Experiment No. 5

Title: Drag and Drop and Stay on page principles

Batch: A3 **Roll No.:16010421073** **Experiment No.:5**

Aim: To create wireframe for Web UI –Drag and Drop and stay on page principles

Resources needed: Wireframing tool

Theory:

Drag and Drop

There are at least 15 events available for cueing the user during a drag and drop interaction:

Page Load: Before any interaction occurs, you can pre-signify the availability of drag and drop. For example, you could display a tip on the page to indicate draggability.

Mouse Hover: The mouse pointer hovers over an object that is draggable.

Mouse Down: The user holds down the mouse button on the draggable object.

Drag Initiated: After the mouse drag starts (usually some threshold—3 pixels).

Drag Leaves Original Location: After the drag object is pulled from its location or object that contains it.

Drag Re-Enters Original Location: When the object re-enters the original location.

Drag Enters Valid Target: Dragging over a valid drop target.

Drag Exits Valid Target: Dragging back out of a valid drop target.

Drag Enters Specific Invalid Target: Dragging over an invalid drop target.

Drag Is Over No Specific Target: Dragging over neither a valid or invalid target. Do you treat all areas outside of valid targets as invalid?

Drag Hovers Over Valid Target: User pauses over the valid target without dropping the object. This is usually when a spring loaded drop target can open up. For example, drag over a folder and pause, the folder opens revealing a new area to drag into.

Drag Hovers Over Invalid Target: User pauses over an invalid target without dropping the object.

Drop Accepted: Drop occurs over a valid target and drop has been accepted.

Drop Rejected: Drop occurs over an invalid target and drop has been rejected. Do you zoom back the dropped object?

Drop on Parent Container: Is the place where the object was dragged from special? Usually this is not the case, but it may carry special meaning in some contexts.

During each event one can visually manipulate a number of actors. The page elements available include:

- Page (e.g., static messaging on the page)
- Cursor
- Tool Tip
- Drag Object (or some portion of the drag object, e.g., title area of a module)

- Drag Object’s Parent Container
- Drop Target

Stay on page Principle includes:

- **Overlays**

Instead of going to a new page, a mini-page can be displayed in a lightweight layer over the page.

- **Inlays**

Instead of going to a new page, information or actions can be inlaid within the page.

- **Virtual Pages**

By revealing dynamic content and using animation, we can extend the virtual space of the page.

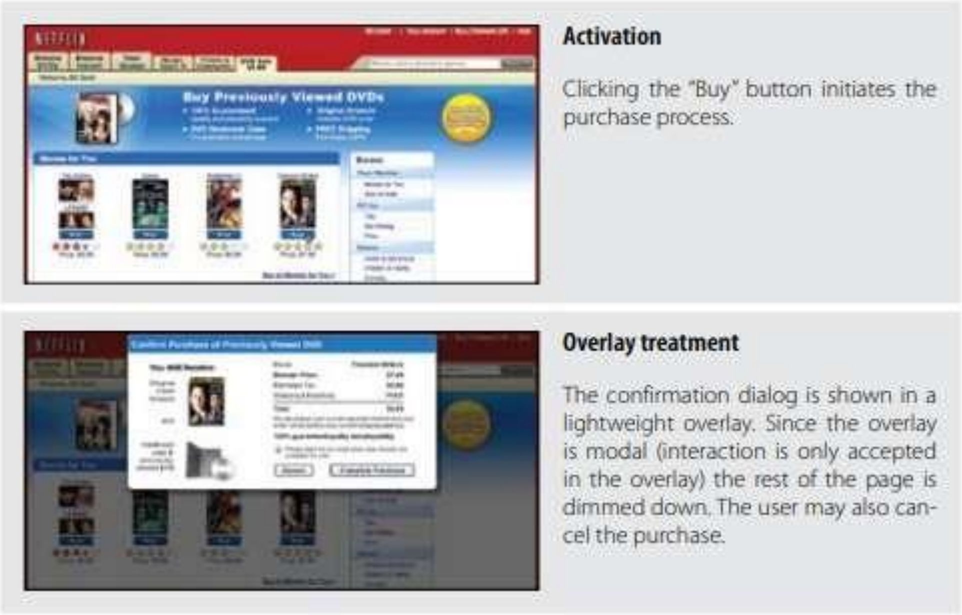
Overlay

Overlays are really just lightweight pop ups. One uses the term lightweight to make a clear distinction between it and the normal idea of a browser pop up.

