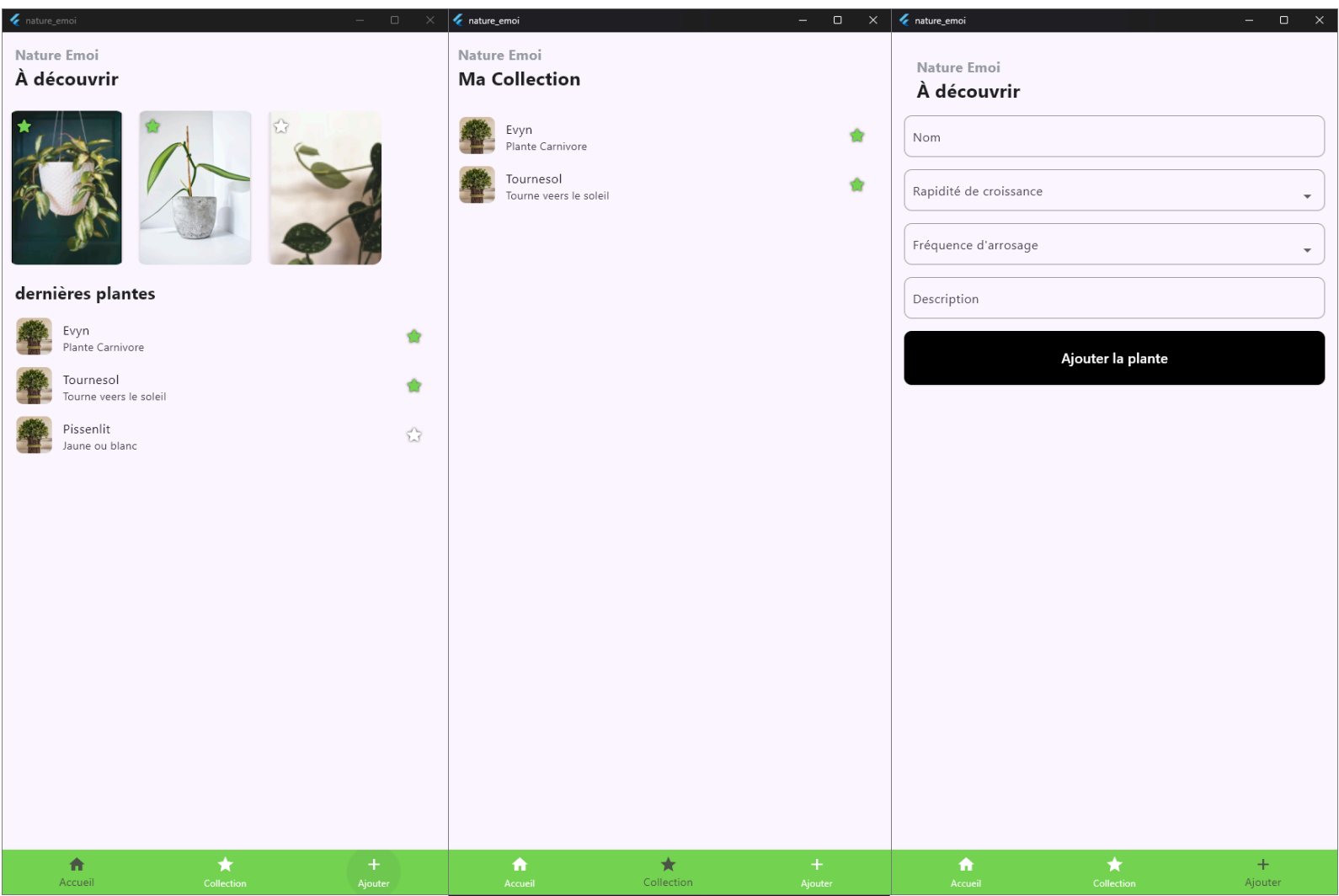


## B2 TP Nature'Emoi

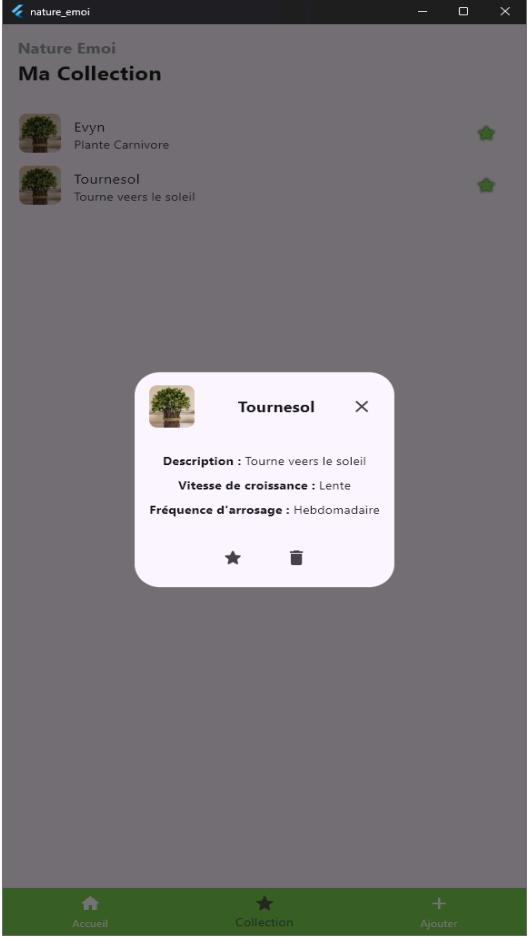
<b>B2 TP Nature'Emoi</b>	<b>1</b>
<b>1. Création du projet</b>	<b>3</b>
<b>2. Code de l'application</b>	<b>4</b>
<b>3. Conclusion</b>	<b>6</b>



Voici l'application terminée.  
Je vais vous expliquer comment  
je suis arrivé jusqu'ici, de la création  
du projet au code de celui-ci.

