

# Component Documentation

This file contains documentation for the React components used in this project.

## DraggableSignature.tsx

### Description

This component provides a draggable and resizable signature element that can be placed on a document. It also supports rotation. The component displays an image of the signature and allows the user to interact with it by dragging, resizing, and rotating.

### Props

Prop	Type	Description
data	SignatureAnnotation	An object containing the si
onUpdate	(id: string, patch: Partial	A function to call when the
onDelete	(id: string) => void	A function to call when the
onSelect	(id: string) => void	A function to call when the
scale	number	The current scale of the do
isSelected	boolean	A boolean indicating whethe

### Example

```

<DraggableSignature
  data={{
    id: 'signature-1',
    pageIndex: 0,
    image: 'data:image/png;base64,...',
    x: 100,
    y: 100,
    width: 150,
    height: 60,
    rotation: 0,
  }}
  onUpdate={(id, patch) => console.log('Update signature:', id, patch)}
  onDelete={(id) => console.log('Delete signature:', id)}
  onSelect={(id) => console.log('Select signature:', id)}
  scale={1.5}
  isSelected={true}
/>

```

## DraggableTextPro.tsx

### Description

This component provides a professional-looking, draggable, resizable, and rotatable text box that can be placed on a document. It allows for in-place editing of the text and supports various formatting options, such as font size, color, bold, italic, underline, and alignment.

### Props

Prop	Type	Description
data	PlacedText	An object containing the te
onUpdate	(id: string, patch: Partial	A function to call when the
onDelete	(id: string) => void	A function to call when the
onSelect	(id: string) => void	A function to call when the
scale	number	The current scale of the do
isSelected	boolean	A boolean indicating whethe

### Example

```

<DraggableTextPro
  data={{
    id: 'text-1',
    text: 'Hello, world!',
    x: 100,
    y: 100,
    fontSize: 14,
    pageIndex: 0,
    color: '#000000',
  }}
  onUpdate={(id, patch) => console.log('Update text:', id, patch)}
  onDelete={(id) => console.log('Delete text:', id)}
  onSelect={(id) => console.log('Select text:', id)}
  scale={1.5}
  isSelected={true}
/>

```

## SelectionHandles.tsx

### Description

This component provides the selection handles (resize and rotate) for a draggable element. It displays 8 resize handles around the element and a rotation handle at the top. It also includes a delete button.

### Props

Prop	Type	Description
rotation	number	The current rotation of the
onResize	(direction: string) => (e:	A function to call when a r
onRotate	(e: React.MouseEvent) => vo	A function to call when the
onDelete	() => void	A function to call when the

### Example

```

{isSelected && (
  <SelectionHandles
    rotation={rotation}
    onResize={handleResize}
    onRotate={handleRotate}
    onDelete={() => onDelete(data.id)}
  />
)
}

```

## PDFViewer.tsx

### Description

This component is a full-featured PDF viewer that allows for rendering PDF documents, navigating between pages, and adding annotations such as text, signatures, and shapes. It also supports zooming and fitting the document to the screen.

### Props

Prop	Type	Description			
activeTool	'text' \  'signature' \  'shapes'	The currently active annotation tool.	null		The currently active annotation tool.
inputText	string	The text to be added as a new annotation.			
setInputText	(text: string)	A function to set the input text for a new annotation.			
signature	string \  null	The data URL of a signature image.	The data URL		
isSignatureFl	boolean	A boolean indicating if the current tool is a signature.			
setIsSignature	(v: boolean)	A function to set the isSignature flag.			
textFormat	TextFormat	An object containing text format settings.			
selectedShape	ShapeType \  null	The currently selected shape type or null.	The currently selected shape type.		
isShapeFloati	boolean	A boolean indicating if the selected shape is floating.			
setIsShapeFlo	(v: boolean)	A function to set the isShapeFloati flag.			
onExitAddText	() => void	A function to handle exiting text addition mode.			
selectedAnnot	string \  null	The ID of the currently selected annotation.	The ID of the currently selected annotation.		
onSelectAnnot	(id: string \  null) => void	A function to handle selecting an annotation by ID.	A function to handle selecting an annotation by ID.		

### Example

```

<PDFViewer
  activeTool="text"
  inputText="Hello"
  setInputText={setInputText}
  signature={null}
  isSignatureFloating={false}
  setIsSignatureFloating={setIsSignatureFloating}
  selectedAnnotationId={selectedAnnotationId}
  onSelectAnnotation={setSelectedAnnotationId}
/>

```

## CropTool.tsx

### Description

This component provides a tool for cropping and rotating an image. It displays the image with a crop box that can be resized and moved. It also includes a rotation handle to rotate the image. The component can auto-detect the bounds of a signature within the image.

### Props

Prop	Type	Description
currentImage	string	The data URL of the image to crop.
onApplyCrop	(croppedDataURL: string) =>	A function to call when the crop is applied.
onToggleTilt	() => void	A function to call when the crop tilt is toggled.
onExit	() => void	A function to call when the crop tool is exited.