The three specific types of overlays:

- Dialog Overlays,
- Detail Overlays, and
- Input Overlays

Dialog Overlay



Activation

Clicking the “Buy” button initiates the purchase process.

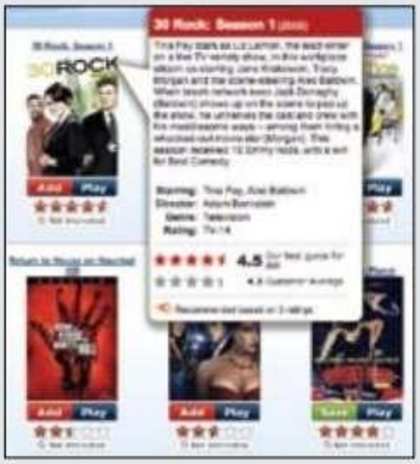
Overlay treatment

The confirmation dialog is shown in a lightweight overlay. Since the overlay is modal (interaction is only accepted in the overlay) the rest of the page is dimmed down. The user may also cancel the purchase.

Figure *Netflix uses a lightweight pop up to confirm a previously viewed DVD purchase; in addition, it uses the Lightbox Effect to indicate modality*

Detail Overlay

The Detail Overlay allows an overlay to present additional information when the user clicks or hovers over a link or section of content. Toolkits now make it easier to create overlays across different browsers and to request additional information from the server without refreshing the page.



Detail overlay activation

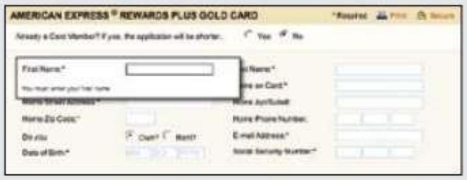
However, often more information is needed to decide whether a movie should be played or added to a movie queue.

By providing a synopsis along with personalized recommendation information, the user can quickly make a determination.

The movie detail information is displayed after a slight delay.

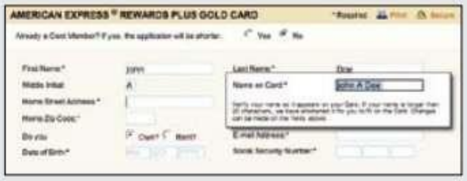
Input Overlay

Input Overlay is a lightweight overlay that brings additional input information for each field tabbed into.



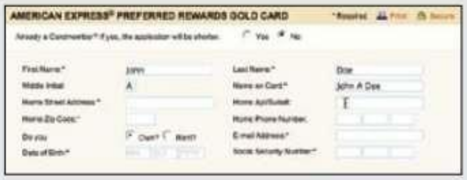
Input overlay

Tabbing or clicking into any field wraps the field in an overlay. The overlay provides additional input information.



Obscuring fields

The overlay does obscure fields just below it, but not to the left or right.

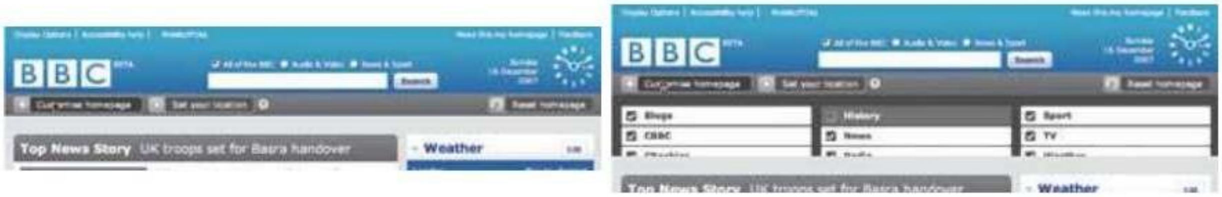


Deactivation

Clicking anywhere removes the overlay. This lets the user click through the field covered by the overlay.

