Module 3

1. **Types of In-Page Editing**

   a. Discussion
   b. Example
   c. Design for the given scenario or application

Single-Field Inline Edit

Editing a single line of text.
The simplest type of In-Page Editing is when editing a single field of text inline.
The editing happens in place instead of in a separate window or on a separate page.

Multi-Field Inline Edit

Editing more complex information.

Overlay Edit

Editing in an overlay panel.

Table Edit

Editing items in a grid.

Group Edit

Changing a group of items directly.
On the iPhone's home screen, the icons are normally locked down. However, there is a way to switch into a special Group Edit mode that allows you to rearrange the icon's positions by drag and drop.

Module Configuration

Configuring settings on a page directly.
Module Configuration is a common pattern on these types of sites. Instead of modifying modules on a separate page, the sites provide ways to directly configure the amount and
type of content that shows in each module.

2. **Challenges associated with In-Page Editing**

   Challenges associated with Single- Field Inline Edit
   ● Discoverability
   ● Accessibility

3. **Guidelines for Choosing Specific Editing Patterns**

   In-Page Edit provides a powerful way to make interfaces direct. Here are some general guidelines to think about when choosing an editing pattern:
   • Whenever you have a single field on the page that needs editing, consider using the **Single-Field Inline Edit**.
   • For multiple fields or more complex editing, use the **Multi-Field Inline Edi**t.
   • If you don't need inline context while editing, or the editing is something that demands the

user's full attention, use **Overlay Edit**.

- For grid editing, follow the pattern **Table Edit**.
- When dealing with multiple items on a page, **Group Edit** provides a way to balance between visual noise and discoverability.
- When providing direct configuring to modules, use the **Module Configuration** pattern.

4. Drag and Drop principle

i. Events
ii. Two common approaches to targeting a drop of the dragged object

a. Discussion
b. Example
c. Design for the given scenario or application

i. Events:

Drag and drop interactions are typically triggered by a series of events, such as a user starting to drag an object, moving the object over a target, and releasing the object to initiate a drop. These events can be detected and handled by the webpage or application using JavaScript or other programming languages.

ii. Two common approaches to targeting a drop of the dragged object:

There are two common approaches for targeting a drop of a dragged object:

Targeted drop:

In a targeted drop, the user must drop the object on a specific target element or area in order for the drop to be successful. This approach can be useful for ensuring that the drop is only accepted in specific locations or contexts.

Free-form drop:

In a free-form drop, the user can drop the object anywhere on the page, and the drop will be accepted as long as the object is within the bounds of the page or application. This approach can be useful for allowing more flexibility in the placement of the object.

5. Drag and Drop List

a. Discussion: Drag and drop lists can be useful for providing a user-friendly and intuitive way for users to rearrange items in a list. This can be particularly useful for lists that are used frequently or that need to be customized by the user.

b. Example: An example of a drag and drop list might be a to-do list application that allows users to rearrange items in the list by dragging and dropping them into a new position. The application might also allow users to group items into different categories or lists by dragging and dropping them into specific areas of the page.

c. Design for the given scenario or application: When designing a drag and drop list for a specific scenario or application, it is important to consider the needs and goals of the user. This might include determining the appropriate level of control and flexibility for rearranging items in the list, as well as any additional features or functionality that might be needed, such as the ability to group items into categories or lists. It is also important to ensure that the drag and drop list is user-friendly and easy to use, as this will help encourage adoption and usage of the feature.

6. Drag and Drop Object

a. Discussion
b. Example
c. Design for the given scenario or application

Drag and drop object is a term that refers to an element or object that can be moved or rearranged on a webpage or application by dragging and dropping it into a new position. Drag and drop objects are often used to allow users to **customize the layout or arrangement of elements on a page or to initiate actions or tasks by dragging and dropping objects**.

Here is an example of how a drag and drop object might work in a practical scenario:
A user starts to drag an object, such as an image or icon, from one location on the page.

The user moves the object over a specific target element or area, such as a folder or container.

