

```

import React, { useState, useEffect } from "react";
import Card from "../Card/Card";
import RuleModal from "../Modal/RuleModal";
import { reportTypes } from "../data/mockData";
import { FiPlus, FiEdit2, FiTrash2, FiUploadCloud } from "react-icons/fi";
import { useOutletContext } from "react-router-dom";
import axios from "axios";

const ReportsPage = () => {
  const [setNotification] = useOutletContext();

  const [rules, setRules] = useState([]); // from backend
  const [activeTab, setActiveTab] = useState("config");
  const [currentRule, setCurrentRule] = useState({
    id: null,
    fileName: "",
    fileStoragePath: "",
    uploadedBy: null,
  });
  const [showRuleModal, setShowRuleModal] = useState(false);

  const [file, setFile] = useState(null);
  const [selectedReportType, setSelectedReportType] = useState(reportTypes[0]);
  const [isUploading, setIsUploading] = useState(false);

  //  Fetch all reports from backend
  useEffect(() => {
    axios
      .get("http://localhost:8080/api/reports")
      .then((res) => setRules(res.data))
      .catch(() =>
        setNotification({
          type: "error",
          message: "Failed to load reports",
        })
      );
  }, []);

  const handleAddRule = () => {
    setCurrentRule({ id: null, fileName: "", fileStoragePath: "", uploadedBy: null });
    setShowRuleModal(true);
  };

  const handleEditRule = (rule) => {
    setCurrentRule(rule);
    setShowRuleModal(true);
  };

```

```

const handleDeleteRule = (id) => {
  if (window.confirm("Are you sure you want to delete this report?")) {
    axios
      .delete(`http://localhost:8080/api/reports/${id}`)
      .then(() => {
        setRules((prev) => prev.filter((r) => r.reportId !== id));
        setNotification({
          type: "success",
          message: "Report deleted successfully!",
        });
      })
      .catch(() =>
        setNotification({
          type: "error",
          message: "Failed to delete report",
        })
      );
  }
};

```

```

const handleSaveRule = () => {
  if (currentRule.reportId) {
    // update existing
    axios
      .put(`http://localhost:8080/api/reports/${currentRule.reportId}`, currentRule)
      .then((res) => {
        setRules((prev) =>
          prev.map((r) => (r.reportId === currentRule.reportId ? res.data : r))
        );
        setNotification({
          type: "success",
          message: "Report updated successfully!",
        });
        setShowRuleModal(false);
      })
      .catch(() =>
        setNotification({
          type: "error",
          message: "Failed to update report",
        })
      );
  } else {
    // add new
    axios
      .post("http://localhost:8080/api/reports", currentRule)
      .then((res) => {
        setRules((prev) => [res.data, ...prev]);
        setNotification({

```

```

        type: "success",
        message: "Report added successfully!",
    });
    setShowRuleModal(false);
  })
  .catch(() =>
    setNotification({
      type: "error",
      message: "Failed to add report",
    })
  );
}
};

```

```

const handleFileChange = (e) => {
  if (e.target.files.length > 0) {
    setFile(e.target.files[0]);
  }
};

```

```

const handleUpload = () => {
  if (!file) {
    setNotification({
      type: "error",
      message: "Please select a file first",
    });
    return;
  }
}

```

```

setIsUploading(true);
setTimeout(() => {
  setIsUploading(false);
  setNotification({
    type: "success",
    message: `File '${file.name}' uploaded successfully!`,
  });
  setFile(null);
}, 1500);
};

```

```

const cardHeader = (
  <ul className="nav nav-tabs card-header-tabs">
    <li className="nav-item">
      <a
        className={`nav-link ${activeTab === "config" ? "active" : ""}`}
        href="#"
        onClick={() => setActiveTab("config")}
        style={{ color: "black" }}
      >

```

```

    >
      Automated Path Config
    </a>
  </li>
  <li className="nav-item">
    <a
      className={`nav-link ${activeTab === "upload" ? "active" : ""}`}
      href="#"
      onClick={() => setActiveTab("upload")}
      style={{ color: "black" }}
    >
      Manual Upload
    </a>
  </li>
</ul>
);

return (
  <div className="page-content">
    <Card headerContent={cardHeader}>
      {activeTab === "config" && (
        <div>
          <button className="btn btn-primary mb-3" onClick={handleAddRule}>
            <FiPlus className="me-1" />
            Add New Report
          </button>
          <div className="table-responsive">
            <table className="table table-hover align-middle">
              <thead>
                <tr>
                  <th>Id</th>
                  <th>Report Name</th>
                  <th>Source Path</th>
                  <th className="text-end">Uploaded By</th>
                  <th className="text-end">Actions</th>
                </tr>
              </thead>
              <tbody>
                {rules.map((rule) => (
                  <tr key={rule.reportId}>
                    <td>{rule.reportId}</td>
                    <td>{rule.fileName}</td>
                    <td>
                      <code>{rule.fileStoragePath}</code>
                    </td>
                    <td className="text-end">{rule.uploadedBy}</td>
                    <td className="text-end">
                      <button

```

```

        className="btn btn-sm btn-outline-secondary me-2"
        onClick={() => handleEditRule(rule)}
      >
        <FiEdit2 />
      </button>
      <button
        className="btn btn-sm btn-outline-danger"
        onClick={() => handleDeleteRule(rule.reportId)}
      >
        <FiTrash2 />
      </button>
    </td>
  </tr>
</tbody>
</table>
</div>
</div>
)}

```

```

{activeTab === "upload" && (
  <div>
    <p className="text-muted">
      This tool is for uploading individual reports for exceptional
      cases.
    </p>

    <div className="mb-3">
      <label className="form-label">Step 1: Select Report Type</label>
      <select
        className="form-select"
        value={selectedReportType}
        onChange={(e) => setSelectedReportType(e.target.value)}
        style={{ width: "30%" }}
      >
        {reportTypes.map((type) => (
          <option key={type} value={type}>
            {type}
          </option>
        ))}
      </select>
    </div>

    <div className="mb-3">
      <label className="form-label">Step 2: Choose File</label>
      <div className="input-group" style={{ width: "30%" }}>
        <input
          type="file"

```

```

        className="form-control"
        onChange={handleFileChange}
      />
    </div>
    {file && (
      <small className="form-text text-muted mt-1">
        Selected: {file.name}
      </small>
    )}
  </div>

  <div className="mt-4">
    <button
      className="btn btn-success"
      onClick={handleUpload}
      disabled={isUploading}
    >
      {isUploading ? (
        <>
          <span
            className="spinner-border spinner-border-sm me-2"
            role="status"
            aria-hidden="true"
          ></span>
          Uploading...
        </>
      ) : (
        <>
          <FiUploadCloud className="me-1" /> Upload Now
        </>
      )}
    </button>
  </div>
</div>
)}
</Card>

<RuleModal
  show={showRuleModal}
  onHide={() => setShowRuleModal(false)}
  onSave={handleSaveRule}
  rule={currentRule}
  setRule={setCurrentRule}
/>
</div>
);
};

```

```
export default ReportsPage;
```