Ex.no 2: DESIGN A DROP-DOWN LIST OR A MENU IN A GUI KEEPING IN VIEW THE SERIAL POSITION EFFECT

Aim: To design a drop-down list or a menu in a GUI keeping in view the serial position effect.

PROCEDURE:

To execute HTML code, you need a web browser. Follow the steps below to run HTML code on your computer:

1.Open a text editor or an integrated development environment (IDE) such as Notepad, Sublime Text, Visual Studio Code, or any other editor of your choice.

2.Copy the HTML code into the editor and save the file with a .html extension. For example, you can save it as "index.html".

3.Open the saved HTML file with a web browser. You can do this by double-clicking the file, or you can right-click on the file, select "Open with," and choose a web browser from the list.

4.The web browser will render and display the HTML code, executing any scripts or displaying the content as intended.

PROCEDURE- TO DESIGN A DROP-DOWN LIST OR A MENU IN A GUI KEEPING IN VIEW THE SERIAL POSITION EFFECT

1.Define User Goals and Context:

- Understand the specific goals and tasks users aim to accomplish with the drop-down list or menu.
- Identify the context in which the GUI will be used, including the target audience, platform, and any relevant constraints or requirements.

2.Conduct User Research:

- Gather user insights through interviews, surveys, or observations to understand user preferences, needs, and expectations related to drop-down lists or menus.
- Focus on understanding how users perceive and interact with lists, their familiarity with GUIs, and any challenges they face.

3.Identify Design Principles:

- Review existing research and design principles related to drop-down lists, menus, and the serial position effect.
- Determine which design principles are relevant to your specific context and align with the user goals.
- Define the Content:
- Determine the specific options or items that will be included in the drop-down list or menu.
- Consider the information architecture and categorize options if necessary.

4.Design the Visual Representation:

- Decide on the visual representation of the drop-down list or menu. This may include a traditional drop-down, cascading menu, or any other suitable design based on the context and platform.

- Ensure that the visual representation is consistent with the overall GUI design and follows established UX/UI standards.

5.Order and Group Options:

- Apply the serial position effect by ordering the options in a logical manner, such as alphabetical or numerical order.
- If the list is extensive, consider grouping options into relevant categories to aid user comprehension and ease of navigation.

6.Prototype and Test:

- Create a low-fidelity or high-fidelity prototype of the GUI, including the drop-down list or menu.
- Conduct usability testing sessions with representative users, asking them to perform tasks that involve interacting with the drop-down list or menu.
- Observe and collect feedback on the ease of use, efficiency, and user satisfaction.
- Iterate and refine the design based on user feedback and observations.

7.Implement and Evaluate:

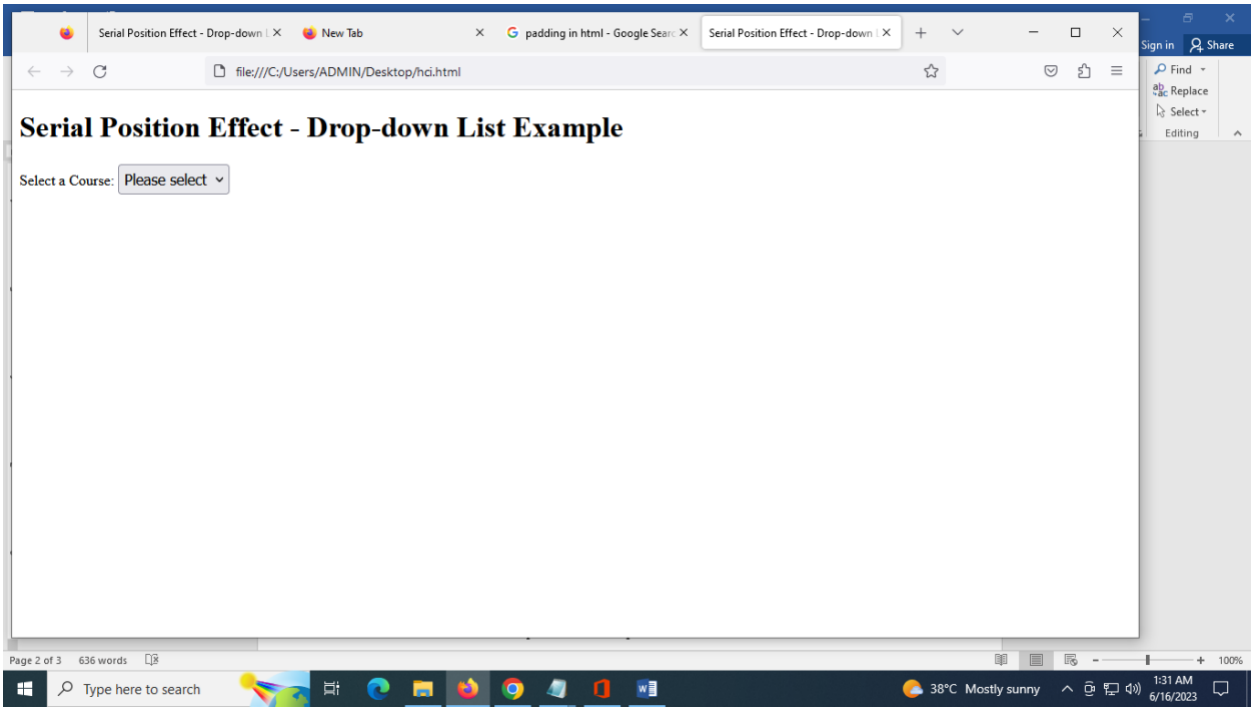
- Once the design has been refined and validated through testing, implement the drop-down list or menu in the final GUI.
- Continuously evaluate the performance and user experience of the GUI in real-world usage.
- Collect feedback from users and monitor metrics to identify areas for improvement.

CODING:

```
<!DOCTYPE html>
<html>
<head>
  <title>Serial Position Effect - Drop-down List</title>
  <style>
    /* Basic styling for the drop-down list */
    select {
      padding: 5px;
      font-size: 16px;
    }
  </style>
</head>
<body>
  <h1>Serial Position Effect - Drop-down List Example</h1>
```

```
  <label for="options">Select a Course:</label>
  <select id="options">
    <option value="" selected disabled hidden>Please select</option>
    <option value="option1">CSE</option>
    <option value="option2">EEE</option>
    <option value="option3">IT</option>
    <option value="option4">MECH</option>
    <option value="option5">CIVIL</option>
  </select>

  <script>
    // Add event listener to capture user selection
    var dropDown = document.getElementById("options");
    dropDown.addEventListener("change", function() {
      var selectedOption = dropDown.value;
      console.log("Selected option: " + selectedOption);
      // You can perform any desired actions based on the selected option here
    });
  </script>
</body>
</html>
```



Result: Thus a drop-down list with serial position effect was designed successfully.