### Example

```

<CropTool
  currentImage="data:image/png;base64,...."
  onApplyCrop={(croppedImage) => console.log('Cropped image:', croppedImage)}
  onToggleTilt={() => console.log('Toggle tilt')}
  onExit={() => console.log('Exit crop tool')}
/>

```

## SignaturePreview.tsx

## Description

This component displays a preview of the original and processed signature images. It allows the user to download the final signature, reset the changes, and access the crop and tilt tools.

## Props

Prop	Type	Description
originalImage	string	The data URL of the original image.
processedImage	string	The data URL of the processed image.
onDownload	() => void	A function to call when the download button is clicked.
onReset	() => void	A function to call when the reset button is clicked.

## Example

```
<SignaturePreview  
  originalImage="data:image/png;base64,...."  
  processedImage="data:image/png;base64,...."  
  onDownload={() => console.log('Download signature')}  
  onReset={() => console.log('Reset signature')}  
>
```

## PDFEditor.tsx

## Description

This is the main component for the PDF editor. It integrates the PDFViewer, PDFThumbnails, and various toolbars to provide a complete PDF editing experience. It manages the active tool, annotations, and other state related to the editor.

## Props

This component does not accept any props.

## Example

```
<PDFEditor />
```

## ImageFilterEditor.tsx

### Description

This component provides an editor for applying filters to an image, specifically for enhancing a signature. It includes controls for brightness and contrast, and it automatically processes the image to create a clean, transparent signature.

### Props

Prop	Type	Description
imageSrc	string	The data URL of the image to edit.
onDone	(dataURL: string) => void	A function to call when the editing process is completed.
onCancel	() => void	A function to call when the user cancels the editing process.

### Example

```
<ImageFilterEditor
  imageSrc="data:image/png;base64,..."
  onDone={(editedImage) => console.log('Edited image:', editedImage)}
  onCancel={() => console.log('Cancel editing')}
/>
```

## TiltCorrectionTool.tsx

### Description

This component provides a tool for correcting the perspective tilt of an image in 3D. It allows the user to adjust the horizontal and vertical tilt of the image using sliders.

### Props

<b>Prop</b>	<b>Type</b>	<b>Description</b>
processedImage	string	The data URL of the image to be processed.
onApply	(correctedImage: string) =>	A function to call when the image is successfully processed.
onCancel	() => void	A function to call when the process is canceled.

## Example

```
<TiltCorrectionTool  
  processedImage="data:image/png;base64,..."  
  onApply={(correctedImage) => console.log('Corrected image:', correctedImage)}  
  onCancel={() => console.log('Cancel correction')}  
>
```

## PDFThumbnails.tsx

### Description

This component displays thumbnails of the pages in a PDF document. It allows the user to navigate between pages by clicking on the thumbnails.

### Props

This component does not accept any props.

## Example

```
<PDFThumbnails />
```

## TextSignature.tsx

### Description

This component allows users to create a text-based signature. It provides a text input for the user to type their name and a selection of fonts to choose from. The component then generates an image of the signature.

### Props

Prop	Type	Description
onGenerate	(dataURL: string) => void	A function to call when the
closeModal	() => void	A function to call to close

## Example

```
<TextSignature  
  onGenerate={(dataURL) => console.log('Signature generated:', dataURL)}  
  closeModal={() => console.log('Close modal')}  
/>
```

## shapes-tool.tsx

### Description

This file contains two components: ShapesTool and ShapeRenderer.

ShapesTool provides a UI for selecting a shape to add to the document.

ShapeRenderer is responsible for rendering the selected shape on the PDF.

### Props

#### ShapesTool

Prop	Type	Description
onShapeSelect	(shape: ShapeType) => void	A function to call when a s

#### ShapeRenderer

Prop	Type	Description
shape	ShapeAnnotation	The shape annotation object
scale	number	The current scale of the do

## Example

```
// ShapesTool
<ShapesTool onShapeSelect={(shape) => console.log('Shape selected:', shape)} />

// ShapeRenderer
<ShapeRenderer shape={shapeAnnotation} scale={1.5} />
```

## IslandCard.tsx

### Description

This component displays a card for a step in the IR pathway roadmap. It shows the step number, title, subtitle, and badges for the responsible person and timeline. The card's appearance changes based on whether the step is completed.

### Props

Prop	Type	Description
step	Step	The step data to display.
isCompleted	boolean	A boolean indicating if the
onClick	() => void	A function to call when the

### Example

```
<IslandCard
  step={step}
  isCompleted={true}
  onClick={() => console.log('Card clicked')}
/>
```

## CompletionDialog.tsx

### Description

This component displays a dialog to congratulate the user upon completing all the steps in the IR pathway roadmap. It shows the user's progress and some achievement badges.

## Props

Prop	Type	Description
open	boolean	A boolean to control the visibility of the component.
onOpenChange	(open: boolean) => void	A function to call when the visibility changes.
progress	number	The user's progress percent.

## Example

```
<CompletionDialog
  open={true}
  onOpenChange={setOpen}
  progress={100}
/>
```

## compress-pdf.tsx

## Description

This component provides a UI for compressing a PDF file. It allows the user to select a PDF file, sends it to a server for compression, and then provides a download link for the compressed file.

## Props

This component does not accept any props.

## Example

```
<Compress />
```