

ExpensesTrip Component - React.js

This component is part of a CRM client project and is designed to manage and display trip-related expenses such as travel, accommodation, meals, and other categories.

Key Features:

1. Add new expense entries dynamically.
2. Display expenses in a table with status indicators.
3. Manage state using React useState hook.
4. Styled with modern CSS for a clean dashboard look.

File: ExpensesTrip.js

```
import React, { useState } from "react";
import "./ExpensesTrip.css";

const ExpensesTrip = () => {
  const [expenses, setExpenses] = useState([
    { id: 1, date: "2024-11-01", category: "Travel", description: "Flight to New York", amount: 450, status: "Approved" },
    { id: 2, date: "2024-11-02", category: "Accommodation", description: "Hotel - Marriott NYC", amount: 250, status: "Pending" },
    { id: 3, date: "2024-11-03", category: "Meals", description: "Client Dinner", amount: 75, status: "Approved" }
  ]);

  const [newExpense, setNewExpense] = useState({ date: "", category: "", description: "", amount: "" });

  const handleChange = (e) => {
    setNewExpense({ ...newExpense, [e.target.name]: e.target.value });
  };

  return (
    <div>
      <table>
        <thead>
          <tr>
            <th>Date</th>
            <th>Category</th>
            <th>Description</th>
            <th>Amount</th>
            <th>Status</th>
          </tr>
        </thead>
        <tbody>
          {expenses.map((expense) => (
            <tr key={expense.id}>
              <td>{expense.date}</td>
              <td>{expense.category}</td>
              <td>{expense.description}</td>
              <td>{expense.amount}</td>
              <td>{expense.status}</td>
            </tr>
          ))}
        </tbody>
      </table>
      <button>Add Expense</button>
    </div>
  );
}
```

```
};
```

```
const handleAddExpense = (e) => {
  e.preventDefault();
  if (!newExpense.date || !newExpense.category || !newExpense.description || !newExpense.amount) {
    alert("Please fill all fields");
    return;
  }
  setExpenses([...expenses, { id: expenses.length + 1, ...newExpense, status: "Pending" }]);
  setNewExpense({ date: "", category: "", description: "", amount: "" });
};

return (
  <div className="expenses-container">
    <h2>Trip Expenses</h2>
    <form className="expense-form" onSubmit={handleAddExpense}>
      <input type="date" name="date" value={newExpense.date} onChange={handleChange} required />
      <select name="category" value={newExpense.category} onChange={handleChange} required>
        <option value="">Select Category</option>
        <option value="Travel">Travel</option>
        <option value="Accommodation">Accommodation</option>
        <option value="Meals">Meals</option>
        <option value="Other">Other</option>
      </select>
      <input type="text" name="description" placeholder="Description" value={newExpense.description} onChange={handleChange} required />
      <input type="number" name="amount" placeholder="Amount ($)" value={newExpense.amount} onChange={handleChange} required />
      <button type="submit">Add Expense</button>
    </form>
    <table className="expense-table">
```

```

<thead>
  <tr>
    <th>Date</th><th>Category</th><th>Description</th><th>Amount ($)</th><th>Status</th>
  </tr>
</thead>
<tbody>
  {expenses.map((exp) => (
    <tr key={exp.id}>

      <td>{exp.date}</td><td>{exp.category}</td><td>{exp.description}</td><td>{exp.amount}</td>
      <td><span className={exp.status === "Approved" ? "status approved" : "status pending"}>{exp.status}</span></td>
    </tr>
  )))
  </tbody>
</table>
</div>
);
};

}


```

export default ExpensesTrip;

CSS (ExpensesTrip.css)

```

.expenses-container { width: 90%; margin: 20px auto; background: #f9fafb; padding: 20px; border-radius: 12px; box-shadow: 0px 2px 8px rgba(0,0,0,0.1); }
.expenses-container h2 { text-align: center; color: #333; margin-bottom: 15px; }
.expense-form { display: flex; gap: 10px; flex-wrap: wrap; margin-bottom: 20px; justify-content: center; }
.expense-form input, .expense-form select { padding: 8px; border-radius: 6px; border: 1px solid #ccc; min-width: 150px; }

```

```
.expense-form button { background-color: #2563eb; color: white; border: none; padding: 8px 16px; border-radius: 6px; cursor: pointer; }

.expense-form button:hover { background-color: #1d4ed8; }

.expense-table { width: 100%; border-collapse: collapse; margin-top: 10px; }

.expense-table th, .expense-table td { padding: 10px; border-bottom: 1px solid #ddd; text-align: center; }

.status.approved { color: green; font-weight: bold; }

.status.pending { color: orange; font-weight: bold; }
```