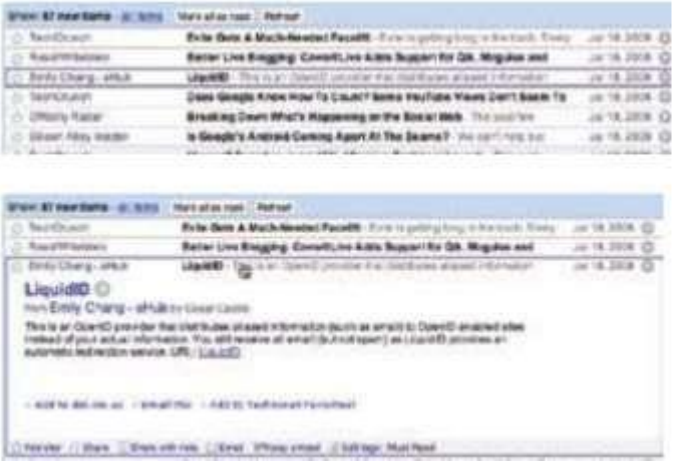
Inlays

A simple technique is to expand a part of the page, revealing a dialog area within the page.



List Inlays

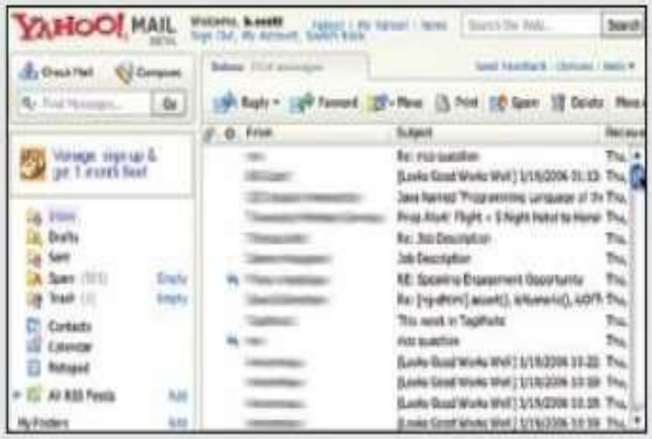
Lists are a great place to use Inlays. The List Inlay works as an effective way to hide detail until needed.



Virtual Page

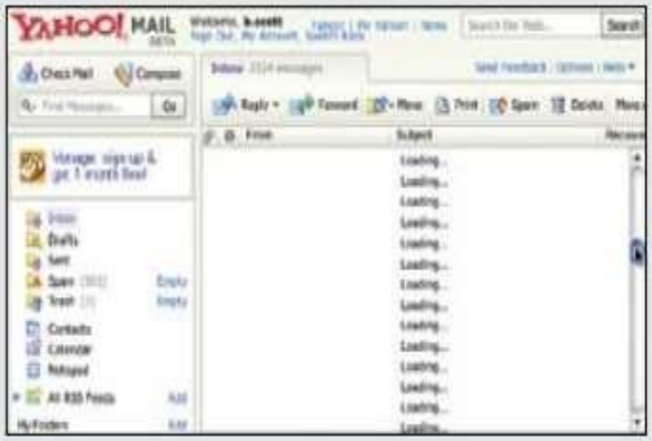
- Overlays allow bringing additional interactions or content in a layer above the current page. Inlays allow doing this within the page itself.
- However, another powerful approach to keeping users engaged on the current page is to create a virtual page.
- Patterns that support virtual pages include:
    - Virtual Scrolling
    - Inline Paging
    - Scrolled Paging
    - Panning
    - Zoomable User Interface

Virtual scrolling



Scrolled list

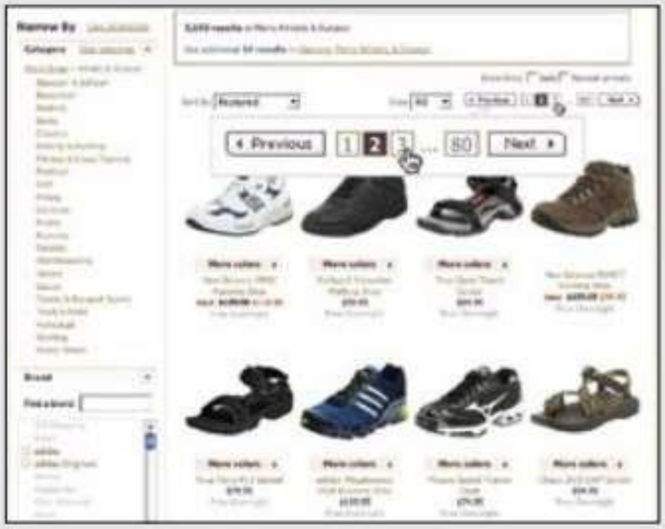
Email messages are displayed as a scrolled list. This has been the normal approach on desktop mail clients. Yahoo! Mail brings that approach to the Web.



