

#Submitted.jsx

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import Cookies from 'js-cookie';
import { TextField, Button, IconButton, InputAdornment, Grid, Paper, CircularProgress, Alert } from '@mui/material';
import { createTheme, ThemeProvider } from '@mui/material/styles';
import SearchIcon from '@mui/icons-material/Search';
import ClearIcon from '@mui/icons-material/Clear';
import ReactPaginate from 'react-paginate';
import Accordion from './Accordion';
import './css/Submitted.css'; // Import the new CSS file for styles
```

```
const theme = createTheme({
  palette: {
    primary: {
      main: '#ff4081',
    },
    secondary: {
      main: '#7c4dff',
    },
  },
  typography: {
    fontFamily: 'Montserrat, sans-serif',
  },
  shape: {
    borderRadius: 8,
  },
  components: {
    MuiButton: {
      styleOverrides: {
        root: {
          textTransform: 'none',
          transition: 'all 0.3s',
          '&:hover': {
            boxShadow: '0 4px 8px rgba(124, 77, 255, 0.2)',
          },
        },
      },
    },
    MuiTextField: {
      styleOverrides: {
        root: {
          transition: 'transform 0.3s, box-shadow 0.3s',
          '&:hover': {
            transform: 'scale(1.05)',
            boxShadow: '0 4px 8px rgba(255, 64, 129, 0.2)',
          },
        },
      },
    },
  },
});
```

```
const Submitted = () => {
  const [audits, setAudits] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const [searchTerm, setSearchTerm] = useState('');
  const [startDate, setStartDate] = useState('');
```

```

const [endDate, setEndDate] = useState("");
const [currentPage, setCurrentPage] = useState(0);
const auditsPerPage = 10;

useEffect(() => {
  const accessToken = Cookies.get('access_token');

  if (accessToken) {
    axios.get('http://127.0.0.1:8000/api/auth/user-audits/', {
      headers: {
        "Content-Type": "application/json",
        Authorization: `Bearer ${accessToken}`,
      }
    })
    .then(response => {
      setAudits(response.data);
      setLoading(false);
    })
    .catch(err => {
      setError('Failed to fetch audits');
      setLoading(false);
    });
  } else {
    setError('No access token found');
    setLoading(false);
  }
}, []);

const validateDate = (date) => {
  return !isNaN(Date.parse(date));
};

const filteredAudits = audits.filter(audit => {
  const auditDate = new Date(audit.created_at);
  const isWithinDateRange =
    (!startDate || (validateDate(startDate) && auditDate >= new Date(startDate))) &&
    (!endDate || (validateDate(endDate) && auditDate <= new Date(endDate)));
  const matchesSearchTerm =
    !searchTerm || (audit.barcode && audit.barcode.toLowerCase().includes(searchTerm.toLowerCase()));

  return isWithinDateRange && matchesSearchTerm;
});

const handleClear = () => {
  setSearchTerm("");
  setStartDate("");
  setEndDate("");
};

const handleShowAll = () => {
  setSearchTerm("");
  setStartDate("");
  setEndDate("");
  setCurrentPage(0); // Reset to the first page
};

const handlePageClick = ({ selected }) => {
  setCurrentPage(selected);
};

```

```
const offset = currentPage * auditsPerPage;
const currentPageData = filteredAudits.slice(offset, offset + auditsPerPage);
const pageCount = Math.ceil(filteredAudits.length / auditsPerPage);
```

```
return (
  <ThemeProvider theme={theme}>
    <div className="wrapper" sx={{overflow:'hidden !important'}}>
      <div className="headtitle">
        <div className="audit-count">
          Showing {currentPageData.length} out of {filteredAudits.length} audits
        </div>
        <h1>Submitted</h1>
        <form className="searchBar" id="searchBar">
          <Grid container spacing={2} justifyContent="center">
            <Grid item>
              <TextField
                variant="outlined"
                placeholder="Search Cases..."
                className="admin-search"
                value={searchTerm}
                onChange={e => setSearchTerm(e.target.value)}
                InputProps={{
                  endAdornment: (
                    <InputAdornment position="end">
                      <SearchIcon />
                    </InputAdornment>
                  ),
                }}
              />
            </Grid>
            <Grid item>
              <TextField
                variant="outlined"
                placeholder="Start Date"
                type="date"
                className="admin-search2"
                value={startDate}
                onChange={e => setStartDate(e.target.value)}
                InputLabelProps={{ shrink: true }}
              />
            </Grid>
            <Grid item>
              <TextField
                variant="outlined"
                placeholder="End Date"
                type="date"
                className="admin-search2"
                value={endDate}
                onChange={e => setEndDate(e.target.value)}
                InputLabelProps={{ shrink: true }}
              />
            </Grid>
            <Grid item>
              <Button
                variant="outlined"
                color="primary"
                className="admin-searchBtn"
                // startIcon={<SearchIcon />}
                style={{ borderRadius: '0.5rem' }}
              >
```