**Fonctionnalités principales :**

- Voir les plantes enregistrées
- Mettre une plante en favori
- Ajouter une plante à la BDD
- Consulter les détails d'une plante
- Supprimer une plante

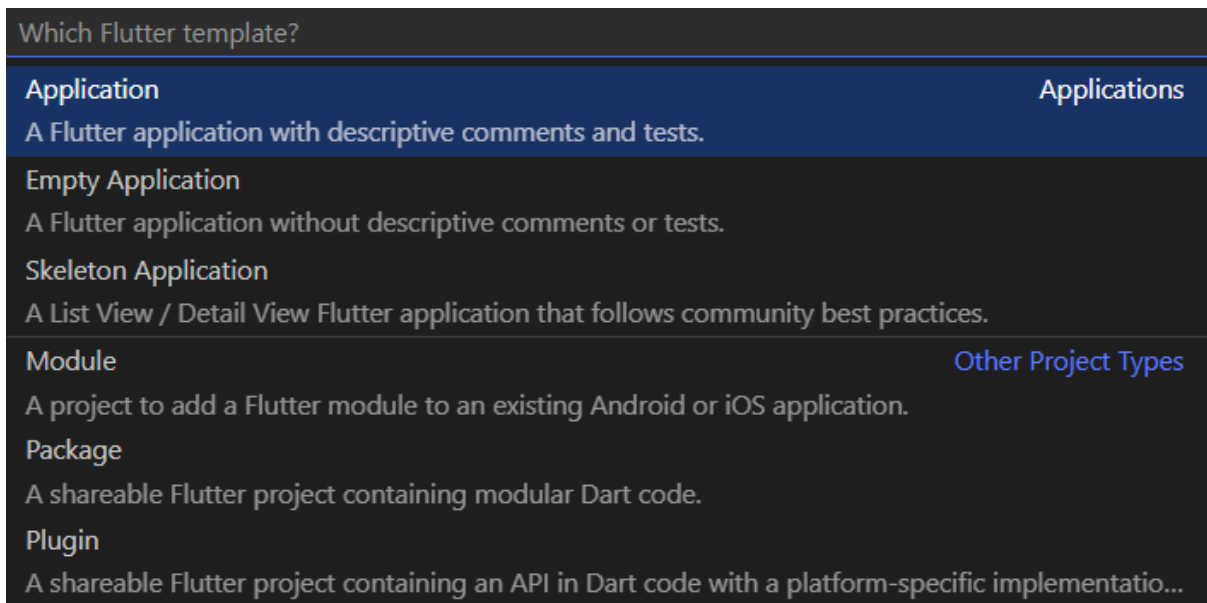


# 1. Création du projet

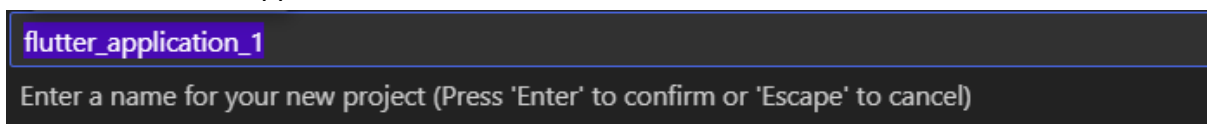
Avec le raccourci Ctrl + Shift + P et flutter d'installé vous allez avoir cette onglet avec Flutter: New Project cliquez dessus



