# **Image to PDF Converter**

## **Overview**

The **Image to PDF Converter** is a web application that allows users to upload multiple images (JPG/PNG), reorder them via drag-and-drop, preview the final PDF, and download the generated PDF. The application is built using **ReactJS** and utilizes **jsPDF** for PDF generation.

## **Objectives**

* Provide an intuitive UI for users to upload and manage images.
* Enable drag-and-drop functionality for reordering images.
* Generate a PDF from the uploaded images.
* Allow users to preview the PDF before downloading.

## **Tech Stack**

* **Frontend**: ReactJS
* **Libraries**:
  + jsPDF for PDF generation.
  + react-dnd for drag-and-drop image reordering.
* **Styling**: CSS for layout and responsiveness.

## **System Design**

### **Component Breakdown**

1. **App Component**: The main component that holds the state and logic.
2. **ImageUpload Component**: Handles image selection and file validation.
3. **DraggableImage Component**: Implements drag-and-drop functionality using react-dnd.
4. **DroppableArea Component**: Defines the drop zones for reordering images.
5. **PDFPreview Component**: Displays a preview of the generated PDF.
6. **DownloadButton Component**: Allows users to download the final PDF.

### **Workflow**

1. **User uploads images** → Images are converted to URLs and stored in the state.
2. **User reorders images** → Drag-and-drop updates the state dynamically.
3. **User converts to PDF** → jsPDF generates the PDF using the images.
4. **User previews the PDF** → The generated PDF is displayed using a blob URL.
5. **User downloads the PDF** → The final document is available for download.

## **API & Libraries Usage**

* **jsPDF**: Used to generate the PDF with images.
* **react-dnd**: Enables drag-and-drop image reordering.

## **Error Handling & Edge Cases**

* Validate file formats (only JPG and PNG allowed).
* Ensure large image files are handled efficiently.
* Prevent duplicate uploads.
* Display meaningful error messages.

## **Future Enhancements**

* Support additional image formats (e.g., BMP, GIF).
* Implement image compression for smaller file sizes.
* Provide customization options (page size, orientation, margins).
* Add cloud storage integration for saving PDFs.

## **Conclusion**

This project provides a smooth and efficient way for users to convert multiple images into a single PDF file. By leveraging **ReactJS**, **jsPDF**, and **react-dnd**, it ensures an interactive and user-friendly experience.