When the user releases the object, it is dropped onto the target element or area, initiating an action or task.

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.

7. Fitts Law

a. Discussion

According to Fitts's law, the time required to rapidly move to a target object is a function of the distance to the target and the size of the target.

In other words, the larger and closer the target is, the faster and more accurately it can be selected or activated. Conversely, the smaller and farther away the target is, the slower and less accurately it can be selected or activated.
A target object, in the context of User Interface, can be an interactive element such as a submit button, a hyperlink, and an input field in a web form.

Fitts' law is a binary logarithm.
This means that the predicted results of the usability of an object run along a curve, not a straight line.
In web design, this means that a very small object will become significantly easier to click when given a 20% size increase, while a very large object will not share the same boost in usability when given the same 20% boost in size.

Fitts's law is often used to design user interfaces and web pages that are efficient and easy to use, as it helps to identify the optimal size and placement of target objects to facilitate quick and accurate selection by users. It can also be used to evaluate and compare the usability of different user interface designs.
It is based on the work of psychologist Paul Fitts in 1954
For example, this law influenced the convention of making interactive buttons larger (especially on finger-operated mobile devices) smaller buttons are more difficult (and time-consuming) to click.
The ideal fitts' law application would let us know where the user cursor is when he lands on the application. This point would be called **Prime Pixel**, the point where the user would carry out all his tasks from.
Unfortunately, while our browser and applications(example windows apps, desktop games) can utilize the prime pixel but a website cannot. Even if we determine the prime pixel, it would change every time the user moves the cursor.

$$\mathrm{MT} = \mathrm{a} + \mathrm{b} * log_2(\frac{D}{W} * 2)$$

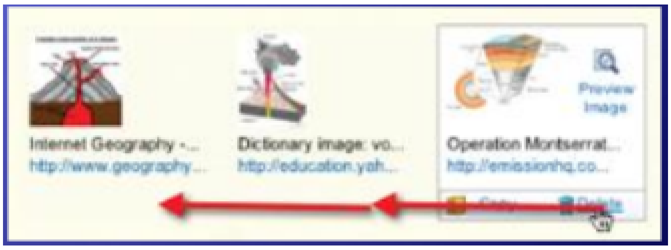
Where:

- MT (Movement Time) is the time required to move to the target object.
- a and b are constants that represent the intercept and slope of the regression line, respectively.
- D is the distance from the starting point to the center of the target object.
- W is the width of the target object.
- The formula can be used to predict the time required to move to a target object based on its size and distance from the starting point. The constants a and b can be determined through experimental testing and are specific to the particular task and user group being studied.

8. Contextual Tools

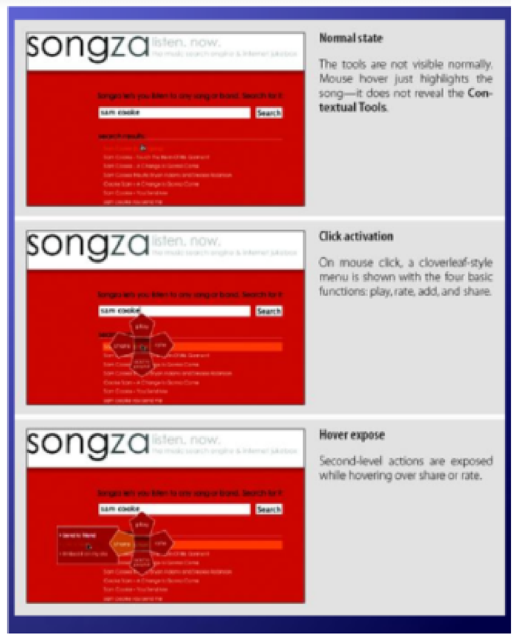
1. **Always-Visible Tools**
2. **Hover-Reveal Tools**
3. **Toggle-Reveal Tools**
4. **Multi-Level Tools**
5. **Secondary Menus**