```

        Search
      </Button>
    </Grid>
    <Grid item>
      <Button
        variant="outlined"
        color="secondary"
        className="admin-clearBtn"
        // startIcon={<ClearIcon />}
        style={{ borderRadius: '0.5rem' }}
        onClick={handleClear}
      >
        Clear
      </Button>
    </Grid>
    <Grid item>
      <Button
        variant="outlined"
        color="secondary"
        className="admin-showAllBtn"
        style={{ borderRadius: '0.5rem' }}
        onClick={handleShowAll}
      >
        Show All
      </Button>
    </Grid>
  </Grid>
</form>
<div className="underline"></div>
</div>

<div className="reverse">
  {loading ? (
    <CircularProgress />
  ) : error ? (
    <Alert severity="error">{error}</Alert>
  ) : (
    currentPageData.map(audit => (
      <Accordion key={audit.barcode} audit={audit} />
    ))
  )}
</div>

<ReactPaginate
  previousLabel={'Previous'}
  nextLabel={'Next'}
  breakLabel={'...'}
  breakClassName={'break-me'}
  pageCount={pageCount}
  marginPagesDisplayed={2}
  pageRangeDisplayed={5}
  onPageChange={handlePageClick}
  containerClassName={'pagination'}
  subContainerClassName={'pages pagination'}
  activeClassName={'active'}
/>
</div>
</ThemeProvider>
);
};

```

export default Submitted;

#Submitted.css

@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@400;600&display=swap');

```
.wrapper {  
  font-family: 'Montserrat', sans-serif;  
  padding: 20px;  
  background: linear-gradient(to right, #f7f8fc, #ffe0e0);  
  animation: fadeIn 1s ease-in;  
}
```

```
.headtitle {  
  text-align: center;  
  margin-bottom: 20px;  
  animation: slideInDown 0.5s ease-out;  
}
```

```
.headtitle h1 {  
  font-size: 2.5em;  
  color: #007BFF;  
}
```

```
.searchBar {  
  display: flex;  
  justify-content: center;  
  gap: 20px;  
  margin-bottom: 20px;  
}
```

```
.admin-search,  
.admin-search2 {  
  width: 200px;  
}
```

```
.admin-searchBtn,  
.admin-clearBtn,  
.admin-showAllBtn {  
  padding: 10px 20px;  
  border-radius: 8px;  
  cursor: pointer;  
  transition: background-color 0.3s, box-shadow 0.3s;  
}
```

```
.admin-searchBtn:hover,  
.admin-clearBtn:hover,  
.admin-showAllBtn:hover {  
  box-shadow: 0 4px 8px rgba(124, 77, 255, 0.2);  
}
```

```
.underline {  
  width: 100%;  
  height: 4px;  
  background-color: #007BFF;  
  margin: 10px 0;  
  animation: expand 0.5s ease-out;  
}
```

```
.reverse {  
  display: flex;  
  flex-direction: column;  
  gap: 10px;  
  animation: fadeInUp 0.7s ease-in;  
}
```

```
/* Pagination Styles */
```

```
.pagination {  
  display: flex;  
  justify-content: center;  
  padding: 20px 0;  
}
```

```
.pagination .active a {  
  background-color: #007BFF;  
  color: white;  
  border: 1px solid #007BFF;  
}
```

```
.pagination a {  
  color: #007BFF;  
  border: 1px solid #007BFF;  
  padding: 8px 16px;  
  margin: 0 4px;  
  text-decoration: none;  
  transition: background-color 0.3s;  
}
```

```
.pagination a:hover {  
  background-color: #007BFF;  
}
```

```
ul {  
  list-style-type: none;  
}
```

```
@keyframes fadeIn {  
  from {  
    opacity: 0;  
  }  
  to {  
    opacity: 1;  
  }  
}
```

```
@keyframes slideInDown {  
  from {  
    transform: translateY(-20px);  
    opacity: 0;  
  }  
  to {  
    transform: translateY(0);  
    opacity: 1;  
  }  
}
```

