

```

// // import logo from './logo.svg';
// // import './App.css';

// // function App() {
// //   return (
// //     <div className="App">
// //       <header className="App-header">
// //         <img src={logo} className="App-logo" alt="logo" />
// //         <p>
// //           Edit <code>src/App.js</code> and save to reload.
// //         </p>
// //         <a
// //           className="App-link"
// //           href="https://reactjs.org"
// //           target="_blank"
// //           rel="noopener noreferrer"
// //         >
// //           Learn React
// //         </a>
// //       </header>
// //     </div>
// //   );
// // }

// // export default App;
import React, { useState } from 'react';
import axios from 'axios';

// function App() {
//   const [selectedFile, setSelectedFile] = useState(null);
//   const [selectedPages, setSelectedPages] = useState([]);
//   const [loading, setLoading] = useState(false);
//   const [downloadLink, setDownloadLink] = useState('');
//   const [uploadMessage, setUploadMessage] = useState('');
//   const handleFileChange = (event) => {
//     console.log(event.target.files);
//     setSelectedFile(event.target.files[0]);
//   };

//   const handlePageToggle = (pageNumber) => {
//     console.log(pageNumber, selectedPages, selectedFile)
//     if (selectedPages.includes(pageNumber)) {
//       setSelectedPages(selectedPages.filter((page) => page !== pageNumber));
//     } else {
//       setSelectedPages([...selectedPages, pageNumber]);
//     }
//     console.log(pageNumber, selectedPages, selectedFile)

```

```

// };
// const handleUpload = async () => {
//   try {
//     const formData = new FormData();
//     formData.append('file', selectedFile);

//     const response = await axios.post('http://localhost:5000/upload-pdf',
formData, {
//       headers: {
//         'Content-Type': 'multipart/form-data',
//       },
//     });

//     setUploadMessage(response.data.message);
//   } catch (error) {
//     console.error('Error uploading PDF:', error);
//     setUploadMessage('Error uploading PDF. Please try again.');
```

```

//   }
// };
// const handleCreatePDF = async () => {
//   console.log('PDF created successfully');

//   setLoading(true);
//   try {
//     const formData = new FormData();
//     formData.append('file', selectedFile);
//     formData.append('pages', JSON.stringify(selectedPages));
//     console.log(formData, selectedFile, JSON.stringify(selectedPages));
//     const response = await axios.post('http://localhost:5000/create-pdf',
formData, {
//       headers: {
//         'Content-Type': 'multipart/form-data',
//       },
//       responseType: 'blob', // Ensure response is treated as binary data
//     });

//     // Create download link for the new PDF
//     const downloadUrl = window.URL.createObjectURL(new
Blob([response.data]));
//     setDownloadLink(downloadUrl);
//   } catch (error) {
//     // Handle error

//     console.error('Error creating PDF:', error);
//   } finally {
//     setLoading(false);
//   }

```

```

//   };
//   console.log(selectedPages,selectedFile)

//   return (
//     <div>
//       <input type="file" accept=".pdf" onChange={handleFileChange} />
//       {selectedFile && (
//         <div>
//           <h2>Selected Pages</h2>
//           {Array.from({ length: 10 }, (_, i) => (
//             <label key={i}>
//               <input type="checkbox" onChange={() => handlePageToggle(i + 1)}
//             </label>
//             Page {i + 1}
//           )}}
//           <button onClick={handleCreatePDF} disabled={loading}>
//             Create PDF
//           </button>
//           <div>
//             <input type="file" accept=".pdf" onChange={handleFileChange} />
//             <button onClick={handleUpload}>Upload PDF</button>
//             {uploadMessage && <p>{uploadMessage}</p>}
//           </div>
//           {downloadLink && (
//             <div>
//               <a href={downloadLink} download="new_pdf.pdf">
//                 Download New PDF
//               </a>
//             </div>
//           )}
//         </div>
//       )}
//     </div>
//   );
// }

// export default App;

```

```

// const express = require('express');
// const multer = require('multer');
// const PDFDocument = require('pdfkit');
// const fs = require('fs');
// const cors = require('cors');