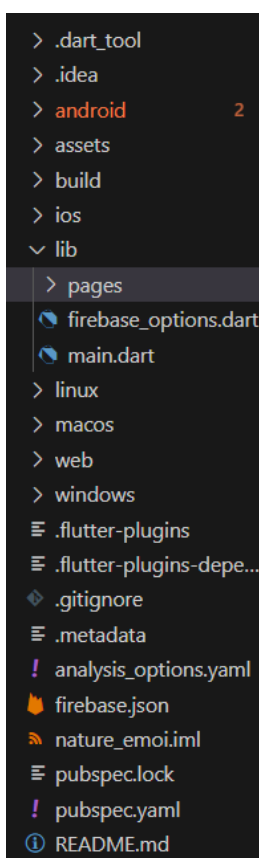
en cliquant dessus plusieurs option s'offre à vous je suis partie sur Empty Application



Entrez le nom de l'application



Voilà votre application est créé



le code de l'application doit se trouver dans lib en .dart

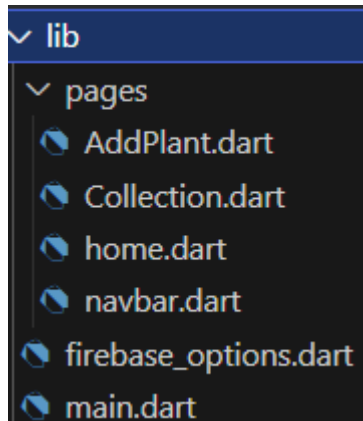
Voici mon fichier pubspec.yaml on voit quelques modifications.

- Les images par exemple doivent avoir leur chemin ici (elles ne sont pas gérées par firebase car payant)
- il y a aussi les imports de la BDD

```
1 name: nature_emoi
2 description: "A new Flutter project."
3 publish_to: 'none'
4 version: 0.1.0
5
6 environment:
7   sdk: ^3.5.1
8
9 dependencies:
10  flutter:
11    sdk: flutter
12  firebase_core: ^3.9.0
13  cloud_firestore: ^5.6.0
14
15 dev_dependencies:
16  flutter_test:
17    sdk: flutter
18  flutter_lints: ^4.0.0
19
20 flutter:
21   uses-material-design: true
22
23 assets:
24   - assets/Plante1.png
25   - assets/Plante2.png
26   - assets/Plante3.png
27   - assets/image.png
```

Pour la connexion BDD suivez les étapes sur le site de firebase

## 2. Code de l'application



Voici l'arborescence de l'app

Pour faire ce projet j'ai commencer par coder tout le visuel en préparant le fonctionnelle (mettre les bouton etc)

Une fois le visuel bon j'ai attaqué le fonctionnelle avec l'affichage des plante en fonction de la BDD à la place de les afficher en brut grâce à ces lignes de codes pour l'affichage des image en haut de la page accueil.

```
child: StreamBuilder<QuerySnapshot>(  
  stream:  
    FirebaseFirestore.instance.collection('Plante').snapshots(),  
  builder: (context, snapshot) {  
    if (snapshot.connectionState == ConnectionState.waiting) {  
      return const Center(child: CircularProgressIndicator());  
    }  
  
    if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {  
      return const Center(child: Text("Aucune plante disponible."));  
    }  
  
    return ListView.builder(  
      scrollDirection: Axis.horizontal,  
      itemCount: snapshot.data!.docs.length,  
      itemBuilder: (context, index) {  
        var plant = snapshot.data!.docs[index];  
        return buildPlantCard(  
          plant,  
          listimage[index %  
            listimage  
              .length]); // Passer les données de Firestore  
      },  
    );  
  },  
),  
),  
),
```

Pour l'ajout a la BDD la fonction ressemble à ça :

```
void _addPlant() async {  
  if (_formKey.currentState!.validate()) {  
    String name = _nameController.text;  
    String Describe = _DescribeController.text;  
  
    // Ajouter les données à Firestore  
    await FirebaseFirestore.instance.collection('Plante').add({  
      'Name': name,  
      'Grow': _selectedGrowthSpeed,  
      'Consumption': _selectedWaterSchedule,  
      'Describe': Describe,  
      'Favorite': false,  
    });  
  
    // Afficher un message de confirmation  
    ScaffoldMessenger.of(context).showSnackBar(  
      const SnackBar(content: Text('Plant added successfully')),  
    );  
  
    // Réinitialiser les champs  
    _nameController.clear();  
    setState(() {  
      _selectedGrowthSpeed = null; // Réinitialiser la sélection du dropdown  
    });  
    setState(() {  
      _selectedWaterSchedule = null; // Réinitialiser la sélection du dropdown  
    });  
    _DescribeController.clear();  
  }  
}
```

### 3. Conclusion

Pour conclure, ma méthode a été de travailler sur le visuel en premier, en préparant l'arrivée de la base de données et du fonctionnel. Cela m'a facilité le travail une fois le visuel terminé.

Ce TP m'a permis de découvrir de nombreuses nouvelles choses sur Flutter et d'approfondir les liaisons avec la base de données dans Flutter.