

Assignment 7

Task 1

a) Interaction Technique: Virtual Joystick vs. Swipe Gesture for Mobile Game Control

Problem Solved: In mobile gaming, intuitive and responsive controls are crucial for an enjoyable user experience. Traditional on-screen buttons can obscure the game view and are often less responsive. Virtual joysticks and swipe gestures are alternative interaction techniques that aim to provide more seamless and immersive control options for players.

What It Makes Possible:

- **Virtual Joystick:** Allows players to control the character by simulating a physical joystick on the touch screen. This can offer precise control with a familiar interface for those used to console gaming.
- **Swipe Gesture:** Enables players to control the character by swiping in different directions on the screen. This method can reduce screen clutter and offer a more fluid and less obstructive control scheme.

b)

Research Question: How does the performance and user satisfaction of virtual joystick controls compare to swipe gesture controls in a mobile game?

Variants Compared:

1. **Virtual Joystick Control:** An on-screen joystick that users drag to move the character.
2. **Swipe Gesture Control:** Swiping in different directions on the screen to move the character.

c)

Hypothesis: Players will experience higher accuracy and greater user satisfaction with the virtual joystick compared to the swipe gesture control in a mobile game.

Minimal Prototype Description: Develop a simple mobile game where a character navigates through a maze. Implement two control schemes:

1. **Virtual Joystick:** An on-screen joystick appears in the bottom-left corner of the screen. The player drags the joystick to move the character.
2. **Swipe Gesture:** The player swipes in the desired direction on the screen to move the character.

Data Collection:

- **Performance Metrics:**
 - Time taken to complete the maze.
 - Number of collisions with maze walls.

- User Satisfaction Metrics:
 - Post-game survey using a Likert scale to rate ease of use, enjoyment, and preference between the two control schemes.

Task 2

a) AJAX Model Process and Advantages

Process:

1. User Action: A user interacts with a web page, like clicking a button or entering text.
2. JavaScript Initiates Request: JavaScript code on the page creates an XMLHttpRequest object and sends it to the server with specific data (if needed).
3. Asynchronous Communication: The request is sent in the background, without reloading the entire page. The user can continue interacting with the current content.
4. Server Processes Request: The server receives the request, processes it (like fetching data from a database), and prepares a response.
5. Server Sends Response: The server sends the response data back to the web page.
6. JavaScript Handles Response: The JavaScript code receives the response data and dynamically updates the relevant part of the web page without reloading.

Advantages over Classic Model (Full Page Reloads):

1. Faster and Smoother User Experience: Since the entire page doesn't reload, the interaction feels faster and more responsive. Users can stay on the same page and continue their task without interruption.
2. Reduced Bandwidth Usage: Only the requested data is transferred between the server and browser, saving bandwidth and improving performance, especially for slower internet connections.

b)

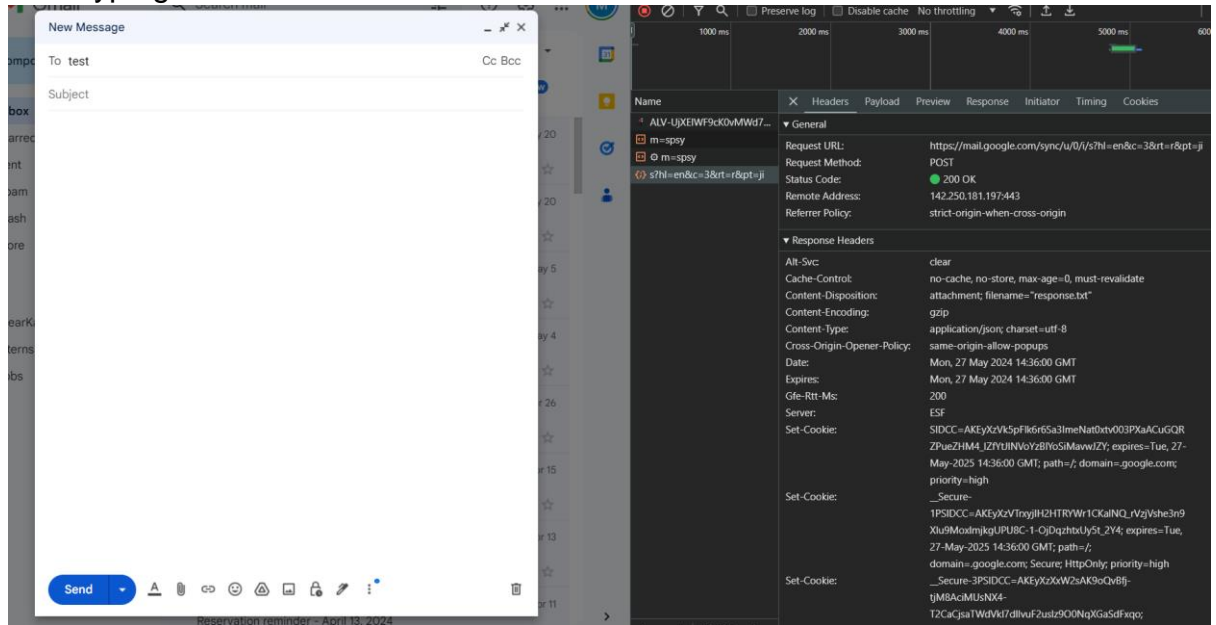
Website: Gmail (<https://mail.google.com/mail/>)

Action: Composing a new email

Data Reloaded: List of contacts (autocomplete suggestions) as we type in the "To" field.