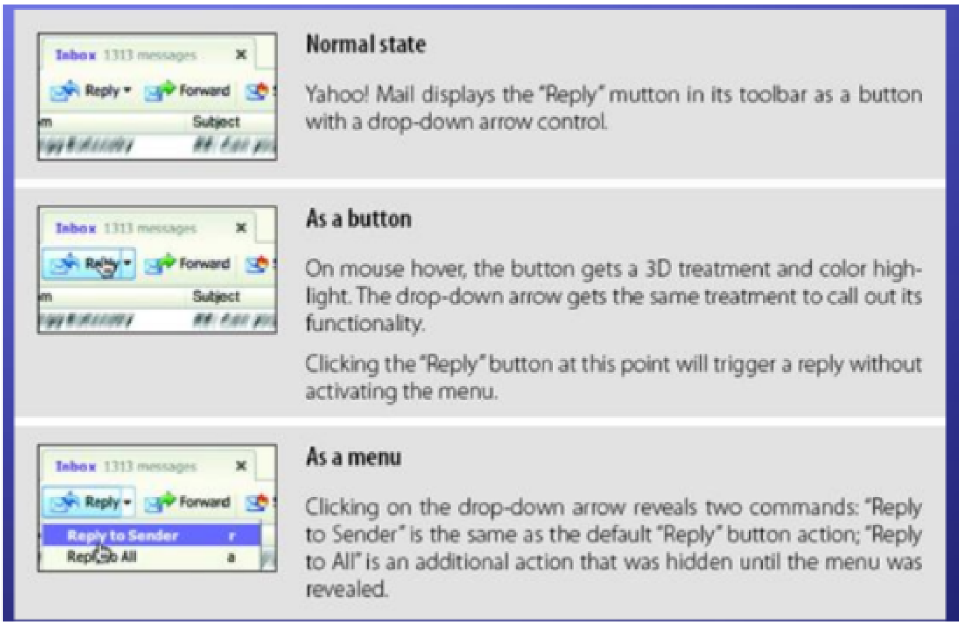
Name of the Contextual Tool Type	Always-Visible Tools
What	These tools are always displayed on the screen, regardless of the task or content being viewed. They are typically located in a fixed location on the interface, such as in a toolbar or menu.
Example	A web-based email application might have a set of tools that are always visible at the top of the screen, such as buttons for composing a new email, searching for emails, and managing the inbox. These tools are always available to the user, regardless of the current context or situation.
Best Practices (If Applicable)	● Make your Contextual Tools always visible if it is important to make a prominent call to action. ● Keep visual clutter to a minimum. ● Keep the number of visual items to a minimum.
Anti Patterns (If Applicable)	-
Apply for the given scenario	

Name of the Contextual Tool Type	Hover-reveal tools
What	These tools are hidden until the user moves their mouse cursor over a specific area of the interface. They are typically used to reveal additional options or information related to the task or content being viewed.
Example	A photo-editing application might have a set of tools that are hidden until the user moves the mouse cursor over a specific area or element, such as a toolbar or menu. When the cursor is moved over the area, the tools are revealed, allowing the user to access them as needed.
Best Practices (If Applicable)	Avoid using overlays when revealing additional tools. They will lead to the Hover and Cover anti-pattern, as well as require the user to perform mouse gymnastics to accomplish the most basic tasks. When additional tools are revealed, make sure that all parts of the page remain stable. Make sure revealed icons are clear and understandable. When possible, just use text labels.
Anti Patterns (If Applicable)	Hover and Cover: -picture gets covered Mystrey Meat: -no idea of what the icons mean when they are revealed on hover
Apply for the given scenario 	

Name of the Contextual Tool Type	Toggle-Reveal Tools
What	These tools are revealed when the user clicks on a specific button or link. They can be used to display additional options or information related to the task or content being viewed.
Example	A document-editing application might have a set of formatting tools that are hidden until the user clicks a specific button or link. When the user clicks the button, the tools are revealed, and the user can click the same button again to hide the tools.
Best Practices (If Applicable)	Toggle a tool mode for an area or page when the actions are not the main flow, but you want to provide the most direct way to act on these objects when the need arises. Make the activation and deactivation of the edit mode as symmetrical as possible.
Anti Patterns (If Applicable)	
Apply for the given scenario	

