

## LAB 6

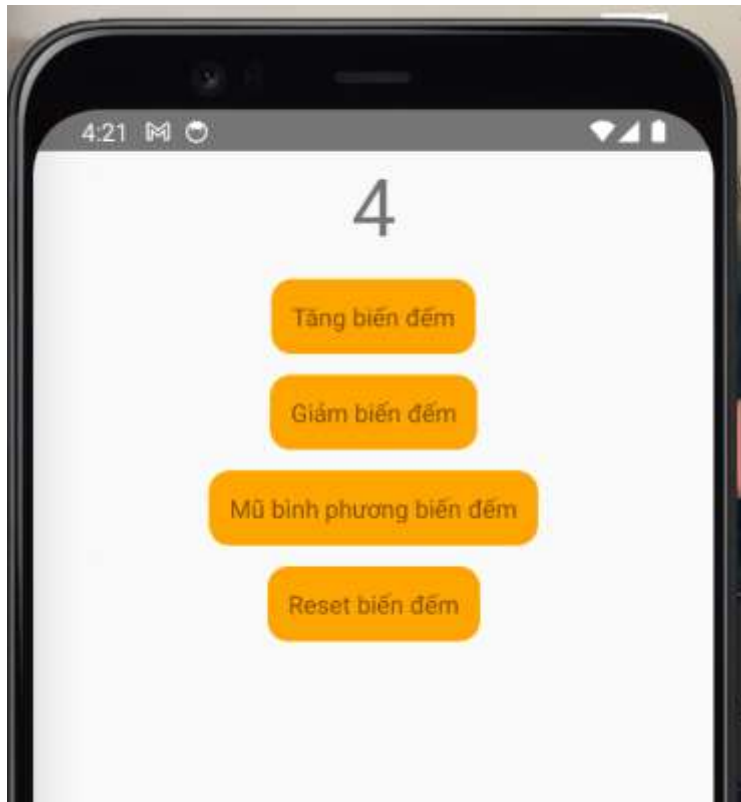
### MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Tạo reducer để quản lý dữ liệu trong store của redux
- ✓ Thực hiện dispatch các action với Redux.
- ✓ Thực hiện các cuộc gọi api từ các function của Redux Toolkit

### NỘI DUNG

#### **BÀI 1: TẠO CHƯƠNG TRÌNH BIẾN ĐẾM, LƯU TRỮ GIÁ TRỊ CỦA BIẾN ĐẾM VÀO STORE**



Yêu cầu:

- Phải lưu trữ giá trị state vào store của **Redux**.
- Sử dụng **createSlice** để tạo các function reducer để xử lý giá trị trong **reducer**

Hướng dẫn:

- Gọi hàm **createSlice** để tạo các reducer, hàm **multiply** dùng để nhân biến đếm, hàm **increment** để cộng biến đếm một giá trị, hàm **decrement** để giảm biến đếm đi một giá trị.
- Các bạn sử dụng **extraReducers** để nhận action được gửi lên, nếu action bằng **RESET\_COUNTER.type** thì reset lại tất cả biến đếm.

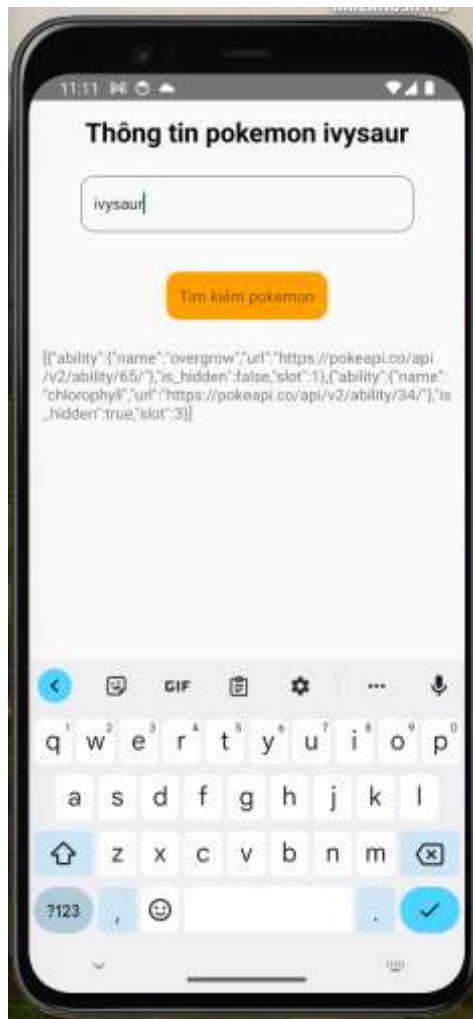
```
DaNenTang2 - counter.ts
1 export const counterSlice = createSlice({
2   name: 'counter',
3   initialState,
4   reducers: {
5     multiply: {
6       reducer: (state, action: PayloadAction<number>) => {
7         state.value = state.value * action.payload;
8       },
9       prepare: (value?: number) => ({payload: value || 2}),
10    },
11    increment: state => {
12      state.value += 1;
13    },
14    decrement: state => {
15      state.value -= 1;
16    },
17  },
18  extraReducers: builder => {
19    builder.addMatcher(
20      (action: AnyAction) => action.type === RESET_COUNTER.type,
21      () => {
22        return initialState;
23      },
24    );
25  },
26 });
```

- Lấy giá trị state biến đếm từ **store** bằng **useSelector**, sau đó hiển thị lên UI.
- Chuẩn bị các hàm để **dispatch** các **action** vào **reducer**

```
DaNenTang2 - HomeScreen.tsx
1  const counter = useSelector(state => state.counter);
2  const dispatch = useDispatch();
3
4  const onIncreaseCounter = () => dispatch(increment());
5  const onDecrementCounter = () => dispatch(decrement());
6  const onMultiplyCounter = () => dispatch(multiply(3));
7  const onResetCounter = () => dispatch(RESET_COUNTER());
```

- Cuối cùng thêm hàm, state vào UI để hoàn thiện bài tập

## **BÀI 2: GỌI LỆNH QUERY VỚI REDUX TOOLKIT**



Yêu cầu:

- Sử dụng **createApi** để tạo hàm gọi lệnh query trong **RTK**.