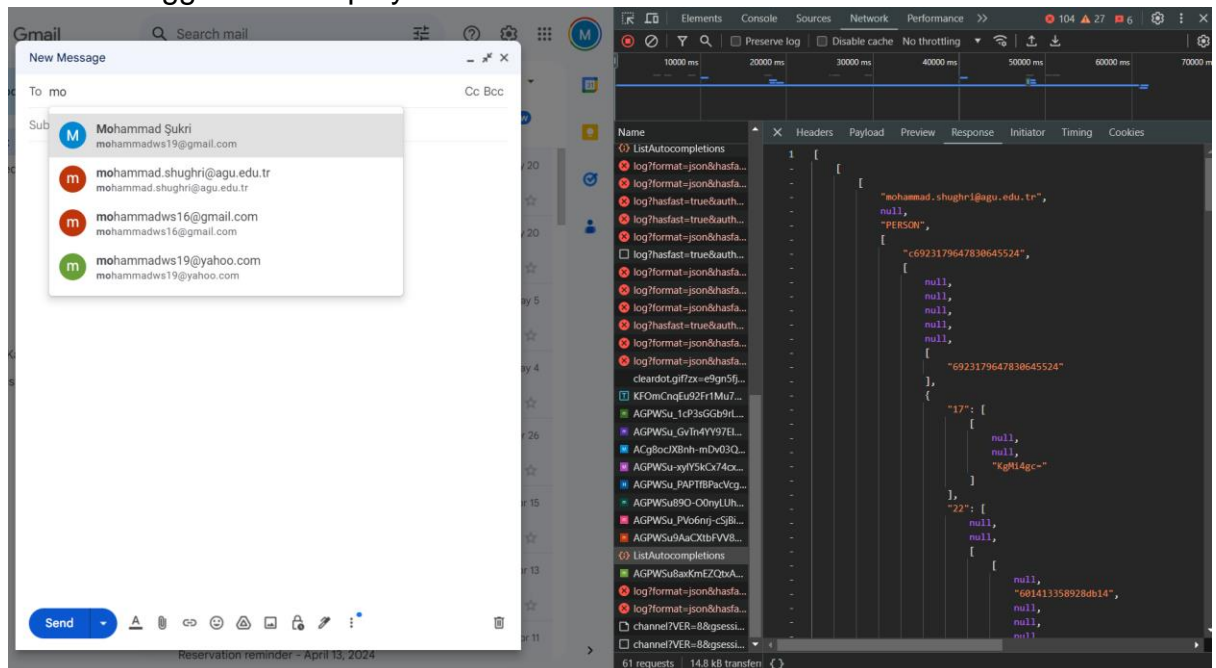
Screenshots:

1. Start typing in the "To" field:



2. AJAX request fetches contacts:

3. Contact suggestions displayed:



Explanation:

As we start typing in the recipient's name in the "To" field, Gmail uses AJAX to send a request to the server in the background. The request likely includes the partially typed email address. The server retrieves matching contacts from the address book and sends them back to the browser. The JavaScript code then parses the response data and dynamically displays a list of suggested contacts in a dropdown menu, all without reloading the entire Gmail page.

Task 3

a) Upon investigating the Leibniz University homepage using browser dev tools, the layout changes significantly at specific breakpoints, commonly around:

- 864<x pixels (px)
 - 863>x pixels (px)
 - 647>x pixels (px)
1. 864px and above (Desktop View):
 - The layout is wide, with multiple columns. The navigation menu is fully displayed horizontally at the top.
 - Images and text blocks are arranged in a grid-like structure.
 - Sidebar elements, if present, are visible.
 2. 863px (Tablet View):
 - The layout switches to fewer columns. The navigation menu may collapse into a hamburger menu.
 - Images and text blocks are stacked more vertically.
 - Sidebar elements are often repositioned below the main content or hidden.
 3. Below 647px (Mobile View):
 - The layout becomes single-column. The navigation menu collapses into a hamburger menu.
 - Images are resized to fit within the single column.
 - Text blocks are stacked vertically.
 - Sidebar elements are either hidden or positioned below the main content.

b)

A class called (.c-section) is being used in order to react to the changes made during responsiveness. For example,

```
.c-section {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
}
```

the content section (.c-section) uses display: flex with a row direction, allowing items to be displayed in a horizontal layout and spaced out with justify-content: space-between.

Also, these layout changes are managed through CSS media queries. For example:

```
@media (max-width: 1023px) {  
  .c-section {  
    flex-direction: column;  
    align-items: center;  
  }  
}
```

```
@media (max-width: 768px) {  
  .c-section {  
    display: block;  
    width: 100%;  
  }  
}
```

@media is a CSS at-rule used for applying styles based on specific conditions, such as the width of the viewport. Media queries allow you to create responsive designs that adapt to different screen sizes and orientations. Here is a breakdown:

By examining the CSS class .c-section and the associated media queries, we can see how the responsive design is implemented to ensure usability across different devices. This approach ensures a seamless experience regardless of screen size.