Name of the Contextual Tool Type	Multi-Level Tools Muttons
What	These tools are hierarchical and allow users to drill down through multiple levels of information or options. They are often used to display options that are relevant to a specific task or content area.
Example	Imagine that you are using a project management application that has a multi-level

	tool for viewing and organizing tasks. When you click on the tool, you are presented with a list of top-level tasks. If you click on one of the top-level tasks, you are taken to a second level of the tool where you can view and edit the details of that specific task. If you click on one of the sub-tasks within the task, you are taken to a third level of the tool where you can view and edit the details of that specific sub-task. This allows you to drill down through multiple levels of tasks and sub-tasks, making it easier to organize and manage your work.
Best Practices (If Applicable)	Use Multi-Level Tools when you want to avoid revealing Contextual Tools on a mouse hover. Use Multi-Level Tools to make activation explicit. Use muttons when you have a default action that the user normally takes but alternate actions are still fairly frequent. Avoid cascades where possible. Users have a hard time maneuvering the various mouse turns that are required to get to these secondary menus.
Anti Patterns (If Applicable)	
Apply for the given scenario  **Normal state** The tools are not visible normally. Mouse hover just highlights the song—it does not reveal the **Contextual Tools**. **Click activation** On mouse click, a cloverleaf-style menu is shown with the four basic functions: play, rate, add, and share. **Hover expose** Second-level actions are exposed while hovering over share or rate.	



Normal state

Yahoo! Mail displays the "Reply" mutton in its toolbar as a button with a drop-down arrow control.

As a button

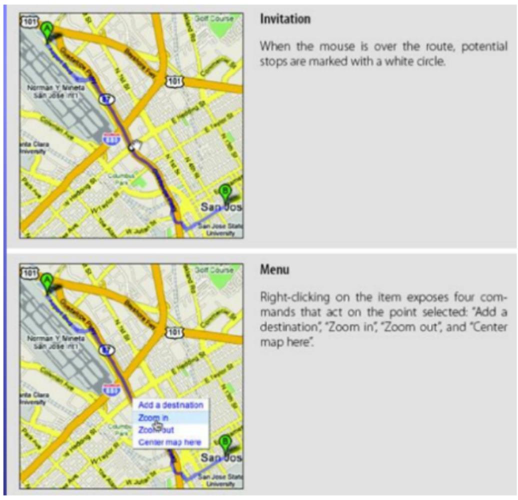
On mouse hover, the button gets a 3D treatment and color highlight. The drop-down arrow gets the same treatment to call out its functionality.

Clicking the "Reply" button at this point will trigger a reply without activating the menu.

As a menu

Clicking on the drop-down arrow reveals two commands: "Reply to Sender" is the same as the default "Reply" button action; "Reply to All" is an additional action that was hidden until the menu was revealed.

Name of the Contextual Tool Type	Secondary Menus
What	These menus are used to display additional options or information related to the task or content being viewed. They are typically accessed by clicking on a specific button or link and can be used to display options that are not immediately relevant or frequently used. Secondary Menus have not been common in web applications.
Example	Imagine that you are using a word processing application and you have selected some text in a document. If you right-click on the selected text, a secondary menu appears with a list of options related to the selected text. These options might include things like copying, cutting, or pasting the text, as well as formatting

	options like bold, italic, or underline
Best Practices (If Applicable)	
Anti Patterns (If Applicable)	
Apply for the given scenario  **Invitation** When the mouse is over the route, potential stops are marked with a white circle. **Menu** Right-clicking on the item exposes four commands that act on the point selected: "Add a destination", "Zoom in", "Zoom out", and "Center map here".	