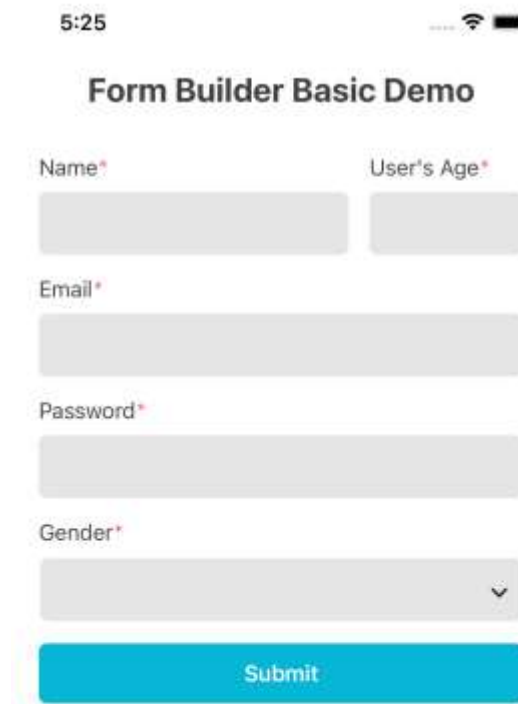
Hướng dẫn:

- Export hàm get dữ liệu thành hook **useLazy**, để gọi api khi nhấn nút tìm kiếm.

```
DaNenTang2 - pokemon.ts
1 export const pokemonApi = createApi({
2   reducerPath: 'pokemonApi',
3   baseQuery: fetchBaseQuery({
4     baseUrl: 'https://pokeapi.co/api/v2/',
5   }),
6   endpoints: builder => ({
7     getPokemonByName: builder.query<PokemonType[], string>({
8       query: name => `pokemon/${name}`,
9     }),
10  }),
11 });
12
13 export const {useLazyGetPokemonByNameQuery} = pokemonApi;
14
```

- Gọi hook ở UI để thực hiện lệnh query khi nhấn nút Tìm kiếm, nhập tên pokemon từ TextInput, thực hiện tạo UI giống như đề bài

### BÀI 3: GỬI DỮ LIỆU LÊN SERVER BẰNG RTK



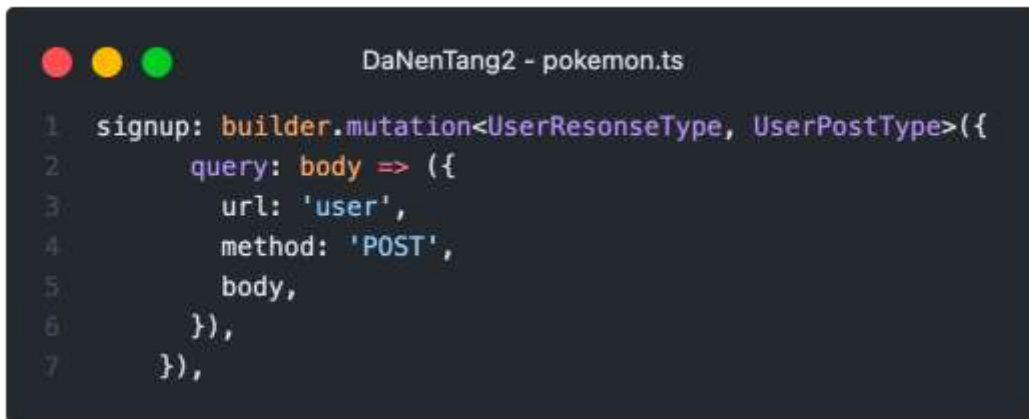
A mobile app interface titled "Form Builder Basic Demo". At the top, it shows the time "5:25" and status icons for signal, Wi-Fi, and battery. The form contains five fields: "Name\*" and "User's Age\*" (two small text input fields), "Email\*" (a medium text input field), "Password\*" (a medium text input field), and "Gender\*" (a dropdown menu with a downward arrow). Below these fields is a large blue "Submit" button.

Yêu cầu:

- Tạo form có giao diện giống hình.
- Sử dụng **createApi mutation** để gửi dữ liệu form nhập lên **server**.

Hướng dẫn:

- Sử dụng **mutation** tạo hàm **signup** để gửi dữ liệu lên server với phương thức POST.



```
DaNenTang2 - pokemon.ts
1  signup: builder.mutation<UserResonseType, UserPostType>({
2    query: body => ({
3      url: 'user',
4      method: 'POST',
5      body,
6    }),
7  }),
```

- Gọi hook **useSignupMutation** để thực hiện gọi dữ liệu lên server. Để bắt đầu gọi api gọi hàm **signup** và truyền dữ liệu body bạn muốn gửi lên server.

-



```
DaNenTang2 - PokemonScreen.tsx
1  const [signup, resultSignup] = useSignupMutation();
2
3  const onSignUp = () => {
4    signup({
5      name: 'bulbasaur',
6      age: 12,
7      email: 'vana@gmail.com',
8      password: '123123123',
9      gender: 'male',
10   });
11  };
```

#### BÀI 4: GV CHO THÊM

#### \*\*\* YÊU CẦU NỘP BÀI:

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết