```typescript
import "./style.css";

function initialiseIt() {
  const dropinBox = document.querySelector(".dropin-box")! as HTMLDivElement;

  window.addEventListener("dragover", (e: any) => {
    e.preventDefault();
    e.stopPropagation();
    e.dataTransfer.dropEffect = "none";
  });

  window.addEventListener("drop", (e: any) => {
    e.preventDefault();
    e.stopPropagation();
  });

  dropinBox?.addEventListener("dragenter", (e: any) =>
e.target.classList.add("active-drag"));
  //
  dropinBox?.addEventListener("dragleave", (e: any) =>
e.target.classList.remove("active-drag"));

  // drag over event
  dropinBox?.addEventListener("dragover", (e: any) => {
    e.preventDefault();
    e.stopPropagation();
  });

  // Drop Event
  dropinBox?.addEventListener("drop", (e: any) => {
    e.preventDefault();
    e.stopPropagation();

    const { files } = e.dataTransfer;

    const arrayOfFiles = acceptedFile(files);

    if (arrayOfFiles) {
      filePreview(arrayOfFiles);
    }
  });

  type Files = { name: string; size: number; type: string; lastModified: Date
}[] | any;

  // File Accept Function
  function acceptedFile(files: Files) {
```

```typescript
    return [...files].filter((file) => ["image/jpg", "image/jpeg",
"image/png", "image/svg", "application/pdf"].includes(file.type));
  }

  // Determine which HTML Tag to render for previewing
  function chooseTagToPrview(e: Event | any, file: { name: string; type:
string }) {
    if (["image/jpg", "image/jpeg", "image/png",
"image/svg"].includes(file.type)) {
      return `<img src="${e.target?.result}" alt="${file.name}" />`;
    }

    if (["application/pdf"].includes(file.type)) {
      return `<embed src="${e.target?.result}" type="application/pdf"/>`;
    }
  }

  // Preview file function
  function filePreview(files: Files) {
    const filesBox = document.querySelector(".files-box");

    [...files].forEach((file) => {
      const fileReader = new FileReader();

      fileReader.addEventListener("loadend", (e: Event | any) => {
        const fileBox = document.createElement("div");
        fileBox.classList.add("file-box");

        fileBox.innerHTML = `
          <div class="file-preview-box">${chooseTagToPrview(e, file)}</div>

          <div class="file-details-box">
            <div class="file-name-box">
              <span class="file-name-label">File name: </span>
              <span class="file-name">${file.name}</span>
            </div>

            <div class="file-size-box">
              <span class="file-size-label">File size: </span>
              <span class="file-size-value">${file.size}k</span>
            </div>

            <div class="file-date-box">
              <span class="file-date-label">File date: </span>
              <span class="file-date-value">${new
Date(file.lastModified).toDateString()}</span>
            </div>
          </div>
```

```
                `;

        filesBox?.append(fileBox);
      });

      fileReader.readAsDataURL(file);
    });
  }
}

if ("draggable" in document.createElement("div")) {
  initialiseIt();
}
```