

---

---

# Designing for accessibility and mobile devices

— By Engr. Dr. Sidra Sultana —

---

---

# HCI Outline

1. Introduction to HCI
2. Basic principles and guidelines of HCI
3. User-centered design and usability testing
4. Designing Effective User Interfaces
5. User interface design principles and guidelines
6. User interface prototyping
7. Prototyping through Wireframes

## 8. Designing for accessibility and mobile devices

# Designing for accessibility and mobile devices

## Outline

- What is **Accessibility**?
- Why accessibility matters: ethical, legal, and usability perspectives
- Real-world motivation:
  - 1 in 6 people worldwide live with some form of disability
  - Accessibility improves usability for everyone (situational impairments, older adults)
- Connection to **Universal Design** and **Inclusive Design**

## Discussion Prompt:

“Think about a time when you couldn’t use an app easily — what caused that difficulty?”

# Accessible Designs for everyone



Cognitive &  
Learning  
Disabilities



Blindness  
Low Vision  
Color-blindness



Speech Inputs



Hearing  
Impairment



Motor &  
Dexterity

Interaction Design Foundation



[interaction-design.org](https://www.interaction-design.org)

# Accessibility Principles

**POUR principles of web accessibility defined by WCAG (Web Content Accessibility Guidelines)**

**Perceivable:** Information must be visible or audible (e.g., alt text for images), so that individuals with sensory impairments can understand the information being conveyed

**Operable:** Interface must be usable via multiple input modes (keyboard, voice, touch), navigate to information via multiple methods (not only the mouse)

**Understandable:** Clear navigation, predictable layouts, consistent feedback and enough understandable so that all different learning styles can engage

**Robust:** Works well with assistive technologies (screen readers, voice control). Technology should be compatible with a user's desired devices, applications or system preferences

# POUR Principles

The Four Principles of Web Accessibility  
as defined by the WCAG 2.0 Guidelines



## Perceivable

Users can recognize the presented information via sight, hearing, or touch.



## Operable

Users can navigate and operate the user interface via alternative input methods.



## Understandable

Users can make sense of the textual, visual, and audio content and available operations.



## Robust

A wide variety of web browsers and assistive technologies can interpret the information.

# Examples

**Perceivable:** Netflix provides subtitles and audio descriptions.

**Operable:** Gmail keyboard shortcuts and focus indicators.

**Understandable:** Google Forms error feedback.

**Robust:** Websites that work with NVDA or VoiceOver.

# Activity

Identify which accessibility principle each of these violations breaks:

1. Button labeled “Click here” (no context)
2. Low contrast text on colored background
3. Hover-only menus (no keyboard access)

# Accessibility Design Techniques

## Design Practices

- Use **semantic HTML** (<button>, <label>, <header>)
- Provide **text alternatives** (alt, aria-label)
- Ensure **keyboard navigation** and visible focus indicators
- Maintain **color contrast** (minimum 4.5:1)
- Avoid flashing animations
- Use **scalable text** (em/rem units)

# What is a Contrast Ratio

A **contrast ratio** measures the difference in brightness (luminance) between two colors — usually text and its background.

It's expressed as **a ratio like 4.5:1, 7:1, or 3:1**, where:

- **1:1** = no contrast (e.g., white text on white background)
- **21:1** = maximum contrast (black on white)

# Mobile Design Constraints & Principles

## Mobile Design Challenges

1. Small Screen Size - Limited screen space
2. Touch-Based Interaction
3. Variable Lighting and Glare
4. Context of Use (outdoors, one-handed, glare)
5. Limited Hardware Controls
6. Dynamic Screen Orientation
7. Complex Gestures and Motion Sensitivity
8. Speech and Audio Dependence
9. Network and Performance Constraints (Variable network speeds)
10. Accessibility Tool Compatibility

# 1. Small Screen Size

- **Problem:** Limited display area makes it difficult for users with **low vision** or **motor impairments** to select small elements accurately.
- **Design impact:**
  - Buttons or links may be too small to tap.
  - Important content may be hidden or require scrolling.
- **Best practice:** Use **large touch targets (at least 48×48 dp)** and prioritize essential content.

