**Experiment No. 5**

**Title: Drag and Drop and Stay on page principles**

**Batch: B2 Roll No.: 16010421119 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.

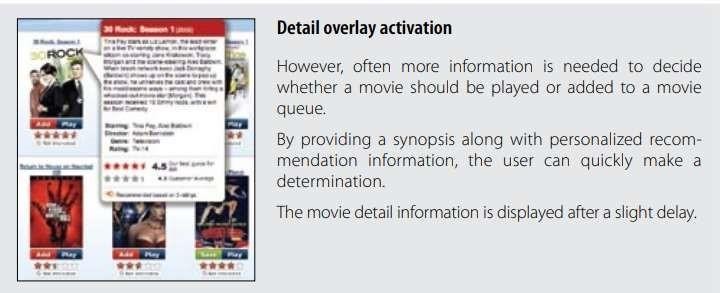
The three specific types of overlays:

* + Dialog Overlays,
  + Detail Overlays, and
  + Input Overlays Dialog Overlay



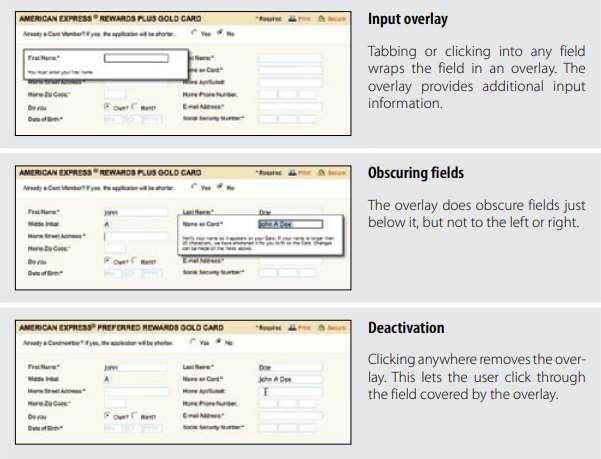
# Detail Overlay

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.



# Input Overlay

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



# Inlays

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

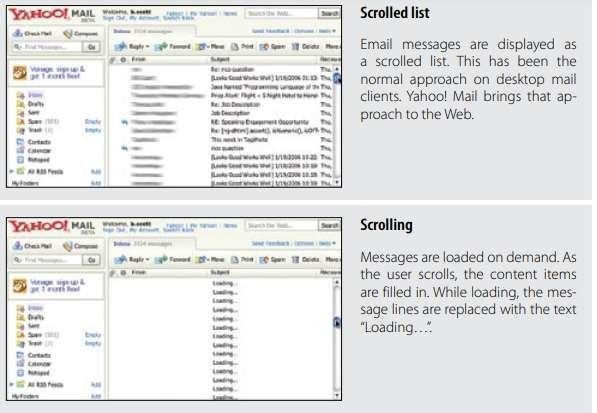
List Inlays

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



Inline Paging



Scrolled paging carousel

Panning and Zoomable Interface

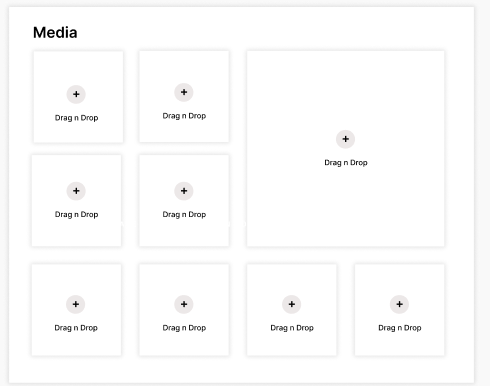


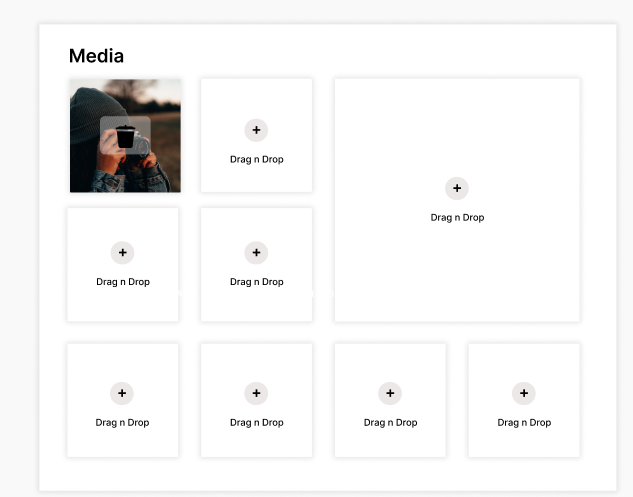
# Procedure:

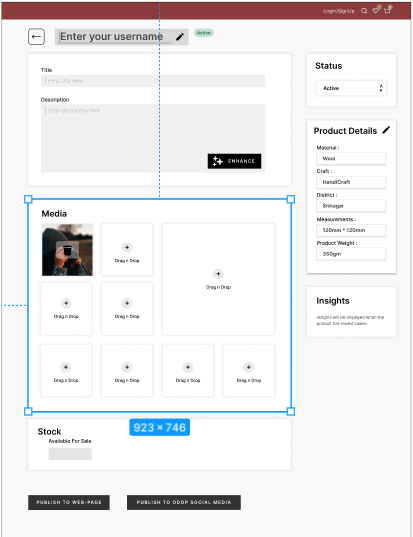
Create wireframes incorporating Drag and Drop and Stay on page principles for chosen topic

# Result:

**Drag and Drop:**







**This is a method to insert images   
considering a very common method of use , we have included this in our app to make it more easy to use.**

**Question:**

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

**Answer:**

One popular web development framework/technology that implements the drag-and-drop principle is HTML5 Drag and Drop, which is a native feature of HTML5 and JavaScript. This technology allows you to create interactive and user-friendly web applications where users can drag elements and drop them into predefined target areas.

# HTML5 Drag and Drop:

* 1. **Overview:** HTML5 Drag and Drop is a web technology that allows you to create dynamic and intuitive user interfaces by enabling drag-and-drop interactions without relying on third-party libraries or plugins. It is based on standard web technologies, primarily HTML, CSS, and JavaScript.

# Key Components:

1. Drag Sources: These are elements or objects that users can pick up and drag. For example, an image, a file, or a div element can be made draggable by setting the draggable attribute to true.
2. Drop Targets: These are areas where users can drop the dragged elements. These targets can be any HTML element, such as a div or a drop zone.

# HTML5 Drag and Drop API:

HTML5 Drag and Drop involves several key events and methods to facilitate the drag-and- drop functionality:

1. dragstart: This event is triggered when a draggable element is first selected for dragging. You can use it to specify the data to be transferred during the drag.
2. dragenter: This event occurs when the draggable element enters a drop target. It allows you to control the appearance of the target element when a valid drag-and-drop operation is detected.
3. dragover: This event is fired as long as the draggable element is over a drop target. It provides the opportunity to control the drop target's behavior, like preventing the default action.
4. dragleave: Triggered when the draggable element leaves a drop target, allowing you to reset the target's appearance.
5. drop: This event occurs when the draggable element is released over a drop target. It lets you access the transferred data and handle the drop operation.
6. dragend: Fired when the drag operation completes. You can use this event to clean up any resources or visual feedback used during the drag-and-drop process.

# Outcomes:

**CO2:** Apply principles of information organization and navigation along with data handling in web interface design

**Conclusion:** Thus, in this experiment we implemented a drag-and-drop feature and also used a stay on page feature in it.

# 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