```
@keyframes expand {
```

```

from {
  width: 0;
}
to {
  width: 100%;
}
}

```

```

@keyframes fadeInUp {
  from {
    transform: translateY(20px);
    opacity: 0;
  }
  to {
    transform: translateY(0);
    opacity: 1;
  }
}

```

```

.searchBar input {
  padding: 10px 16px !important;
  border: 0px !important;
  border-radius: 4px;
}

```

```

.css-1laqs7-MuiInputAdornment-root {

  padding: 0px 10px !important;

}

```

```

.searchBar button {

  margin-top: 2px !important;

  padding: 8px 20px !important;

}

```

#Accordion.jsx

```

import React, { useState } from 'react';
import { Paper, Typography, Collapse, Table, TableBody, TableCell, TableContainer, TableHead, TableRow, IconButton } from '@mui/material';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';
import '../css/Accordion.css';

```

```

const Accordion = ({ audit }) => {
  const [isActive, setIsActive] = useState(false);

```

```

  const toggleAccordion = () => {
    setIsActive(!isActive);
  };

```

```

  return (
    <Paper className={`accordion ${isActive ? 'active' : ''}`} elevation={3}>
      <div className="accordion-summary" onClick={toggleAccordion}>
        <div className="data">
          <div>
            <strong>Audit Type:</strong>
            <p className="audit_type">{audit?.audit_type || 'None'}</p>
          </div>
        </div>
      </div>
    </Paper>
  );
}

```

```

</div>
<div>
  <strong>Barcode:</strong>
  <p className="bars">{audit?.barcode || 'None'}</p>
</div>
<div className="bigDet">
  <strong>Status:</strong>
  <p>{audit?.case_status || 'None'}</p>
</div>
<div className="bigDet">
  <strong>Submitted Date:</strong>
  <p className="date">{audit?.created_at || 'None'}</p>
</div>
</div>
<IconButton onClick={toggleAccordion} className="expand-icon">
  <ExpandMoreIcon />
</IconButton>
</div>
<Collapse in={isActive} timeout="auto" unmountOnExit>
  <div className="accordion-details">
    {/* Case Errors Table */}
    <Typography variant="h6" className="table-title">Case Errors</Typography>
    <TableContainer>
      <Table>
        <TableHead>
          <TableRow>
            <TableCell>Category</TableCell>
            <TableCell>Description</TableCell>
            <TableCell>Final Error Category</TableCell>
            <TableCell>Remarks for Audit</TableCell>
            <TableCell>Remarks for Ops</TableCell>
            <TableCell>Attached File</TableCell>
            <TableCell>Auditor's Error</TableCell>
          </TableRow>
        </TableHead>
        <TableBody>
          {audit?.error_cases?.length > 0 ? (
            audit.error_cases.map((error, index) => (
              <TableRow key={index}>
                <TableCell>{error?.select_category || 'None'}</TableCell>
                <TableCell>{error?.error_definition || 'None'}</TableCell>
                <TableCell>{error?.final_error_category || 'None'}</TableCell>
                <TableCell>{error?.remarks || 'None'}</TableCell>
                <TableCell>{error?.remarks_for_operations || 'None'}</TableCell>
                <TableCell>{error?.files && <a href={error.files.url}>{error.files.name}</a>}</TableCell>
                <TableCell>{error?.auditor_error || 'None'}</TableCell>
              </TableRow>
            ))
          ) : (
            <TableRow>
              <TableCell colspan={7} align="center">No errors found</TableCell>
            </TableRow>
          )}
        </TableBody>
      </Table>
    </TableContainer>

    {/* Components with Errors Table */}
    <Typography variant="h6" className="table-title">Components with Errors</Typography>
    <TableContainer>

```



```

<Table>
  <TableHead>
    <TableRow>
      <TableCell>Sub-barcode</TableCell>
      <TableCell>Audit Type</TableCell>
      <TableCell>Case Error</TableCell>
      <TableCell>Case Status</TableCell>
      <TableCell>Remarks</TableCell>
      <TableCell>Remarks for Operations</TableCell>
    </TableRow>
  </TableHead>
  <TableBody>
    {audit?.componentsWithError?.length > 0 ? (
      audit.componentsWithError.map((component, index) => (
        <TableRow key={index}>
          <TableCell>{component?.sub_barcode || 'None'}</TableCell>
          <TableCell>{component?.check_status || 'None'}</TableCell>
          <TableCell>{component?.change_status || 'None'}</TableCell>
          <TableCell>{component?.select_category || 'None'}</TableCell>
          <TableCell>{component?.error_definition || 'None'}</TableCell>
          <TableCell>{component?.remarks_for_operations || 'None'}</TableCell>
        </TableRow>
      ))
    ) : (
      <TableRow>
        <TableCell colspan={6} align="center">No components with errors</TableCell>
      </TableRow>
    )}
  </TableBody>
</Table>
</TableContainer>

