**<CardTable />**

**<PlayingCardList />**

**const [cards, setCards] = useState([]);**

// response data from API is mapped to array ‘setCards’ with each given an id from UUID

**setCards**(cards => [...cards, { ...response.data, id: uuid() }]);

{cards.map(cardData => (

**<PlayingCard** key={cardData.id} front={cardData.cards[0].image} />

function **PlayingCard**({ front, back = backOfCard }) {

const [isFacingUp, setIsFacingUp] = useState(true);

**<PokeDex />**

function PokeDex() {

**const [pokemon, setPokemon] = useState([])**

// response data from API is mapped to array ‘setPokemon’ with each given an id from UUID

**setPokemon**(pokemon => [...pokemon, { ...response.data, id: uuid() }]);

**<PokemonSelect** add={addPokemon} />

function **PokemonSelect**({ add, pokemon = pokemonList }) {

**const [pokeIdx, setPokeIdx] = useState(0);**

**<PokemonCard**

key={cardData.id}

front={cardData.sprites.front\_default}

back={cardData.sprites.back\_default}

name={cardData.name}

stats={cardData.stats.map(stat => ({

value: stat.base\_stat,

name: stat.stat.name

}))}

function **PokemonCard**({ front, back, name, stats }) {

**const [isFacingUp, setIsFacingUp] = useState(true);**