

```
import React, { useState, Dispatch, SetStateAction } from "react";
import {
  LeftOutlined,
  RightOutlined,
  ZoomInOutlined,
  ZoomOutOutlined,
} from "@ant-design/icons";
import { Modal, Button, Row, Progress } from "antd";
import { Document, Page } from "react-pdf";
import { trpc } from "@/_trpc/client";
```

```
import styles from "./AttachmentPreviewModal.module.css";
```

```
export interface IFilePreviewModalProps {
  fileKey: string;
  onClose: () => void;
}
```

```
export interface ZoomActionsProps {
  isLoading: boolean;
  scale: number;
  setScale: Dispatch<SetStateAction<number>>;
}
```

```
export interface PageActionsProps {
  isLoading: boolean;
  pageNumber: number;
  numPages: number;
  setPageNumber: Dispatch<SetStateAction<number>>;
}
```

```
export interface PDFLoadingProgressProps {
  loadingProgress: number;
}
```

```
export interface ProgressEvent {
  loaded: number;
  total: number;
}
```

```
const ZoomActions: React.FC<ZoomActionsProps> = ({
  isLoading,
  scale,
  setScale,
```

```

}) => {
  const zoomIn = () => {
    if (scale <= 2) {
      setScale(scale + 0.1);
    }
  };

  const zoomOut = () => {
    if (scale > 0.1) {
      setScale(scale - 0.1);
    }
  };

  if (!isLoading) {
    return null;
  }

  return (
    <div className={styles.zoomActions}>
      <Button onClick={zoomIn} icon={<ZoomInOutlined />} />
      <Button
        onClick={zoomOut}
        disabled={scale <= 0.1}
        icon={<ZoomOutOutlined />}
      />
    </div>
  );
};

const PageActions: React.FC<PageActionsProps> = ({
  isLoading,
  pageNumber,
  numPages,
  setPageNumber,
}) => {
  const isLastPage = pageNumber === numPages;
  const isFirstPage = pageNumber === 1;

  const goBack = () => {
    if (pageNumber > 1) {
      setPageNumber(pageNumber - 1);
    }
  };
};

```

```

const goForward = () => {
  if (pageNumber < numPages) {
    setPageNumber(pageNumber + 1);
  }
};

if (!isLoading) {
  return null;
}

return (
  <div className={styles.pageActions}>
    <Button onClick={goBack} disabled={isFirstPage} icon={<LeftOutlined />} />
    <p>
      {pageNumber} / {numPages}
    </p>
    <Button
      onClick={goForward}
      disabled={isLastPage}
      icon={<RightOutlined />}
    />
  </div>
);
};

const PDFLoadingProgress: React.FC<PDFLoadingProgressProps> = ({
  loadingProgress,
}) => {
  const isLoading = loadingProgress === 100;

  return (
    <div
      className={isLoading ? styles.hiddenContent : styles.progressBarContainer}
    >
      <Progress size="small" type="circle" percent={loadingProgress} />
    </div>
  );
};

const AttachmentPreviewModal = (props: IFilePreviewModalProps) => {
  const fileUrl = trpc.attachment.getAttachmentS3UrlFromKey.useQuery(
    props.fileKey,
  ).data;

```

```

const [numPages, setNumPages] = useState<number>(0);
const [pageNumber, setPageNumber] = useState<number>(1);
const [scale, setScale] = useState<number>(0.7);
const [loadingProgress, setLoadingProgress] = useState<number>(0);

const onLoadProgress = ({ loaded, total }: ProgressEvent) => {
  setLoadingProgress(Math.round((loaded / total) * 100));
};

const onLoadSuccess = ({ numPages }: { numPages: number }) => {
  setNumPages(numPages);
};

const isLoading = loadingProgress === 100;

return (
  <Modal
    title="Attachment Preview"
    open={true}
    onCancel={props.onClose}
    width={800}
    footer={null}
    centered
  >
    <ZoomActions isLoading={isLoading} scale={scale} setScale={setScale} />
    <div className={styles.mainContent}>
      <div className={isLoading ? "" : styles.hiddenContent}>
        <Document
          file={fileUrl}
          onLoadSuccess={onLoadSuccess}
          onLoadProgress={onLoadProgress}
        >
          <Page
            scale={scale}
            pageNumber={pageNumber}
            renderTextLayer={false}
            renderAnnotationLayer={false}
            key={`page_${pageNumber + 1}`}
            className="pdf-page"
          />
        </Document>
      </div>
      <PDFLoadingProgress loadingProgress={loadingProgress} />
    </div>
  </Modal>
);

```

```
    <PageActions
      isLoading={isLoading}
      pageNumber={pageNumber}
      numPages={numPages}
      setPageNumber={setPageNumber}
    />
  </Modal>
);
};

export default AttachmentPreviewModal;
```