## 2. Touch-Based Interaction

- **Problem:** Users with missing limbs, or limited dexterity may struggle with **gestures** or **precise taps**.
- **Design impact:**
  - Difficult to perform gestures like pinch, swipe, or long-press.
- **Best practice:**
  - Provide **alternative input options** (e.g., voice commands or external switches).
  - Avoid gesture-only interactions.

### 3. Variable Lighting and Glare

- **Problem:** Outdoor use or bright environments can cause **low visibility** and poor contrast.
- **Design impact:**
  - Text and icons may be unreadable in sunlight.
- **Best practice:**
  - Ensure **high color contrast ( $\geq 4.5:1$ )** and support **dark/light modes**.

## 4. Context of Use

- **Problem:** Users often interact with mobile devices while **moving, multitasking, or one-handed**.
- **Design impact:**
  - Divided attention, increased error rate.
- **Best practice:**
  - Simplify navigation, reduce cognitive load, and support quick actions.

## 5. Limited Hardware Controls

- **Problem:** Unlike desktop environments, mobile devices rely on **touch and limited hardware buttons**.
- **Design impact:**
  - Difficult for users who depend on **keyboard or assistive input** devices.
- **Best practice:**
  - Support **screen readers (VoiceOver, TalkBack)** and **external keyboards**.

## 6. Dynamic Screen Orientation

- **Problem:** Users who lock orientation (e.g., for stability or accessibility) may miss content optimized for one layout.
- **Best practice:**
  - Ensure layouts adapt to both **portrait and landscape** without loss of functionality.

## 7. Complex Gestures and Motion Sensitivity

- **Problem:** Some animations or motion effects can cause **dizziness** or **disorientation**.
- **Best practice:**
  - Respect **system settings** like “Reduce Motion.”
  - Offer ways to disable complex animations.

## 8. Speech and Audio Dependence

- **Problem:** Users with hearing impairments can't rely on **audio feedback** (alerts, notifications).
- **Best practice:**
  - Provide **visual cues** (e.g., vibration + text alert).
  - Add **captions** for multimedia content.

## 9. Network and Performance Constraints

- **Problem:** Slow or unstable mobile networks affect the **timing of audio, video, and assistive feedback.**
- **Best practice:**
  - Design lightweight pages and ensure core functions work **offline** or with **low bandwidth.**

# 10. Accessibility Tool Compatibility

- **Problem:** Some custom UI components are not recognized by **mobile screen readers**.
- **Best practice:**
  - Use **semantic controls** and test with **TalkBack (Android)** and **VoiceOver (iOS)**.
  - Label elements with **ARIA attributes (Accessible Rich Internet Applications)** when necessary.