Scrolling

Messages are loaded on demand. As the user scrolls, the content items are filled in. While loading, the message lines are replaced with the text "Loading...".

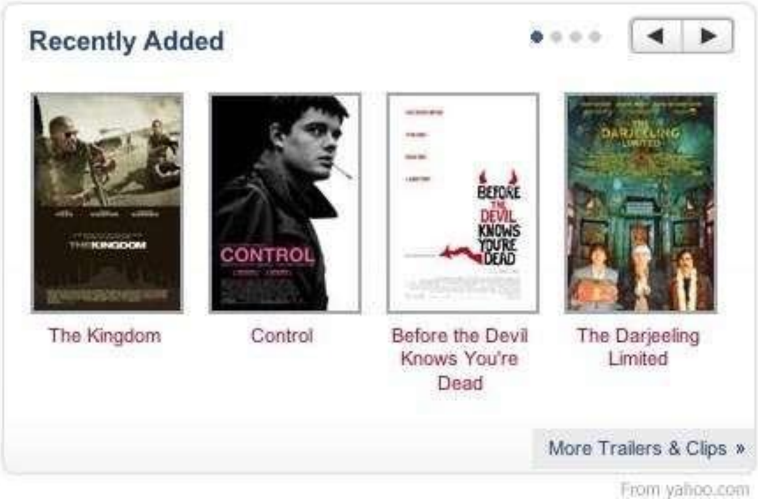
Inline Paging



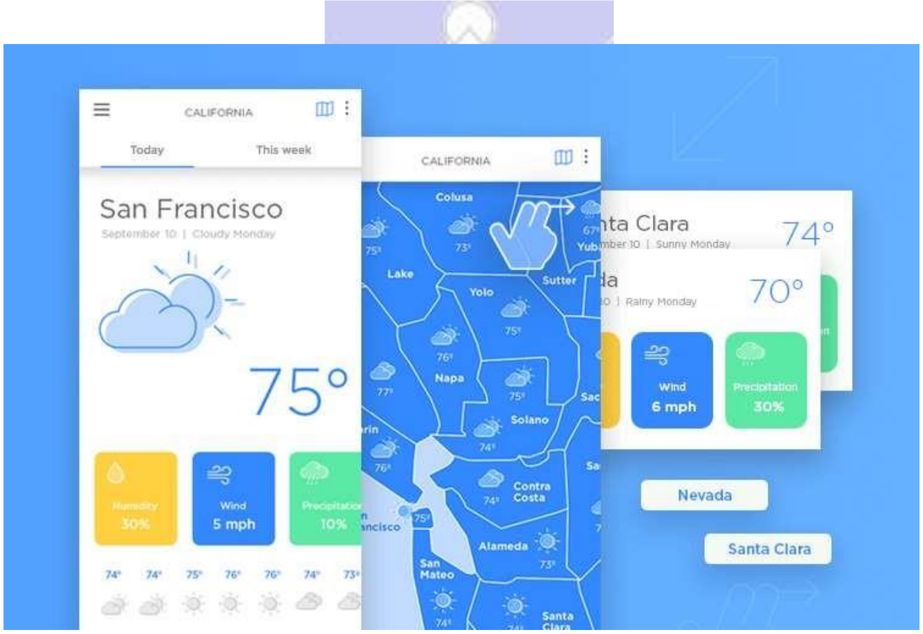
Paginated results

Searching for "Men's athletic shoes" displays a traditional-looking set of search results. The pagination controls are familiar (shown as an exploded callout).

