonction pour recalculer le montant total et mettre à jour le ref partagé.



import { ref } from 'vue';

export const articles = ref([

  { reference: '1', designation: 'la vaisselle', qte: 3, prix: 1500 },

  { reference: '3', designation: 'Machine a laver', qte: 3, prix: 1200 },

  { reference: '17', designation: 'Micro-onde', qte: 4, prix: 300 },

]);

export const montantTotal = ref(0);

export const recalculateMontantTotal = () => {

    const total = articles.value.reduce((total, article) => total + article.qte \* article.prix, 0);

    montantTotal.value = total;

  };

Composant Facture.vue

<template>

    <div>

<Entetefacture :fact="facture" />

**<Detailfacture  />**

<Formfacture :ajoutLigne="ajoutLigne"/>

    </div>

</template>

<script setup>

import { ref } from "vue";

import Entetefacture from "./Entetefacture.vue";

import Formfacture from "./Formfacture.vue";

import Detailfacture from "./Detailfacture.vue";

**import { articles } from './sharedStore';**

const facture =ref({

                numfact:"100",

                datefact:"18/05/2022",

                raisoc:"Confort-Plus",

                responsable:"Turki Mounir"

            });

    const ajoutLigne =(ref,des,qt,pr)=>{

   //on construit la nouvelle ligne

         let ligne={reference:ref,designation:des,qte:qt,prix:pr}

//on change le state des articles en lui ajoutant la ligne

        articles.value.push(ligne)

 }

</script>

<style lang="scss" scoped>

</style>

Composant EnteteFacture.vue

<template>

    <div>

      <div>

        <h1>Impression Facture</h1>

        Facture N° {{ fact.numfact }}<br />

        Date Facture : {{ fact.datefact }}<br />

        Montant : {{ montantTotal }}

      </div>

      <div>

        <h3>Client:</h3>

        Raison sociale : {{ fact.raisoc }}<br />

        Responsable : {{ fact.responsable }}

      </div>

    </div>

  </template>

  <script setup>

import { onMounted } from 'vue';

import { montantTotal, recalculateMontantTotal } from './sharedStore';

const props = defineProps(['fact']);

onMounted(() => {

  recalculateMontantTotal();

});

</script>

  <style lang="scss" scoped>

  </style>

Composant Detailfacture.vue

<template>

    <div>

 <h3>Liste des articles</h3>

  <table class="table table-striped">

  <thead>

<tr>

<th width="30px">Référence</th>

<th width="40px">Désignation</th>

<th width="25px">Qté</th>

<th width="35px">Prix</th>

<th width="35px">Montant</th>

<th width="100px">Actions</th>

</tr>

</thead>

<tbody>

        <tr v-for="(art, index) in articles" :key="art.reference">

          <td width="40px">{{ art.reference }}</td>

          <td width="125px">{{ art.designation }}</td>

          <td width="40px">

            <span v-if="!art.editMode">{{ art.qte }}</span>

            <input

              v-else

              type="number"

              v-model="art.qte"

              @change="updateArticle(index)"

            />

          </td>

          <td width="55px">

             <span v-if="!art.editMode">{{ art.prix }}</span>

            <input

              v-else

              type="number"

              v-model="art.prix"

              @change="updateArticle(index)"

            />

         </td>

          <td width="55px">{{ art.prix \* art.qte }}</td>

          <td width="100px">

            <button

                @click="art.editMode ? enregistrer(art) : toggleEditMode(art)"

                class="btn btn-primary btn-sm"

                >

                {{ art.editMode ? "Enregistrer" : "Modifier" }}

                </button>

                <button

              @click="deleteArticle(index)"

              class="btn btn-danger btn-sm"

            >

              Supprimer

            </button>

          </td>

        </tr>

      </tbody>

    </table>

</div>

</template>

<script setup>

~~//const props = defineProps(['articles']);~~

import { articles, recalculateMontantTotal } from './sharedStore';

const deleteArticle = (index) => {

  // on utilise l'index de l'article à supprimer

  if (index >= 0 && index < **articles.value**.length) {

**articles.value**.splice(index, 1);

  }

  recalculateMontantTotal ()

};

const toggleEditMode = (article) => {

  article.editMode = !article.editMode;

};

const updateArticle = (index) => {

  if (index >= 0 && index < **articles.value**.length) {

    const article = **articles.value**[index];

    article.montant = article.prix \* article.qte;

  }

  recalculateMontantTotal ()

};

// Initialise l'état "editMode" pour chaque article

articles.value.forEach((article) => {

  article.editMode = false;

});

const enregistrer =(article)=>{

    toggleEditMode(article); // Désactive le mode édition après la mise à jour

}

</script>

<style lang="scss" scoped>

</style>

Composant Formfacture.vue

<template>

    <div>

           <table border="1">

    <tbody>

  <tr>

   <td> <input type="text" placeholder="Réf"

    name="reference"

    v-model="reference"

    size="1"/>

    </td>

   <td>

    <input type="text" placeholder="Désignation"

   name="des"

   v-model="des"

   size="12"/> </td>

   <td>

    <input type="number"

   name="qte"

   v-model="qte"

   size="2"/> </td>

   <td>

    <input type="number" name="pr"

   v-model="prix"/>

   </td>

    <td width="55px"> {{prix\*qte}}</td>

   </tr>

   </tbody>

  </table>

    <button @click="validerLigne">

       Ajout ligne

    </button>

  </div>

</template>

<script setup>

import { ref } from 'vue';

import { recalculateMontantTotal } from './sharedStore';

const props = defineProps(['ajoutLigne'])

        const reference=ref()

        const des=ref()

        const qte=ref(0)

        const prix=ref(0)

        const validerLigne=()=>{

          props.ajoutLigne(reference.value,des.value,qte.value,prix.value);

          reference.value=''

          des.value=""

          qte.value=0

          prix.value=0

          recalculateMontantTotal()

        }

</script>

<style lang="scss" scoped>

</style>