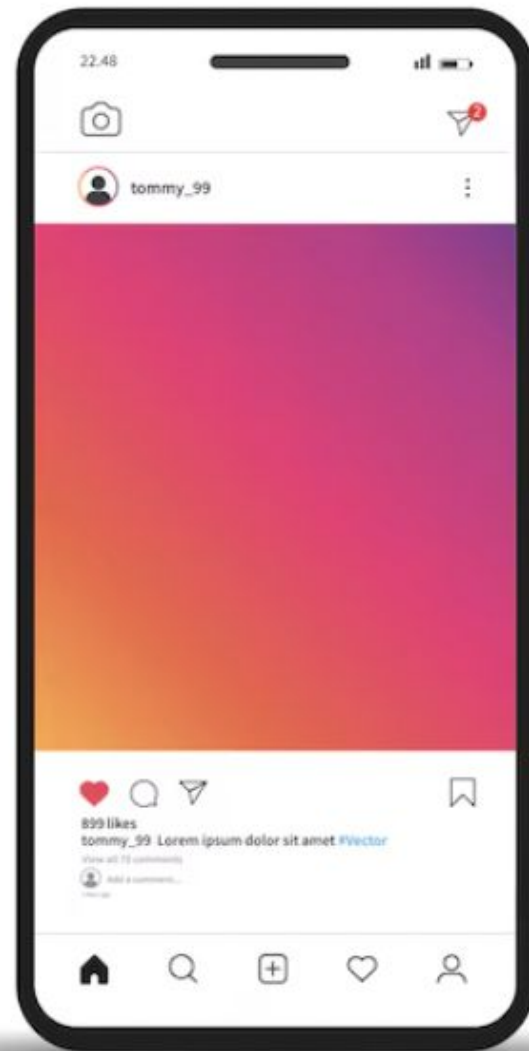
# Mobile Design Principles

1. Simplify Navigation
2. Prioritize Content
3. Use Large, Accessible Touch Targets
4. Ensure Readability
5. Optimize for Different Screen Sizes
6. Minimize User Effort
7. Provide Feedback and Visibility
8. Support One-Handed Use
9. Design for Accessibility
10. Optimize Performance
11. Design for Context of Use
12. Test on Real Devices

# 1. Simplify Navigation

- Keep navigation **clear, shallow, and minimal**.
- Use **bottom navigation bars** or **hamburger menus** for compactness.
- Always provide a visible “**Back**” or “**Home**” option.
- Avoid deep menu hierarchies — aim for 3 clicks or fewer to reach content.

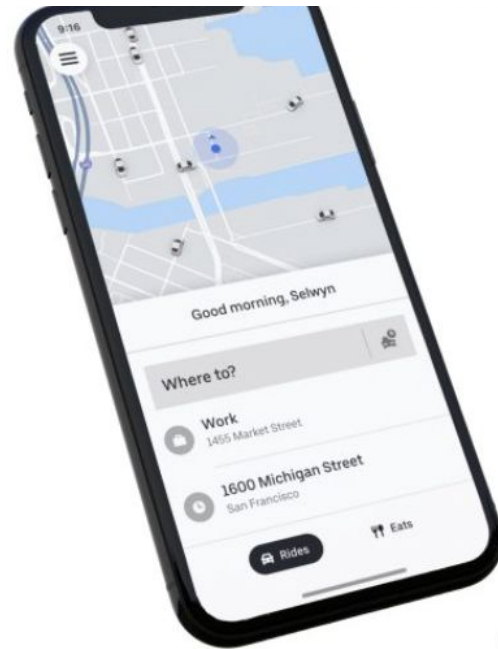
*Example:* Instagram uses a 5-icon bottom bar for key tasks



## 2. Prioritize Content

- Show **important information first** — design for quick scanning.
- Use **progressive disclosure** (show more details only when needed).
- Avoid clutter; focus on **one primary action per screen**.

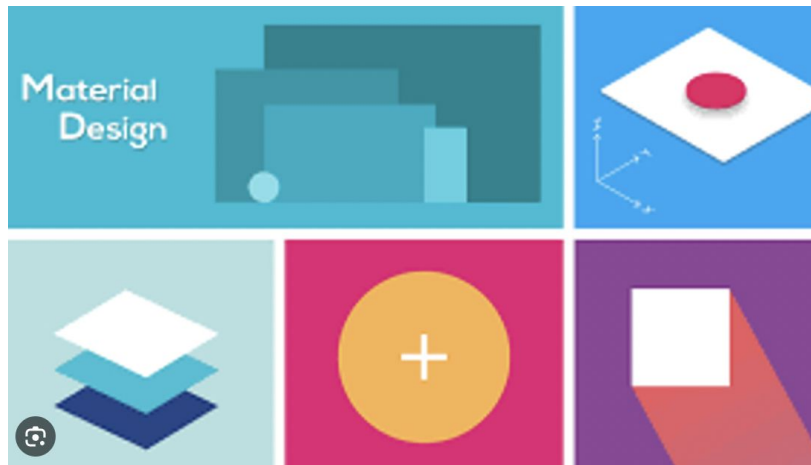
*Example:* Uber's home screen focuses only on "Where to?" input.



### 3. Use Large, Accessible Touch Targets

- Touch targets should be at least **48×48 dp** ( $\approx 9$  mm).
- Provide **enough spacing (8 dp)** between tappable elements.
- Avoid small icons or closely packed links.

*Example:* Google Material Design follows 48 dp target areas for all buttons.

