LABORATOIRE 1 : React Native

Exercise 1 : Installation

Objectif : Installer et configurer l'environnement de développement pour React Native.

Étapes :

1. **Installer**
2. **Installer Node.js et npm**
    - Téléchargez et installez Node.js depuis nodejs.org.
    - Vérifiez l'installation en exécutant les commandes suivantes dans votre terminal :

```
node -v
npm -v
```

3. **Installer Expo CLI**
    - Expo est un outil qui facilite le développement avec React Native.
    - Installez Expo CLI globalement en utilisant npm :

```
npm install -g expo-cli
```

4. **Créer un nouveau projet Expo**
    - Créez un nouveau projet en exécutant la commande suivante :

```
expo init MyFirstApp
```

    - Sélectionnez le template "blank" lorsque vous y êtes invité.
    - Accédez au répertoire du projet :

```
cd MyFirstApp
```

5. **Lancer l'application**

- Démarrez le serveur de développement Expo :

```
expo start
```

- Suivez les instructions pour ouvrir l'application sur un émulateur ou un appareil physique.

Vérification

Assurez-vous que l'application par défaut s'affiche correctement sur votre appareil ou émulateur.

Exercise 2 : Création d'une application de base

Objectif : Créer une application React Native simple qui affiche un message de bienvenue

Vérification : L'application doit afficher « Bienvenue dans votre première application React Native Au centre de l'écran

Temps estimé : 30 minutes

Corrigé :

Modifier le contenu de App.js

```
// filepath: App.js
import React from 'react';
import { Text, View, StyleSheet } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.welcome}>Bienvenue dans votre première
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  welcome: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  },
});
```

Exercise 3 : Utilisation des états et des props

Objectif : Apprendre à utiliser les états et les props dans une application React Native.

Vérifications : L'application doit afficher un compteur qui s'incrémente à chaque clic sur le bouton

Temps estimé : 30 minutes

Corrigé : Le fichier App.js doit contenir le code fourni ci-dessus

```
// filepath: App.js
import React, { useState } from 'react';
import { Text, View, Button, StyleSheet } from 'react-native';

export default function App() {
  const [count, setCount] = useState(0);

  return (
    <View style={styles.container}>
      <Text style={styles.counter}>Vous avez cliqué {count} fois
      <Button title="Cliquez-moi" onPress={() => setCount(count
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  counter: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  },
});
```

Exercise 4 : Création d'une application de base

Objectif : Créer une application To-Do List qui permet d'ajouter, afficher et supprimer des tâches. Cet exercice couvrira plusieurs concepts de React Native, y compris les états, les props, les entrées utilisateur, les listes, et le style.

Vérification :

- L'application doit permettre d'ajouter des tâches à une liste.
- Les tâches doivent s'afficher dans une liste.
- Les tâches doivent pouvoir être supprimées en cliquant dessus.

Temps estimé : 1 heure

Corrigé : Le fichier App.js doit contenir le code fourni ci-dessus

```jsx
// filepath: App.js
import React, { useState } from 'react';
import { Text, View, TextInput, Button, FlatList, TouchableOpacity, StyleSheet } from 'react-native';

export default function App() {
  const [task, setTask] = useState('');
  const [tasks, setTasks] = useState([]);

  const addTask = () => {
    if (task.length > 0) {
      setTasks([...tasks, { key: Math.random().toString(), value: task }]);
      setTask('');
    }
  };

  const removeTask = (taskKey) => {
    setTasks(tasks.filter(task => task.key !== taskKey));
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>To-Do List</Text>
      <TextInput
        style={styles.input}
        placeholder="Ajouter une tâche"
        onChangeText={text => setTask(text)}
        value={task}
      />
      <Button title="Ajouter" onPress={addTask} />
      <FlatList
        data={tasks}
        renderItem={({ item }) => (
          <TouchableOpacity onPress={() => removeTask(item.key)}>
            <View style={styles.listItem}>
              <Text>{item.value}</Text>
            </View>
          </TouchableOpacity>
        )}
      />
    </View>
  );
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: '#F5FCFF',
  },
  title: {
    fontSize: 24,
    textAlign: 'center',
    marginVertical: 20,
  },
  input: {
    height: 40,
    borderColor: 'gray',
    borderWidth: 1,
    marginBottom: 10,
    paddingHorizontal: 10,
  },
  listItem: {
    padding: 10,
    backgroundColor: '#f9c2ff',
    borderBottomWidth: 1,
    borderColor: '#ddd',
    marginVertical: 5,
  },
});
```