Scrolled paging carousel



Panning and Zoomable Interface

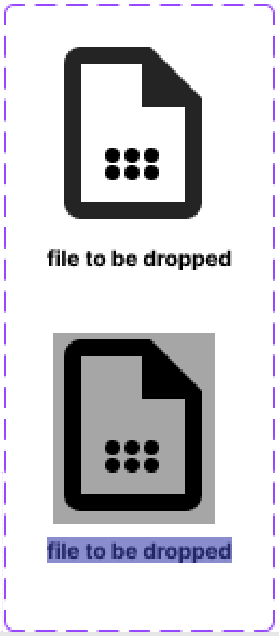


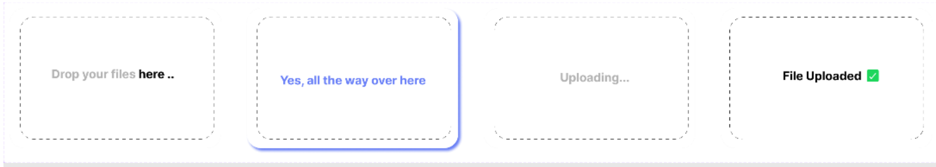
Procedure:

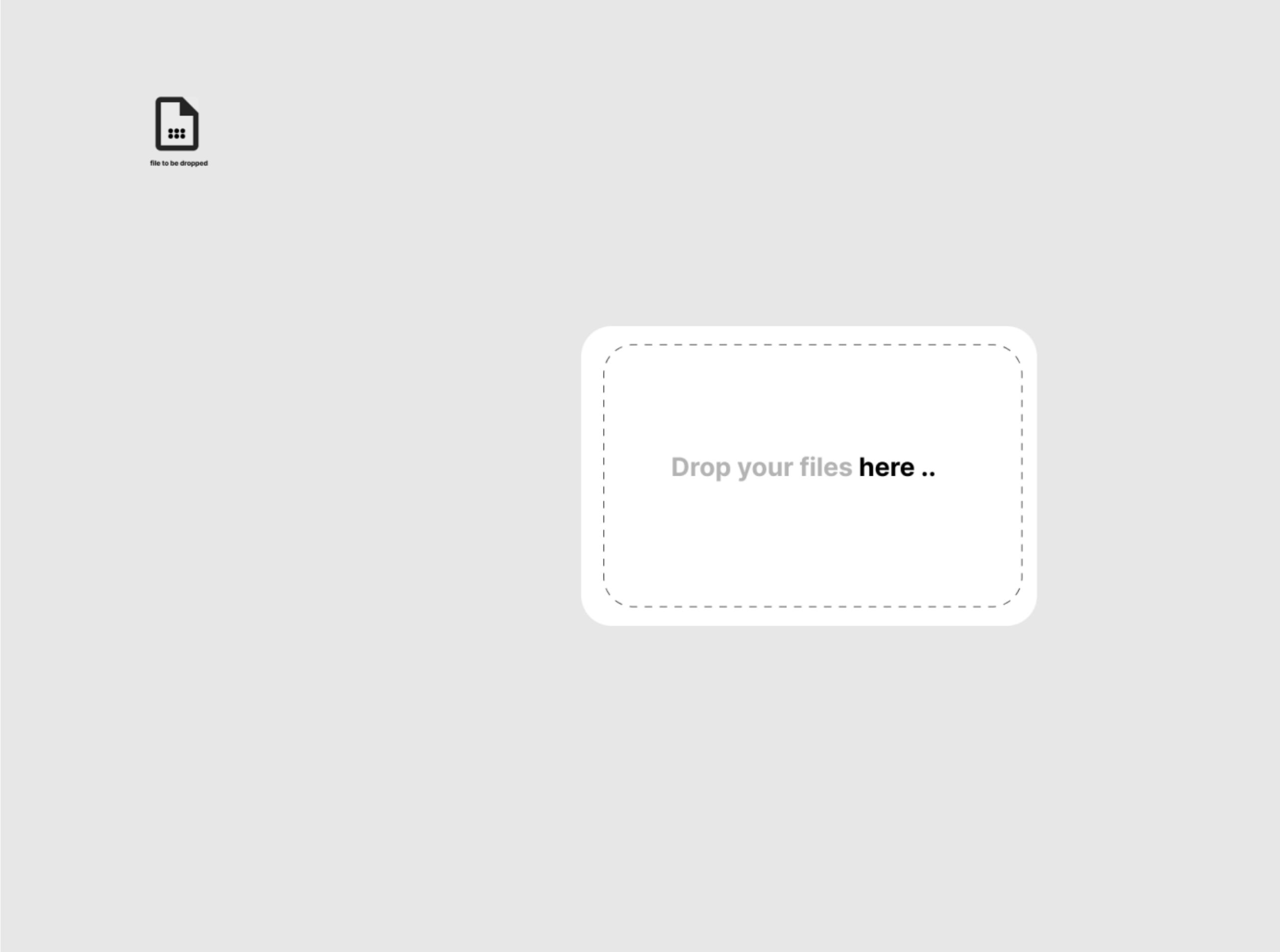
- Create wireframes incorporating Drag and Drop and Stay on page principles for chosen topic