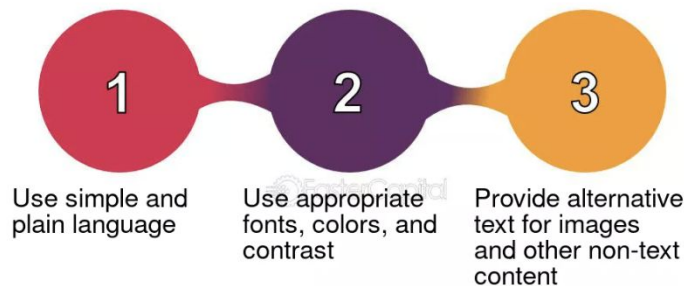


## 4. Ensure Readability

- Use **legible fonts** and **adequate text size** ( $\geq 16\text{sp}$  recommended).
- Maintain **contrast ratio**  $\geq 4.5:1$  between text and background.
- Avoid long paragraphs; use short sentences and clear headings.

*Example:* iOS Human Interface Guidelines recommend a minimum font size of 17 pt for body text.

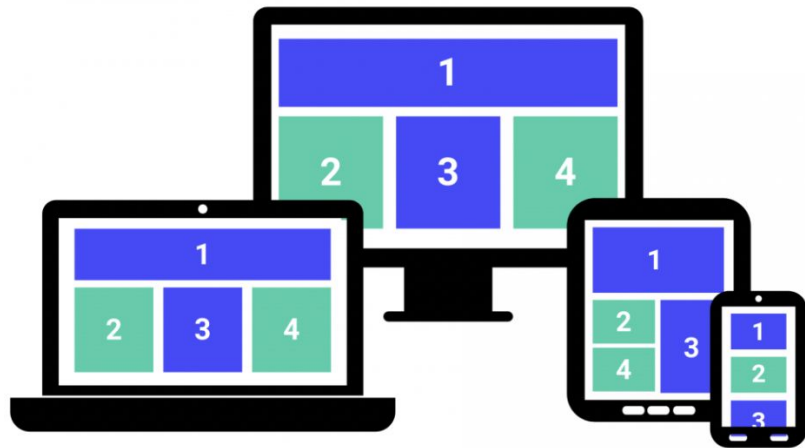
### Ensuring Readability and Clarity



## 5. Optimize for Different Screen Sizes

- Design **responsive layouts** that adapt to screen orientations and resolutions.
- Ensure that rotating the device (portrait ↔ landscape) doesn't break the layout.
- Test across multiple devices (small phones, large phones, tablets).

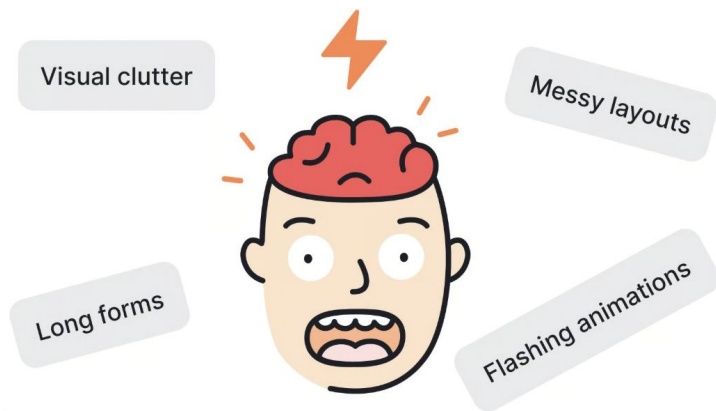
*Example:* YouTube's video player automatically adjusts layout when rotated.



## 6. Minimize User Effort

- Reduce typing and repeated actions.
- Use **auto-complete**, **voice input**, and **default suggestions**.
- Remember user preferences to avoid re-entry of data.

*Example:* Google Maps saves recent searches for faster reuse.



## 7. Provide Feedback and Visibility

- Users should know when an action is recognized.
- Provide visual or haptic feedback (button highlight, vibration, progress spinner).
- Always indicate system status (loading indicators, success messages).

*Example:* “Pull to refresh” animations in Twitter show instant visual feedback.

