

```

import { useState, useEffect } from "react";
import {
  Authenticator,
  Button,
  Text,
  TextField,
  Heading,
  Flex,
  View,
  Grid,
  Divider,
} from "@aws-amplify/ui-react";
import { Amplify } from "aws-amplify";
import "@aws-amplify/ui-react/styles.css";
import { generateClient } from "aws-amplify/data";
import outputs from "../amplify_outputs.json";
/**
 * @type {import('aws-amplify/data').Client<import('../amplify/data/
resource').Schema>}
 */

Amplify.configure(outputs);
const client = generateClient({
  authMode: "userPool",
});

export default function App() {
  const [expenses, setExpenses] = useState([]);

  useEffect(() => {
    client.models.Expense.observeQuery().subscribe({
      next: (data) => setExpenses([...data.items]),
    });
  }, []);

  async function createExpense(event) {
    event.preventDefault();
    const form = new FormData(event.target);

    await client.models.Expense.create({
      name: form.get("name"),
      amount: form.get("amount"),
    });

    event.target.reset();
  }

  async function deleteExpense({ id }) {
    const toBeDeletedExpense = {
      id,
    };
  }

```

```

};

await client.models.Expense.delete(toBeDeletedExpense);
}

return (
  <Authenticator>
    {{{ signOut }}} => (
      <Flex
        className="App"
        justifyContent="center"
        alignItems="center"
        direction="column"
        width="70%"
        margin="0 auto"
      >
        <Heading level={1}>Expense Tracker</Heading>
        <View as="form" margin="3rem 0" onSubmit={createExpense}>
          <Flex
            direction="column"
            justifyContent="center"
            gap="2rem"
            padding="2rem"
          >
            <TextField
              name="name"
              placeholder="Expense Name"
              label="Expense Name"
              labelHidden
              variation="quiet"
              required
            />
            <TextField
              name="amount"
              placeholder="Expense Amount"
              label="Expense Amount"
              type="float"
              labelHidden
              variation="quiet"
              required
            />

            <Button type="submit" variation="primary">
              Create Expense
            </Button>
          </Flex>
        </View>
        <Divider />
        <Heading level={2}>Expenses</Heading>
        <Grid

```

```

        margin="3rem 0"
        autoFlow="column"
        justifyContent="center"
        gap="2rem"
        alignContent="center"
    >
    {expenses.map((expense) => (
        <Flex
            key={expense.id || expense.name}
            direction="column"
            justifyContent="center"
            alignItems="center"
            gap="2rem"
            border="1px solid #ccc"
            padding="2rem"
            borderRadius="5%"
            className="box"
        >
            <View>
                <Heading level="3">{expense.name}</Heading>
            </View>
            <Text fontStyle="italic">${expense.amount}</Text>

            <Button
                variation="destructive"
                onClick={() => deleteExpense(expense)}
            >
                Delete note
            </Button>
        </Flex>
    ))}
    </Grid>
    <Button onClick={signOut}>Sign Out</Button>
    </Flex>
    )}
    </Authenticator>
    );
}

```