9. Overlay

1. **Dialog**
2. **Detail**
3. **Input**

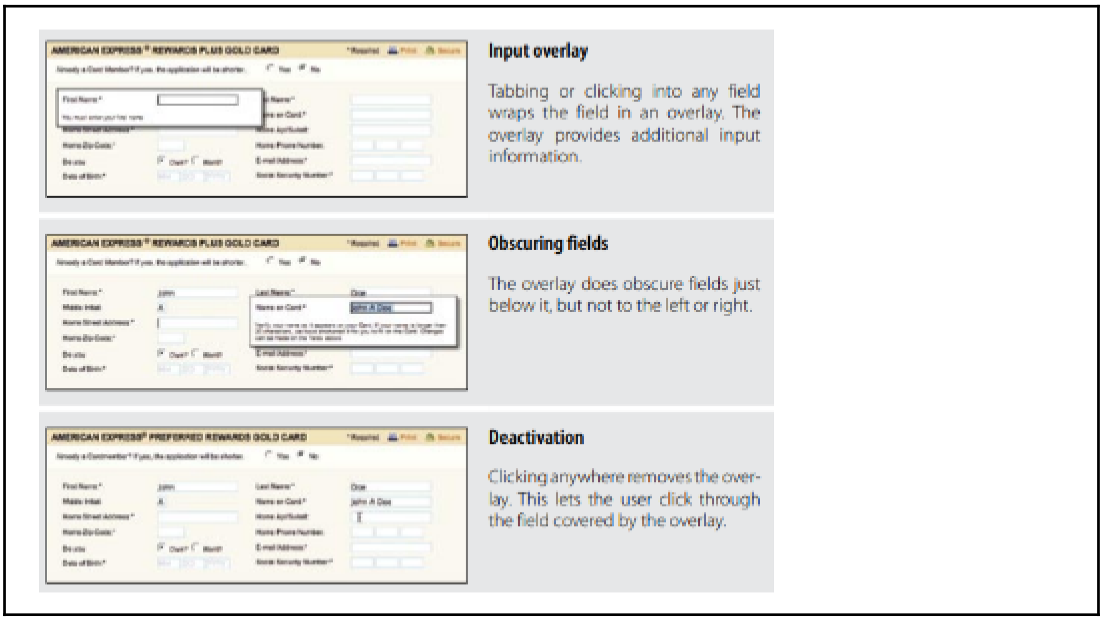
Name of the Overlay Type	Dialog Overlays
What	Dialog Overlays replace the old style browser pop ups.
Example	
Best Practices (If Applicable)	Always use Dialog Overlays instead of browser pop ups. Use the Lightbox Effect when the overlay contains important information, if the user should not ignore it, or if the interaction with the dialog should be modal. Avoid unnecessary Dialog Overlays (Idiot Boxes), as they interrupt

	the user's flow. Don't use an overlay when a simpler, in-page interaction would suffice Avoid JavaScript alert boxes, as they don't provide a consistent user experience between operating systems.
Anti Patterns (If Applicable)	
Apply for the given scenario	

Name of the Overlay Type	Detail Overlay
What	The second type of overlay is somewhat new to web applications. • 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.
Example	Kissasian.li
Best Practices (If Applicable)	Use Detail Overlays to give a sneak peek at detailed information. This will avoid unnecessary page

	transitions. For hover-activated Detail Overlays, provide a slight delay for activation (about half a second). This will avoid the interface behaving like a Mouse Trap. For hover-activated Detail Overlays, provide a simple deactivation (e.g., simple mouse out).
Anti Patterns (If Applicable)	Mouse Traps Non-Symmetrical Activation/Deactivation Needless Fanfare Hover and Cover
Apply for the given scenario	

Name of the Overlay Type	Input Overlay
What	Input Overlay is a lightweight overlay that brings additional input information for each field tabbed into.
Example	
Best Practices (If Applicable)	Use Input Overlays to simplify the visual style of a form. Place additional help in the overlay. • For Input Overlays, make sure the only visual change between the field and the overlay field is intentional (e.g., making input field visually bolder). • For Input Overlays, allow clicking anywhere to deactivate the overlay.
Anti Patterns (If Applicable)	
Apply for the given scenario	



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.