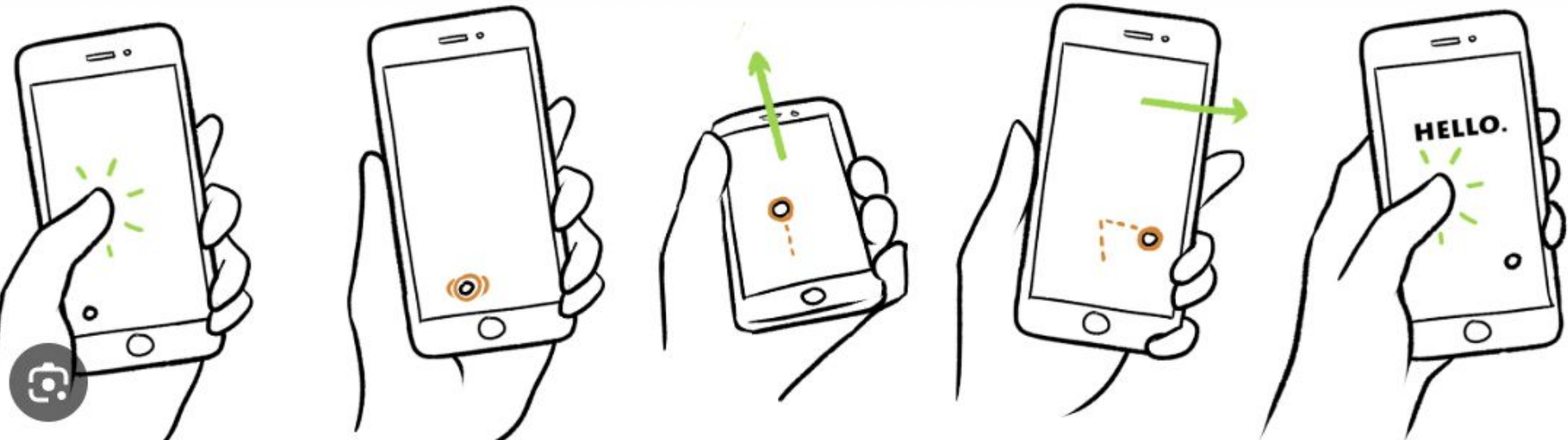
## 4 Ways to Give Constructive Feedback

- 1 BE SPECIFIC**  
If you're vague, your feedback can be misunderstood, and your employee may continue making the same mistakes.
- 2 BE TIMELY**  
Give prompt feedback at the next suitable moment, while the incident is fresh in mind.
- 3 BE POSITIVE**  
For any negative feedback you give, you need to also include positive feedback.
- 4 BE UNDERSTANDING**  
Discuss with your employee about the source of the mistake and what he or she could have done instead.

## 8. Support One-Handed Use

- Place primary interactive elements within the **thumb zone** (lower half of the screen).
- Avoid critical actions in top corners.
- Design for both **left-handed and right-handed** users.

*Example:* Floating Action Buttons (FABs) in Material Design appear at the bottom right for easy reach.



## 9. Design for Accessibility

- Ensure compatibility with **screen readers** (TalkBack, VoiceOver).
- Provide **alt text**, **clear labels**, and **scalable text**.
- Use **sufficient contrast**, **consistent layouts**, and **voice control options**.

