

Expense Listing:

```
import { Button, Modal } from "antd";
import React, { useState, useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import { async_get_category } from "../action/categoryAction";

const ModalExpense = (props) => {
  const {
    buttonName,
    addExpenseOrUpdate,
    expenseErrorHandleReset,
    price: amount,
    categoryId: categorySlNo,
    itemName: productName,
  } = props;
  const [categoryId, setCategoryId] = useState(
    categorySlNo ? categorySlNo : ""
  );
  const [itemName, setItemName] = useState(productName ? productName : "");
  const [price, setPrice] = useState(amount ? amount : 0);

  const [isModalOpen, setIsModalOpen] = useState(false);

  const dispatch = useDispatch();

  const categoryStore = useSelector((state) => {
    return state.category.category;
  });

  const expenseStore = useSelector((state) => {
    return state.expense;
  });

  useEffect(() => {
    dispatch(async_get_category());
  }, [dispatch]);

  const changeHandle = (e) => {
    if (e.target.name === "categoryOption") {
      setCategoryId(e.target.value);
    } else if (e.target.name === "itemName") {
      setItemName(e.target.value);
    } else if ((e.target.name = "price")) {
      setPrice(e.target.value);
    }
  };
};
```

```

const showModal = () => {
  setIsModalOpen(true);
};

const handleOk = () => {
  const formData = {
    itemName: itemName,
    amount: price,
    categoryId: categoryId,
  };

  addExpenseOrUpdate(formData);
};

// useEffect(() => {
//   setCategoryId("");
//   setItemName("");
//   setPrice(0);
//   setIsModalOpen(false);
// }, [expenseStore.expense]);

const handleCancel = () => {
  // setCategoryId("");
  // setItemName("");
  // setPrice(0);

  expenseErrorHandleReset("");
  setIsModalOpen(false);
};

return (
  <>
    <Button type="primary" onClick={showModal}>
      {buttonName}
    </Button>
    <Modal
      title="Basic Modal"
      open={isModalOpen}
      onOk={handleOk}
      onCancel={handleCancel}
    >
      <form className="modal-form" onSubmit={handleOk}>
        <div>
          <label>
            <b>Item Name</b>
          </label>
          <br />
          <input
            type="text"
            value={itemName}

```

```

        onChange={changeHandle}
        name="itemName"
        className="add-expense-form"
      />
    <br />
    <br />
    <select
      value={categoryId}
      onChange={changeHandle}
      name="categoryOption"
      className="add-expense-dropdown"
    >
      <option value="">Select Item</option>
      {categoryStore.length !== 0 &&
        categoryStore.map((ele, i) => {
          return (
            <option value={ele._id} key={i}>
              {ele.categoryName}
            </option>
          );
        })}
    </select>
  </div>
  <div>
    <label>
      <b>Price</b>
    </label>
    <br />
    <input
      text="Number"
      value={price}
      onChange={changeHandle}
      name="price"
      className="add-expense-form"
    />
  </div>
</form>
<b style={{ color: "red" }}>
  {expenseStore.errors && expenseStore.errors}
</b>
</Modal>
</>
);
};

export default ModalExpense;

```

Expense Add And Search Component:

```
import React from "react";
//import { Button, Modal } from "antd";
import { useDispatch } from "react-redux";
//import { async_get_category } from "../action/categoryAction";
import { async_expense_create } from "../action/expenseAction";
import { errors_expense } from "../action/expenseAction";
import ModalExpense from "../ModalExpense";

const ExpenseAddAndSearchComponent = () => {
  const dispatch = useDispatch();
  // const categoryStore = useSelector((state) => {
  //   return state.category.category;
  // });

  // const expenseStore = useSelector((state) => {
  //   return state.expense;
  // });

  // const [categoryId, setCategoryId] = useState("");
  // const [itemName, setItemName] = useState("");
  // const [price, setPrice] = useState(0);

  // useEffect(() => {
  //   dispatch(async_get_category());
  // }, [dispatch]);

  //const [isModalOpen, setIsModalOpen] = useState(false);
  // const showModal = () => {
  //   setIsModalOpen(true);
  // };
  // const handleOk = () => {
  //   const formData = {
  //     itemName: itemName,
  //     amount: price,
  //     categoryId: categoryId,
  //   };
  //   dispatch(async_expense_create(formData));
  // };

  const addExpenseOrUpdate = (formData) => {
    dispatch(async_expense_create(formData));
  };

  const expenseErrorHandleReset = (string) => {
    dispatch(errors_expense(""));
  };
};
```

```

// useEffect(() => {
//   setCategoryId("");
//   setItemName("");
//   setPrice(0);
//   setIsModalOpen(false);
// }, [expenseStore.expense]);

// const handleCancel = () => {
//   setCategoryId("");
//   setItemName("");
//   setPrice(0);
//   dispatch(errors_expense(""));
//   setIsModalOpen(false);
// };

// const changeHandle = (e) => {
//   if (e.target.name === "categoryOption") {
//     setCategoryId(e.target.value);
//   } else if (e.target.name === "itemName") {
//     setItemName(e.target.value);
//   } else if ((e.target.name = "price")) {
//     setPrice(e.target.value);
//   }
// };

return (
  <>
    { /* <Button type="primary" onClick={showModal}>
      Add Expense
    </Button>

    <Modal
      title="Add Expenses"
      open={isModalOpen}
      onOk={handleOk}
      onCancel={handleCancel}
    >
      <form className="modal-form" onSubmit={handleOk}>
        <div>
          <label>
            <b>Item Name</b>
          </label>
          <br />
          <input
            type="text"
            value={itemName}
            onChange={changeHandle}