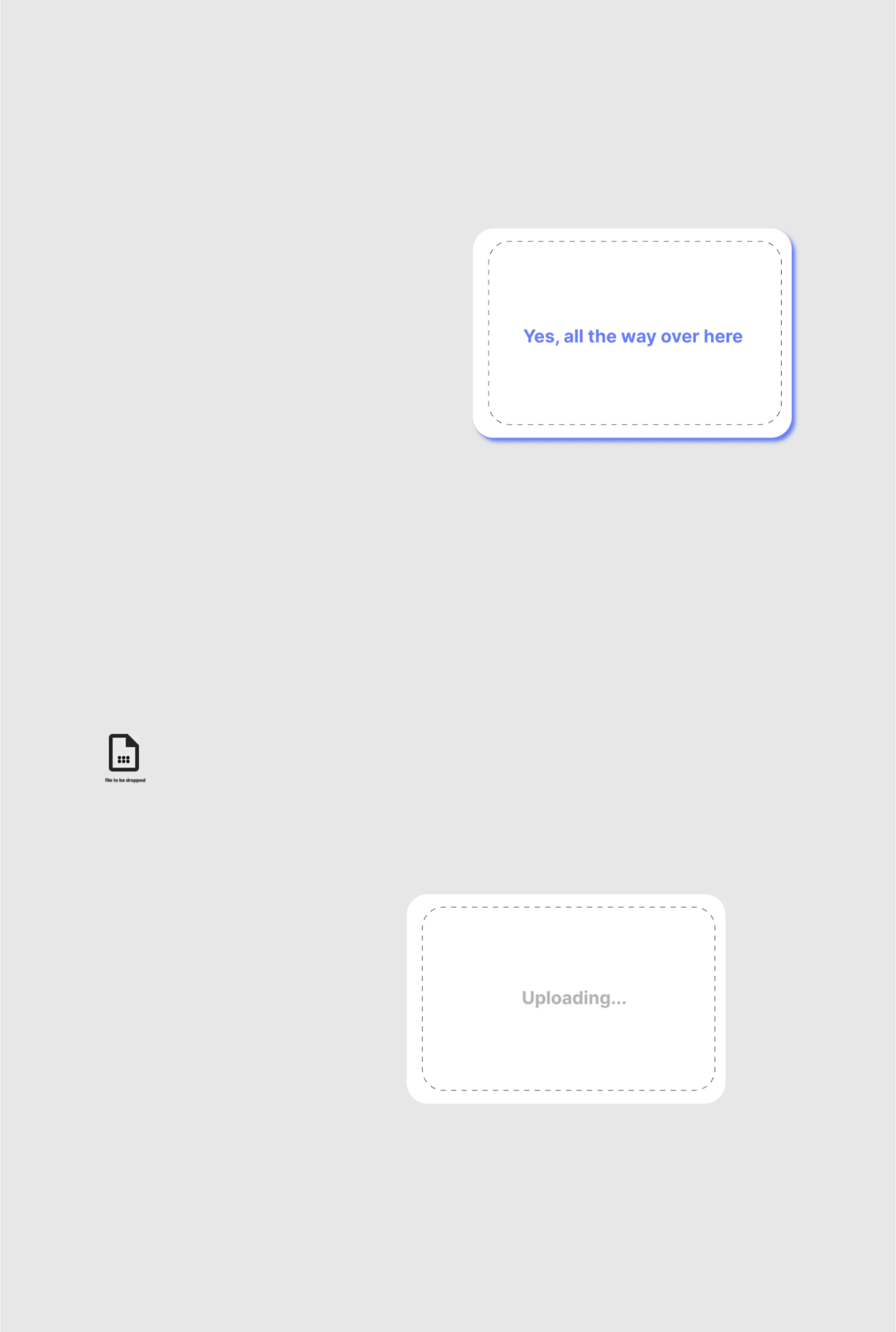
---

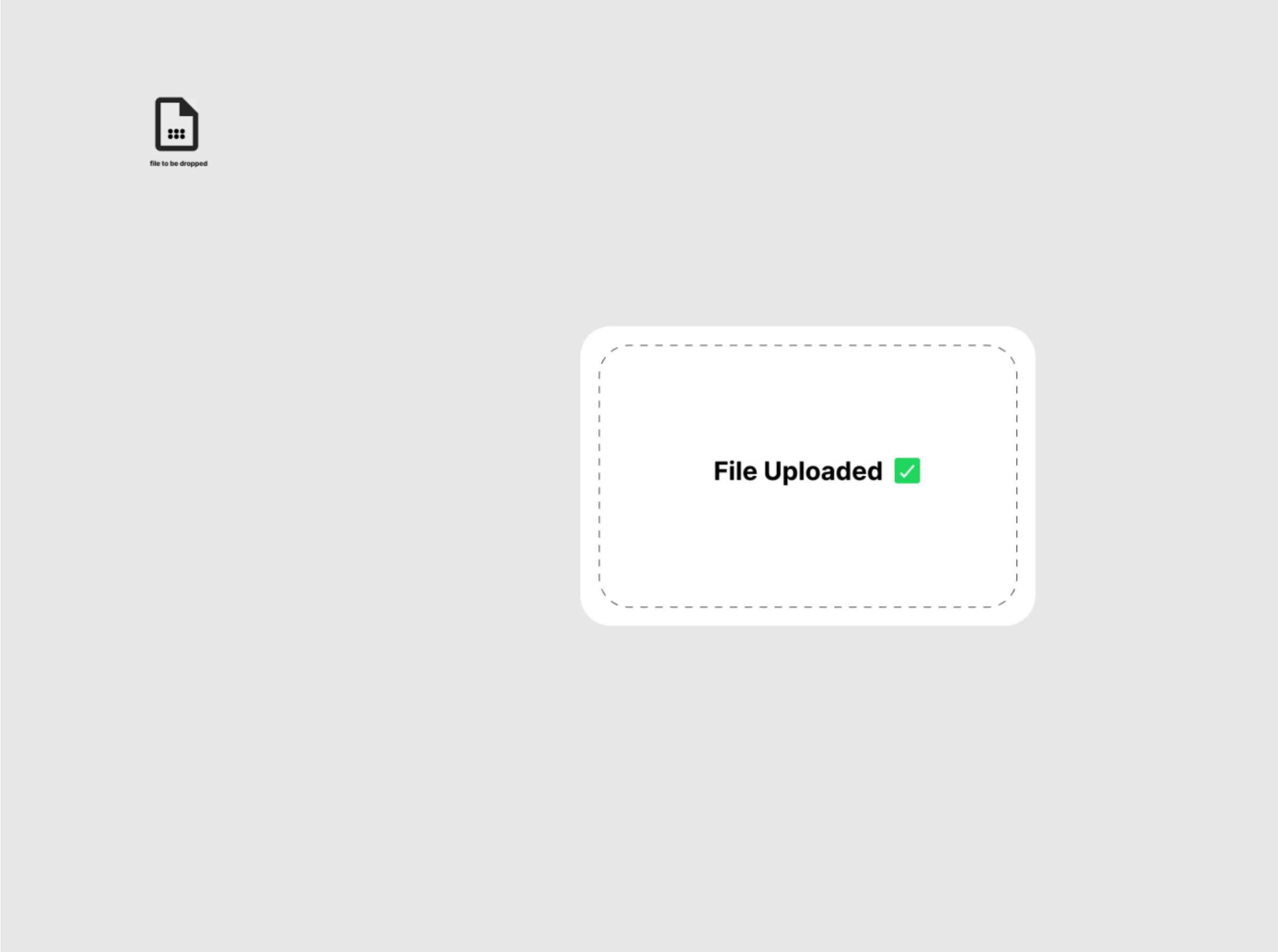
Result:


file to be dropped
file to be dropped


Drop your files here ..
Yes, all the way over here
Uploading...
File Uploaded ✓


file to be dropped
Drop your files here ..