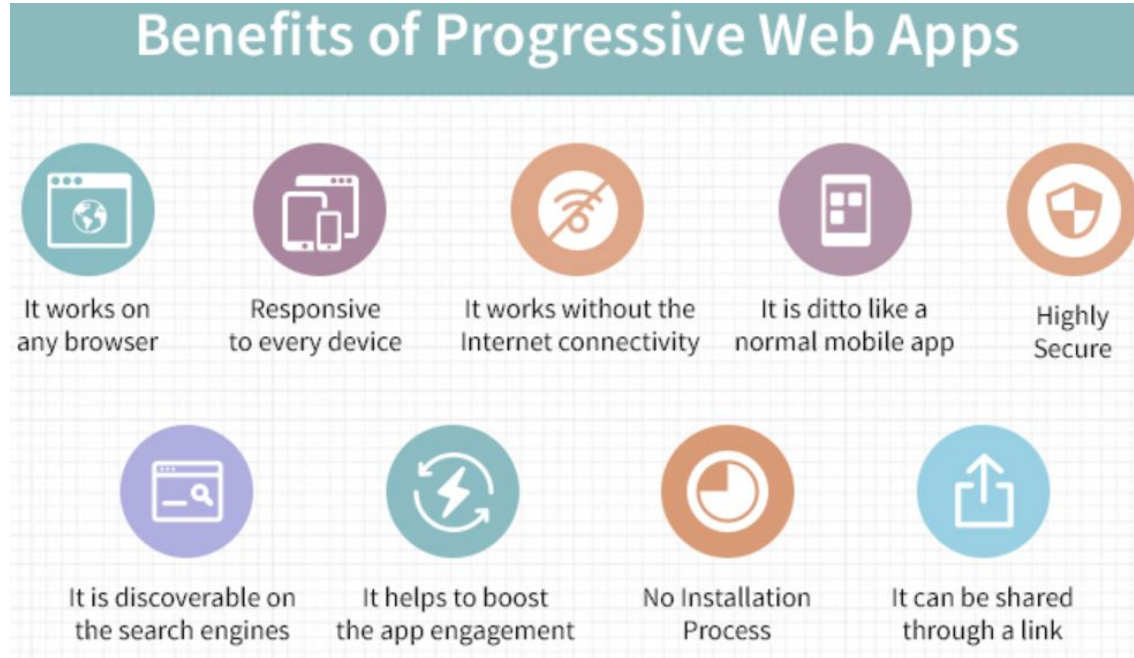
*Example:* iOS Accessibility settings allow users to enlarge text — apps should adapt gracefully.



## 10. Optimize Performance

- Mobile users often have limited data and slower connections.
- Minimize loading times and optimize images, animations, and scripts. Use **lazy loading** and **offline modes** when possible.

*Example:* Progressive Web Apps (PWAs) like Twitter Lite load fast even on 3G networks.



# 11. Design for Context of Use

- Mobile users may be **outdoors**, **multitasking**, or in **low-light** conditions.
- Support **dark/light modes**, clear icons, and minimal distractions.
- Consider **short attention spans** — prioritize speed and clarity.

*Example:* Google Maps automatically switches to dark mode at night.



## 12. Test on Real Devices

- Emulators aren't enough — test on **physical devices** with different screen sizes, lighting, and accessibility settings.
- Observe real user behavior to catch issues with gestures, glare, and reachability.

*Example:*

Conduct usability testing on both iOS and Android phones in different environments.



# Activity

Compare desktop Gmail vs. mobile Gmail — identify what's changed and why.

# Designing for Mobile Accessibility

## Accessibility on Mobile

- **VoiceOver / TalkBack** screen readers
- **Dynamic text resizing**
- **High contrast & dark mode support**
- **Gestures vs. alternative inputs (switch control, voice access)**
- **Orientation handling (portrait/landscape)**

# Case Study: Accessible Mobile Banking App

Scenario: Users with low vision and limited dexterity

- Challenge: Login and transfer money easily
- Discussion: How can we redesign UI to help?
  - Larger tap areas
  - Text-to-speech labels
  - Fewer steps
  - Simple iconography

## Group Activity:

Teams redesign one screen (e.g., login or settings) to meet both **mobile usability** and **accessibility** needs.

Then share solutions briefly.

# Tools, Testing, and Evaluation

## Accessibility Evaluation Tools

- **WAVE, axe, Lighthouse**
- **Color Contrast Analyzer**
- **Screen Readers:** NVDA, JAWS, VoiceOver
- **Mobile tools:** Android Accessibility Scanner, Xcode Accessibility Inspector

## Testing Methods

- Automated accessibility checks
- Heuristic evaluation for accessibility
- User testing with assistive technology users

# Summary & Reflection

- Accessibility = inclusion + better usability
- Mobile-first + accessible design are **complementary**, not separate goals
- Importance of continuous testing and user feedback

## Quick Recap:

“Accessibility makes technology usable by everyone, and mobile constraints push us to design efficiently.”

# Activity

Pick any popular mobile app (e.g., Instagram, Uber, Duolingo).

1. Identify 3 accessibility issues.
2. Suggest design improvements following WCAG(Web Content Accessibility Guidelines) principles.

# Thanks

Any Question