```

```

        name="itemName"
        className="add-expense-form"
      />
    <br />
    <br />
    <select
      value={categoryId}
      onChange={changeHandle}
      name="categoryOption"
      className="add-expense-dropdown"
    >
      <option value="">Select Item</option>
      {categoryStore.length !== 0 &&
        categoryStore.map((ele, i) => {
          return (
            <option value={ele._id} key={i}>
              {ele.categoryName}
            </option>
          );
        })}
    </select>
  </div>
  <div>
    <label>
      <b>Price</b>
    </label>
    <br />
    <input
      text="Number"
      value={price}
      onChange={changeHandle}
      name="price"
      className="add-expense-form"
    />
  </div>
</form>
<b style={{ color: "red" }}>
  {expenseStore.errors && expenseStore.errors}
</b>
</Modal>*/}
<div>
  <ModalExpense
    buttonName={"Add Expense"}
    addExpenseOrUpdate={addExpenseOrUpdate}
    expenseErrorHandleReset={expenseErrorHandleReset}
  />
</div>
<div>

```

```

        <input placeholder="search expense..." />
      </div>
    </>
  );
};

export default ExpenseAddAndSearchComponent;

```

Modal:

```

import { Button, Modal } from "antd";
import React, { useState, useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import { async_get_category } from "../action/categoryAction";

const ModalExpense = (props) => {
  const {
    buttonName,
    addExpenseOrUpdate,
    expenseErrorHandleReset,
    price: amount,
    categoryId: categorySlNo,
    itemName: productName,
  } = props;
  const [categoryId, setCategoryId] = useState(
    categorySlNo ? categorySlNo : ""
  );
  const [itemName, setItemName] = useState(productName ? productName : "");
  const [price, setPrice] = useState(amount ? amount : 0);

  const [isModalOpen, setIsModalOpen] = useState(false);

  const dispatch = useDispatch();

  const categoryStore = useSelector((state) => {
    return state.category.category;
  });

  const expenseStore = useSelector((state) => {
    return state.expense;
  });

  useEffect(() => {
    dispatch(async_get_category());
  }, [dispatch]);

```

```

const changeHandle = (e) => {
  if (e.target.name === "categoryOption") {
    setCategoryId(e.target.value);
  } else if (e.target.name === "itemName") {
    setItemName(e.target.value);
  } else if ((e.target.name = "price")) {
    setPrice(e.target.value);
  }
};

const showModal = () => {
  setIsModalOpen(true);
};

const handleOk = () => {
  const formData = {
    itemName: itemName,
    amount: price,
    categoryId: categoryId,
  };

  addExpenseOrUpdate(formData);
};

// useEffect(() => {
//   setCategoryId("");
//   setItemName("");
//   setPrice(0);
//   setIsModalOpen(false);
// }, [expenseStore.expense]);

const handleCancel = () => {
  // setCategoryId("");
  // setItemName("");
  // setPrice(0);

  expenseErrorHandleReset("");
  setIsModalOpen(false);
};

return (
  <>
    <Button type="primary" onClick={showModal}>
      {buttonName}
    </Button>
    <Modal
      title="Basic Modal"
      open={isModalOpen}
      onOk={handleOk}
      onCancel={handleCancel}
    >

```



```

>
<form className="modal-form" onSubmit={handleOk}>
  <div>
    <label>
      <b>Item Name</b>
    </label>
    <br />
    <input
      type="text"
      value={itemName}
      onChange={changeHandle}
      name="itemName"
      className="add-expense-form"
    />
    <br />
    <br />
    <select
      value={categoryId}
      onChange={changeHandle}
      name="categoryOption"
      className="add-expense-dropdown"
    >
      <option value="">Select Item</option>
      {categoryStore.length !== 0 &&
        categoryStore.map((ele, i) => {
          return (
            <option value={ele._id} key={i}>
              {ele.categoryName}
            </option>
          );
        })}
    </select>
  </div>
  <div>
    <label>
      <b>Price</b>
    </label>
    <br />
    <input
      type="Number"
      value={price}
      onChange={changeHandle}
      name="price"
      className="add-expense-form"
    />
  </div>
</form>
<b style={{ color: "red" }}>

```

```
        {expenseStore.errors && expenseStore.errors}  
      </b>  
    </Modal>  
  </>  
);  
};  
  
export default ModalExpense;
```