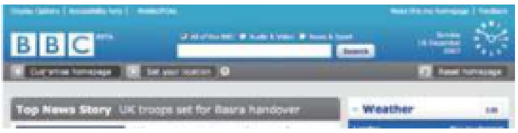
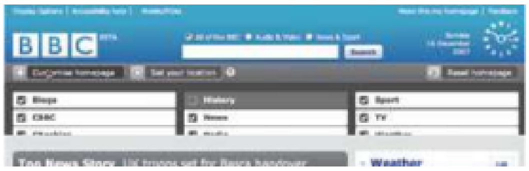
Deactivation

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

10. Inlays

1. **Dialog**
2. **List**
3. **Detail**

Name of the Inlay Type	Dialog
What	A simple technique is to expand a part of the page, revealing a dialog area within the page.
Example	
Best Practices (If Applicable)	Use Dialog Inlays for page customization. It is helpful to tweak the page and see the results at the same time. To smooth the introduction of the Dialog Inlay into the page, use a quick slide in animation. Use Dialog Inlays to connect the dialog with the element it slides out from. Use Dialog Inlays for secondary tools that aren't primary to the main flow of the page.

Anti Patterns (If Applicable)	
Apply for the given scenario  BBC Customise homepage Set your location Reset homepage Top News Story UK troops set for Basra handover Weather  BBC Customise homepage Set your location Reset homepage Blogs History Sport CBBC News TV Top News Story UK troops set for Basra handover Weather	

Name of the Inlay Type	List
What	Lists are a great place to use Inlays. Instead of requiring the user to navigate to a new page for an item's detail or popping up the information in an Overlay. The List Inlay works as an effective way to hide detail until needed.
Example	Accordion
Best Practices (If Applicable)	
Anti Patterns (If Applicable)	
Apply for the given scenario	

Name of the Inlay Type	Detail
What	A common idiom is to provide additional detail about items shown on a page.
Example	Tabs
Best Practices (If Applicable)	Use Detail Inlay to provide additional information in context without hiding other information.

	Use Detail Inlay to avoid the anti-pattern Hover and Cover. Make it easy to dismiss the Detail Inlay
Anti Patterns (If Applicable)	
Apply for the given scenario	

11. Virtual page

a. Type
b. Discussion
c. Example

Virtual Scrolling

The traditional Web is defined by the “page.”

In practically every implementation of websites pagination was the key way to get to additional content.

Of course, websites could preload data and allow the user to scroll through it. However, this process led to long delays in loading the page.

So most sites kept it simple: go fetch 10 items and display them as a page and let the user request the next page of content. Each fetch resulted in a page refresh.

Keep the users informed about where they are. Either use a tool tip or status area to communicate the range of data they are scrolling into.

Give feedback while waiting on data to load.

Create the illusion of an entire loaded virtual space for when the data feels like a data set (e.g., Yahoo! Mail’s mail messages).

Extend the virtual space during scroll for search results (e.g.Microsoft Live Search).

Inline Paging

- Use Inline Paging for material that is naturally “chunked” but for which you still want to create a smooth viewing experience when moving between pages.
- Respect the back button. Make it work for paging (e.g., Gmail).
- Only update the “virtual page” and not the entire page when inline paging.
- Consider progressive loading as a way to page-in more content in a virtual space

Scrolled Paging Carousel

- Besides Virtual Scrolling and Virtual Paging, there is another option.
- You can combine both scrolling and paging into Scrolled Paging.

- Carousels are best for visually distinct contents. Images, CD covers, and movie box shots are all natural items to place in a carousel.
- If the content is highly relevant at the beginning and relevancy drops off, a Carousel is a good solution, since it spotlights the most relevant items.
- If there is a lot of content to display, carousels provide too small of a window and thus can frustrate users when they actually try to find content.
- Placing back and forth arrows next to each other is simpler to operate but somewhat less discoverable.
- Make the content big enough to view easily, providing enough whitespace to make each item distinct.
- Allow a portion of the next item to be partially revealed. This invites the user to scroll the content to see the partial content fully.
- Clicking the left arrow should scroll content in from the left (left to right), and clicking the right arrow should scroll content in from the right (right to left).