```

```

// const app = express();
// const port = 5000;
// const path = require('path');
// const storage = multer.diskStorage({
//   destination: function (req, file, cb) {
//     // cb(null, 'uploads/');
//     cb(null, path.join(__dirname, '/uploads/'));
//     console.log(path.join(__dirname, '/uploads/'));
//   },
//   filename: function (req, file, cb) {
//     cb(null, file.originalname);
//   },
// });

// const upload = multer({ storage: storage });

// app.use(cors()); // Enable CORS for all routes

// app.post('/upload-pdf', upload.single('file'), (req, res) => {
//   console.log('PDF uploaded successfully');
//   res.status(200).json({ message: 'PDF uploaded successfully' });
//   console.log('PDF uploaded successfully');

// });

// app.post('/create-pdf', upload.single('file'), async (req, res) => {
//   const { file, pages } = req.body;
//   const selectedPages = JSON.parse(pages);

//   // Path to the uploaded PDF file
//   const filePath = req.file.path;

//   // Create a new PDF document
//   const doc = new PDFDocument();

//   // Create a write stream to save the new PDF
//   const writeStream = fs.createWriteStream('public/output.pdf');
//   console.log('hhhd b d h h d d h ', writeStream, 'sssssssss', 'dddd', filePath);
//   // Pipe the PDF document into the write stream
//   doc.pipe(writeStream);

//   // Use pdfkit to read the original PDF and extract selected pages
//   const reader = fs.createReadStream(filePath);
//   const originalPDF = new PDFDocument();

//   // Pipe the original PDF into the new PDF document
//   originalPDF.pipe(doc);

```

```

// let pageNumber = 1;

// reader.on('page', () => {
//     if (selectedPages.includes(pageNumber)) {
//         originalPDF.switchToPage(pageNumber);
//         originalPDF.addPage();
//         originalPDF.text(`Page ${pageNumber}`);
//         // You can customize the content or styling of the page here
//     }
//     pageNumber++;
// });

// reader.on('end', () => {
//     originalPDF.end();
// });

// // Close the write stream when the new PDF is finished writing
// writeStream.on('finish', () => {
//     res.sendFile(`${__dirname}/public/output.pdf`); // Send the new PDF file
as a response
// });
// });

// // app.post('/create-pdf', upload.single('file'), async (req, res) => {
// //     const { file, pages } = req.body;
// //     const selectedPages = JSON.parse(pages);
// //     console.log('PDF created successfully');

// //     // Path to the uploaded PDF file
// //     const filePath = req.file.path;
// //     const doc = new PDFDocument();
// //     const writeStream = fs.createWriteStream('public/output.pdf'); // Save
the new PDF in a public directory

// //     // Pipe the PDF document into the write stream
// //     doc.pipe(writeStream);

// //     // Use pdftk to read the original PDF and extract selected pages
// //     const reader = fs.createReadStream(filePath);
// //     const originalPDF = new PDFDocument();

// //     // Pipe the original PDF into the new PDF document
// //     originalPDF.pipe(doc);

// //     let pageNumber = 1;

// //     reader.on('page', () => {
// //         if (selectedPages.includes(pageNumber)) {

```

```

// //      originalPDF.switchToPage(pageNumber);
// //      originalPDF.addPage();
// //      originalPDF.text(`Page ${pageNumber}`);
// //      // You can customize the content or styling of the page here
// //      }
// //      pageNumber++;
// //  });

// //  reader.on('end', () => {
// //      originalPDF.end();
// //  });

// //  writeStream.on('finish', () => {
// //      res.sendFile(`${__dirname}/public/output.pdf`); // Send the new PDF
file as a response
// //  });

// //      // const doc = new PDFDocument();
// //      // const writeStream = fs.createWriteStream('public/output.pdf'); //
Save the new PDF in a public directory
// //      // doc.pipe(writeStream);
// //      // // Use pdftk to read the original PDF and extract selected pages
// //      // const reader = fs.createReadStream(filePath);
// //      // const originalPDF = new PDFDocument();
// //      // originalPDF.pipe(doc);
// //      // let pageNumber = 1;
// //      // originalPDF.registerFont('Helvetica', 'fonts/Helvetica.ttf');
// //      // reader.on('page', () => {
// //      //   if (selectedPages.includes(pageNumber)) {
// //      //     originalPDF.switchToPage(pageNumber);
// //      //     originalPDF.addPage();
// //      //     originalPDF.fontSize(12).font('Helvetica').text(`Page
${pageNumber}`, 50, 50);
// //      //     // You can customize the content or styling of the page here
// //      //     }
// //      //     pageNumber++;
// //      //   });
// //      // reader.on('end', () => {
// //      //     originalPDF.end();
// //      //   });
// //      // doc.end();
// //      // writeStream.on('finish', () => {
// //      //     res.sendFile(`${__dirname}/uploads/output.pdf`); // Send the new
PDF file as a response
// //      //   });
// //  });

```

```
// app.listen(port, () => {  
//   console.log(`Server is running on http://localhost:${port}`);  
// });
```

Hi preet singh