React + Supabase Cheat Sheet

Setup

```
import { createClient } from '@supabase/supabase-js'

const supabaseUrl = import.meta.env.VITE_SUPABASE_URL
const supabaseKey = import.meta.env.VITE_SUPABASE_KEY
const supabase = createClient(supabaseUrl, supabaseKey)
```

React State

```
const [student, setStudent] = useState([])
const [name, setName] = useState('')
const [major, setMajor] = useState('')
const [updating, setUpdating] = useState(false)
const [currentID, setcurrentID] = useState('')
```

Fetch Data

```
useEffect(() => {
  getData()
}, [])

const getData = async () => {
  const { data } = await supabase.from('student').select()
  setStudent(data)
}
```

Insert Data

```
const insertData = async () => {
  if (!name || !major) return window.alert('no name or major')
  const { error } = await supabase.from('student').insert([{ name, major }], { returning: "
  if (!error) {
    setName('')
    setMajor('')
    getData()
  }
}
```

Delete Data

```
const deleteData = async (id) => {
  const { error } = await supabase.from('student').delete().eq('id', id)
  if (!error) getData()
}
```

Update Data

```
const updateData = async (id) => {
  const { error } = await supabase.from('student').update([{ name, major }]).eq('id', id)
  if (!error) {
    setName('')
    setMajor('')
    setcurrentID('')
    setUpdating(false)
    getData()
  }
}
```

Prepare for Update

```
const preUpdateData = (id, name, major) => {
  setName(name)
  setMajor(major)
  setcurrentID(id)
  setUpdating(true)
}
```

UI Overview

```
- Two input fields: name, major
- Button toggles between Insert and Update
- List of students with:
  - ID
  - Name
  - Major
  - Delete button
  - Edit button (triggers update mode)
```