Yes, all the way over here
file to be dropped
Uploading...



File Uploaded ✅

Question:

1. Discuss in detail any one web development framework/technology to implement drag and drop principle.

Ans:

Certainly! One popular web development technology for implementing the drag-and-drop principle is the HTML5 Drag and Drop API. HTML5 provides a native way to enable drag-and-drop interactions in web applications, which is highly flexible and customizable. Let's discuss the HTML5 Drag and Drop API in detail:

HTML5 Drag and Drop API:

The HTML5 Drag and Drop API provides a standard way to make elements on a web page draggable and then allow other elements to accept those dragged items. This API consists of a set of events and attributes that allow developers to create rich and interactive user interfaces.

Here's a breakdown of the key components and concepts involved in the HTML5 Drag and Drop API:

1. Drag Sources:

- These are elements that can be dragged by the user. You can make any HTML element a drag source by setting the `draggable` attribute to `true`.

html

```
<div id="drag-source" draggable="true">Drag me!</div>
```

2. Drop Targets:

- These are elements that can accept the dragged items. You can make an element a drop target by handling specific events and allowing it to receive the dragged data.

html

```
<div id="drop-target">Drop here!</div>
```

3. Events:

- The HTML5 Drag and Drop API relies on several key events for drag-and-drop interactions:

- **`dragstart`**: Fired on a drag source when a drag operation starts. It allows you to set the data being transferred.
- **`dragenter`:** Fired when a draggable element enters a drop target.
- **`dragover`:** Fired when a draggable element is over a drop target. You can use this event to specify whether the drop is allowed.
- **`dragleave`:** Fired when a draggable element leaves a drop target.
- **`drop`:** Fired when a draggable element is dropped onto a drop target. You can retrieve the transferred data here.

4. Data Transfer:

- The `DataTransfer` object is used to pass data between the drag source and drop target. You can set data using the `setData` method during the `dragstart` event and retrieve it during the `drop` event.

```
 javascript
   Setting data during dragstart
  document.getElementById("drag-source").addEventListener("dragstart", function(event) {
      event.dataTransfer.setData("text/plain", "Data to be transferred");
  });

   Retrieving data during drop
  document.getElementById("drop-target").addEventListener("drop", function(event) {
      event.preventDefault();
      const data = event.dataTransfer.getData("text/plain");
       Process the transferred data
  });
```

5. Preventing Defaults:

- To enable drag and drop, you often need to prevent the default behavior of the browser for certain events like `dragover` and `drop` using `event.preventDefault()`.

```
 javascript
   Prevent default behavior to allow dropping
  document.getElementById("drop-target").addEventListener("dragover", function(event) {
      event.preventDefault();
  });
```

6. Styling:

- You can provide visual cues to the user by changing the styling of the drag source and drop target elements using CSS.

```
css
 .dragging {
    opacity: 0.5; / Change opacity when dragging /
 }

 .droppable {
    border: 2px dashed #333; / Highlight the drop target /
 }
```

7. Accessibility:

- Ensure that your drag-and-drop implementation is accessible to all users by providing ARIA roles and labels for screen readers.

The HTML5 Drag and Drop API is a powerful tool for implementing drag-and-drop functionality in web applications. It allows for a wide range of custom interactions, from simple reordering of lists to complex file uploads and more. However, it's important to note that while it's a native HTML5 feature, browser support is generally good but can vary, so you should always check for compatibility with your target audience.

Outcomes:

CO2: Apply principles of Web interface design

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Thus we successfully designed wireframe for drag and drop file.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

1. Wilbert O. Galitz, "The Essential Guide to User Interface Design - An Introduction to GUI Design Principles and Techniques", Wiley Computer Publishing, Second Edition, 2002
2. Bill Scott, Theresa Neil, "Designing Web Interfaces Principles & Patterns for Rich Interaction", O'rielly Media, First Edition, 2009