Panning

- One way to create a virtual canvas is to allow users the freedom to roam in two-dimensional space.
- A great place for Virtual Panning is on a map.
- Google Maps allows you to pan in any direction by clicking the
- mouse down and dragging the map around.

Zoomable User Interface

- A Zoomable User Interface (ZUI) is another way to create a virtual canvas.
- Unlike panning or flicking through a flat, two-dimensional space, a
- ZUI allows the user to also zoom in to elements on the page.
- This freedom of motion in both 2D and 3D supports the concept of
- an infinite

12. Difference

a. Inlay v/s Overlay

b. Paging v/s Scrolling

13. Static Invitation

a. Type

b. Discussion

c. Example

d. Design for the given application

Static invitations are user interface elements that invite the user to perform a specific action or interact with a particular element on the page, but unlike dynamic invitations, static invitations do not change in response to the user's actions or context.

There are two broad patterns of static invitations:

Call to action (CTA) invitation:

- This type of invitation is a specific type of static invitation that is designed to encourage the user to take a specific action, such as filling out a form, purchasing a product, or subscribing to a newsletter. CTAs are often prominently displayed and use action-oriented language and design elements to grab the user's attention and motivate them to take the desired action.
- Call to Action (CTA) Invitations are generally provided as static instructions on the page. But visually they can be provided in many different ways.
- CTA stands for call to action, and it's the part of a webpage,advertisement, or piece of content that encourages the audience to do something.
- In marketing, CTAs help a business convert a visitor, or reader into a lead for the sales team. CTAs can drive a variety of different actions depending on the content's goal.

Tour invitation:

- This type of invitation invites the user to take a guided tour of a website or application, typically by highlighting specific elements or features and providing additional information or guidance as the user progresses through the tour. Tour invitations are often used to introduce new users to a product or to provide a quick overview of key features.
- Use Tour Invitations when you have a newly redesigned site or are launching a new site and need to take the user through a series of features.
- Integrate Tour Invitations with the live site as much as possible.
- Make Tour Invitations short and sweet, easy to exit, and clear to restart.
- Don’t depend on tours to fix interface issues.
- Keep tours simple.

14. Dynamic Invitation

a. Type
b. Discussion
c. Example
d. Design for the given application

There are several ways to engage a user with a dynamic invitation, which is a type of user interface element that invites the user to perform a specific action or interact with a particular element on the page.

Hover invitation:

This type of invitation appears when the user moves the cursor over a specific element on the page.

Affordance invitation:

This type of invitation is based on the principle of affordance, which refers to the inherent properties of an object that suggest how it should be used. For example, a button might have a affordance invitation if it appears to be pressable.

Drag and drop invitation:

This type of invitation invites the user to drag and drop a specific element on the page, typically by displaying a cursor or other visual indicator when the user hovers over the element.

Inference invitation:

This type of invitation relies on the user's past behavior or other contextual information to infer what the user might want to do next and invites them to perform that action.