{ /* Components without Errors Table */}
<Typography variant="h6" className="table-title">Components without Errors</Typography>
<TableContainer>
  <Table>
    <TableHead>
      <TableRow>
        <TableCell>Sub-Barcode</TableCell>
        <TableCell>Check Status</TableCell>
        <TableCell>Status Change</TableCell>
        <TableCell>Remarks for Audit</TableCell>
        <TableCell>Remarks for Operations</TableCell>
      </TableRow>
    </TableHead>
    <TableBody>
      {audit?.componentsWithoutError?.length > 0 ? (
        audit.componentsWithoutError.map((component, index) => (
          <TableRow key={index}>
            <TableCell>{component?.sub_barcode || 'None'}</TableCell>
            <TableCell>{component?.check_status || 'None'}</TableCell>
            <TableCell>{component?.change_status || 'None'}</TableCell>
            <TableCell>{component?.remarks || 'None'}</TableCell>
            <TableCell>{component?.remarks_for_operations || 'None'}</TableCell>
          </TableRow>
        ))
      ) : (
        <TableRow>
          <TableCell colspan={5} align="center">No components without errors</TableCell>
        </TableRow>
      )}
    </TableBody>
  </Table>
</TableContainer>

```

```

    })
  </TableBody>
</Table>
</TableContainer>
</div>
</Collapse>
</Paper>
);
};

```

export default Accordion;

#Accordion.css

@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@400;600&display=swap');

```

body {
  font-family: 'Montserrat', sans-serif;
  background-color: #f9f9f9;
  color: #333;
  margin: 0;
  padding: 0;
  overflow-x: hidden;
}

```

```

.wrapper {
  padding: 20px;
}

```

```

.headtitle {
  text-align: center;
  margin-bottom: 20px;
}

```

```

.headtitle span {
  font-size: 1.5rem;
  color: #007BFF;
}

```

```

.headtitle h1 {
  font-size: 2rem;
  margin: 10px 0;
}

```

```

.searchBar {
  display: flex;
  justify-content: center;
  gap: 10px;
  margin-bottom: 20px;
}

```

```

.searchBar input {
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
}

```

```

.searchBar button {

```

```
background-color: #007BFF;
color: #fff;
border: none;
padding: 10px 20px;
border-radius: 4px;
cursor: pointer;
transition: background-color 0.3s;
}
```

```
.searchBar button:hover {
  background-color: #0056b3;
}
```

```
.accordion-sect {
  margin-bottom: 10px;
}
```

```
.accordion {
  background-color: #a1c9f5;
  color: white;
  cursor: pointer;
  padding: 18px;
  width: 100%;
  border: none;
  text-align: left;
  outline: none;
  font-size: 15px;
  transition: transform 0.3s, box-shadow 0.3s;
  border-radius: 8px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}
```

```
.accordion.active, .accordion:hover {
  background-color: #80aad6;
}
```

```
.data {
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
}
```

```
.data strong {
  margin-right: 5px;
}
```

```
.panel {
  padding: 0 18px;
  background-color: white;
  max-height: 0;
  overflow: hidden;
  transition: max-height 0.4s ease-out;
  border: 1px solid #ddd;
  border-radius: 0 0 4px 4px;
}
```

```
.accordion.active + .panel {
  max-height: 800px;
}
```

```
.table {
  width: 100%;
  border-collapse: collapse;
  margin: 20px 0;
}

.table th, .table td {
  padding: 12px;
  border: 1px solid #ddd;
  text-align: left;
}

.table th {
  background-color: #f2f2f2;
}

.upper {
  text-align: left;
  margin: 10px 0;
}

.remark {
  white-space: pre-wrap;
}

.reverse p {
  text-align: center;
}

.expand-icon {
  color: white;
  transition: transform 0.3s ease-in-out;
}

.accordion.active .expand-icon {
  transform: rotate(180deg);
}

.accordion-summary:hover .expand-icon {
  color: #0056b3;
}
```