



**KAOUTAR  
JABRI**

# **Compte rendu du projet To do list**



---

**BTS 2021-2023**

# Sommaire

---

- 01.** Introduction
- 02.** Le résultat attendu du projet
- 03.** Présentation et explication  
du code
- 04.** Conclusion

## **Introduction**

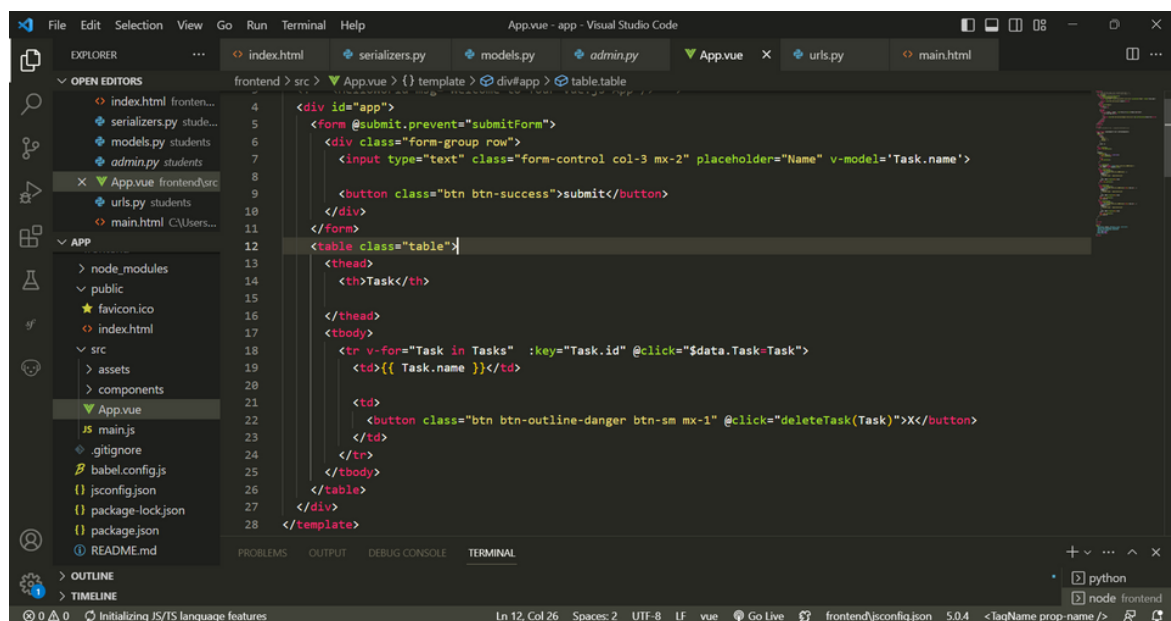
Ce projet est un site pour la prise de notes fait en Vue.js et Django rest Framework qui permet d'ajouter à la liste, de modifier et de supprimer mes éléments de la liste.

## **Résultat attendu**

Le but du projet est de pouvoir mettre en place les connaissances théoriques en Vue.js et Django rest pour pouvoir créer une application qui permet d'ajouter de modifier et de supprimer des éléments dans la liste.

# Presentation et explication du code

La partie HTML est composée d'un input pour pouvoir ajouter des éléments dans la liste, et un tableaux pour pouvoir les voir, supprimer ou les modifier.



```
App.vue - app - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  frontend > src > App.vue > {} template > div#app > table.table
  index.html frontend...
  serializers.py stude...
  models.py students
  admin.py students
  App.vue frontend/src
  urls.py students
  main.html C:\Users...
  APP
    node_modules
    public
      favicon.ico
      index.html
    src
      assets
      components
      App.vue
      main.js
      .gitignore
      babel.config.js
      jsconfig.json
      package-lock.json
      package.json
      README.md
  OUTLINE
  TIMELINE
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
  python
  node frontend

Ln 12, Col 26 Spaces: 2 UTF-8 LF vue Go Live frontend/jsconfig.json 5.0.4 <TagName prop-name />
```

```
<div id="app">
  <form @submit.prevent="submitForm">
    <div class="form-group row">
      <input type="text" class="form-control col-3 mx-2" placeholder="Name" v-model="Task.name">
      <button class="btn btn-success">submit</button>
    </div>
  </form>
  <table class="table">
    <thead>
      <th>Task</th>
    </thead>
    <tbody>
      <tr v-for="Task in Tasks" :key="Task.id" @click="$data.Task=Task">
        <td>{{ Task.name }}</td>
        <td>
          <button class="btn btn-outline-danger btn-sm mx-1" @click="deleteTask(Task)">X</button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