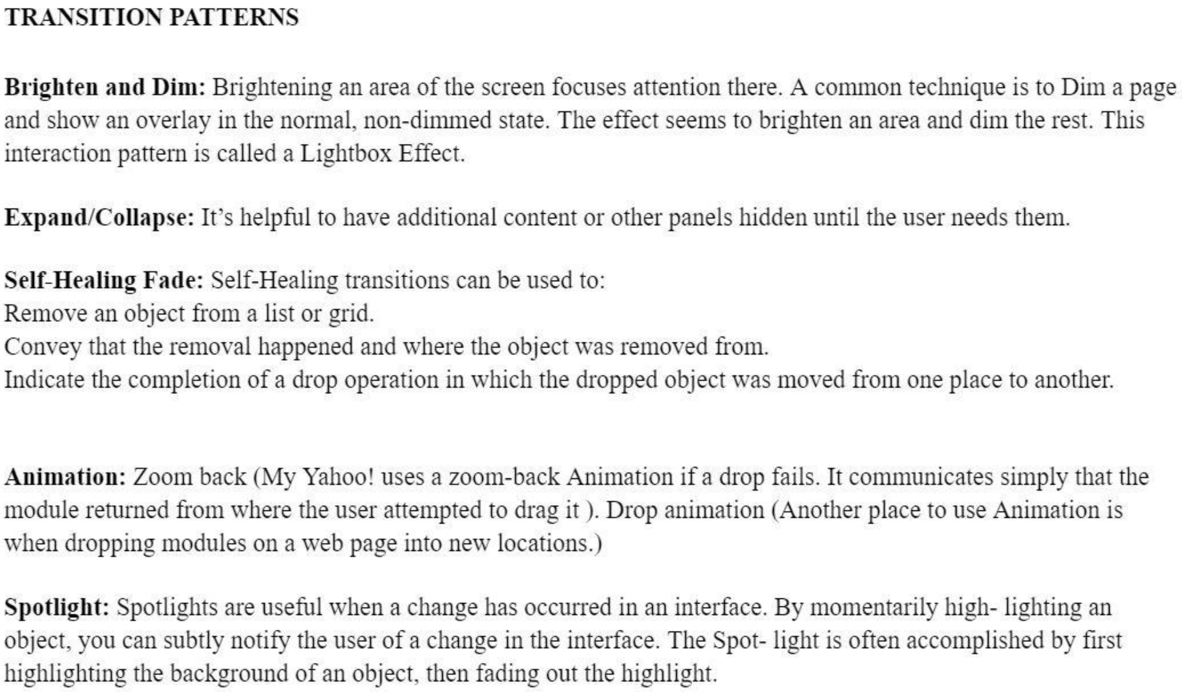
More content invitation:

This type of invitation invites the user to view additional content, such as by clicking on a "read more" link or scrolling down to reveal more content on the page

15. Transition Patterns

1. Brighten and Dim
2. Expand/Collapse
3. Animation
4. Spotlight

a. Discussion
b. Example
BE SAS



TRANSITION PATTERNS

Brighten and Dim: Brightening an area of the screen focuses attention there. A common technique is to Dim a page and show an overlay in the normal, non-dimmed state. The effect seems to brighten an area and dim the rest. This interaction pattern is called a Lightbox Effect.

Expand/Collapse: It's helpful to have additional content or other panels hidden until the user needs them.

Self-Healing Fade: Self-Healing transitions can be used to:
Remove an object from a list or grid.
Convey that the removal happened and where the object was removed from.
Indicate the completion of a drop operation in which the dropped object was moved from one place to another.

Animation: Zoom back (My Yahoo! uses a zoom-back Animation if a drop fails. It communicates simply that the module returned from where the user attempted to drag it). Drop animation (Another place to use Animation is when dropping modules on a web page into new locations.)

Spotlight: Spotlights are useful when a change has occurred in an interface. By momentarily high- lighting an object, you can subtly notify the user of a change in the interface. The Spot- light is often accomplished by first highlighting the background of an object, then fading out the highlight.

16. Purpose of Transition

Transitions are visual or interactive effects that are used to indicate a change in the user interface, such as when a user navigates to a new page, expands or collapses an element, or performs some other action that affects the layout or content of the page. The purpose of transitions is to communicate these changes to the user and help them understand and orient themselves within the user interface.

Transitions can serve a number of different purposes, including:
Guiding the user's attention: Transitions can help draw the user's attention to specific elements or areas on the page and highlight changes or updates.

Indicating cause and effect: Transitions can help the user understand the relationship between their actions and the resulting changes in the user interface, making it easier for them to anticipate and understand what will happen next.

Enhancing the user experience: Transitions can add visual interest and improve the overall aesthetic of the user interface, making it more enjoyable and engaging for the user.