```

    async createTask(){
      await this.getTasks();
      await fetch('http://127.0.0.1:8000/Tasks/', {
        method: 'post',
        headers: {
          'content-type': 'application/json'
        },

        body: JSON.stringify(this.Task)
      });
      await this.getTasks();
    }
  }

```

Cette partie permet d'ajouter des nouveaux éléments à la liste

```

    async editTask(){
      await this.getTasks();
      await fetch('http://127.0.0.1:8000/Tasks/${this.Task.id}/', {
        method: 'put',
        headers: {
          'content-type': 'application/json'
        },

        body: JSON.stringify(this.Task)
      });
      await this.getTasks();
    }
  }

```

Cette partie permet d'ajouter des nouveaux éléments à la liste

```

    async deleteTask(){
      await this.getTasks();
      await fetch('http://127.0.0.1:8000/Tasks/${this.Task.id}/', {
        method: 'delete',
        headers: {
          'content-type': 'application/json'
        },

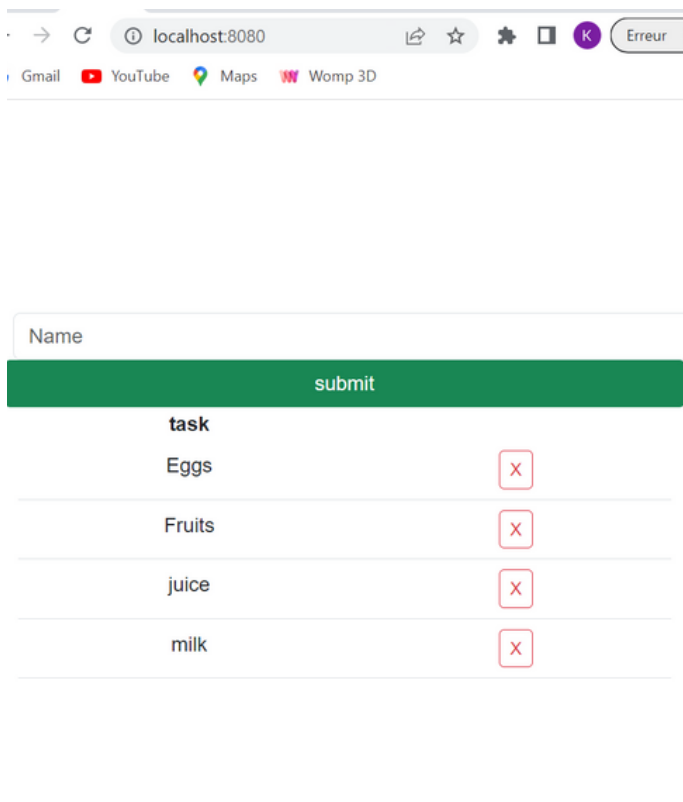
        body: JSON.stringify(this.Task)
      });
      await this.getTasks();
      this.Task = [];
    }
  }

```

Cette partie permet d'ajouter des nouveaux éléments à la liste

# Conclusion

---



Name	
submit	
task	
Eggs	x
Fruits	x
juice	x
milk	x

le résultat final est ainsi:

l'utilisateur peut ajouter dans sa liste ou supprimer, il peut aussi mettre à jour en cliquant sur l'élément