TEAM 2

Orange Sulphur

FRESH ORGANIC PRODUCE DELIVERED TO YOUR DOOR

Introduction

Today our topic is Modern Accessibility and the Web: how screen readers, ARIA, rich media APIs, SpeechAPI and other modern browser features help users with disabilities access websites.

Learning Outcomes

Application & Analysis:

- Implement Web Speech API (both Speech Recognition and Speech Synthesis) in a React application
- Compare different accessibility approaches and select appropriate solutions for specific use
- cases
- **Analyze** websites for accessibility compliance using tools like Lighthouse and WAVE

Synthesis & Evaluation:

- Design accessible user interfaces that support keyboard navigation, screen readers, and voice control
- Evaluate the trade-offs between feature richness and accessibility in modern web applications
- **Create** inclusive web experiences that benefit all users regardless of ability

Why Accessibility Matters

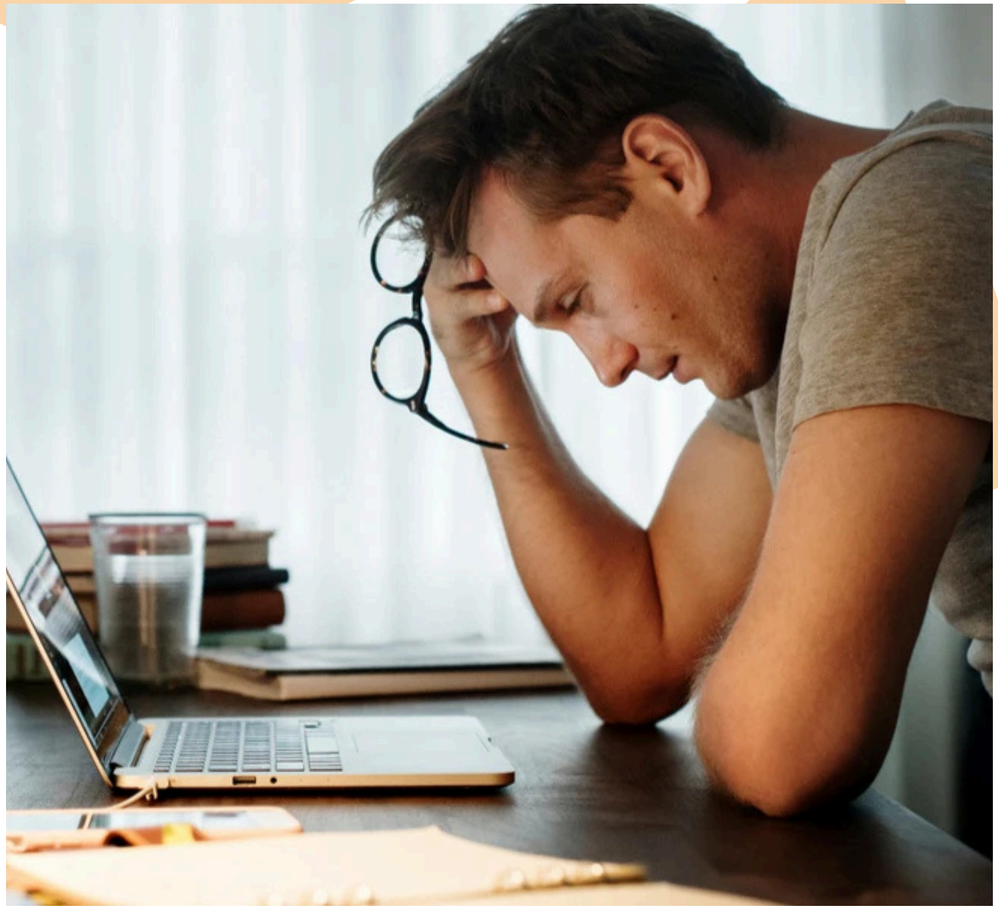
Current scenario:

- ✓ 26% of adults have a disability
- ✓ Websites must be usable for everyone
- ✓ Accessibility is required by law (ADA, Section 508)
- ✓ Makes products better for all users

Common Problems on the Web

Current scenario:

- Buttons have no label
- Images have no descriptions
- Users cannot navigate with keyboard
- Dynamic content not announced
- Color contrast is too low for color-blind users



Accessibility Features Implemented

Motion & Animation

- Smooth/instant transitions based on settings
- Respects user preferences
- Reduced motion option

Keyboard Navigation

- Full keyboard support
- Focus indicators
- Reduced motion option
- Skip to main content links

Visual Accessibility

- Adjustable font sizes (Normal, Large, Extra Large)
- Color-blind modes (Protanopia, Deuteranopia, Tritanopia)
- Dyslexia-friendly font option

Accessibility Features Implemented

Screen Reader Support

- Semantic HTML structure
- Proper ARIA labels and roles
- Live regions for dynamic content
- Alternative text for images

Speech & Voice Control

- Auto-read search results
- Product information narration
- Customizable voice settings
- Voice search functionality
- Hands-free navigation mode

Hands-Free Mode: The Star Feature

Activated via: Ctrl + Alt + A or ⌘ + Option + A

Voice Commands:

- Navigation: "go to store", "go to recipes", "go to cart"
- Search: "search for apples"
- Scrolling: "scroll down", "scroll to top"
- Cart: "show my cart"
- Information: "read this page", "help
- Exit: "exit hands-free mode"

Technical Implementation: Hands-Free Mode

Activated via: Ctrl + Alt + A or ⌘ + Option + A

Architecture:

```
HandsFreeContext (Provider)
    ↓
Web Speech API
    ↓
Voice Command Parser
    ↓
Action Handlers (Navigate, Search, Read)
```

Key Features

- Continuous listening
- Prevents infinite loops (system hearing itself)
- Restart after announcements

TEAM 2

Orange Sulphur

FRESH ORGANIC PRODUCE DELIVERED TO YOUR DOOR

DEMO

USECASE

INTERACTION

ACCESS

FEATURES

Hands free Mode : Challenges

Challenge 1: Infinite Loop Prevention

- System announces "Navigating to store"
- Microphone hears the announcement
- Processes it as a command
- Creates infinite loop

Solution

- Added isSpeakingRefflag
- Abort recognition while speaking
- Filter results during speech
- Delayed restart with buffers

Hands free Mode : Challenges

Challenge 2: Continuous Listening

- Recognition stops after each command
- User must manually restart
- Poor user experience

Solution

- Set continuous: true
- Auto-restart in onendhandler
- State management with refs

Accessibility Audit Results

Before Implementation:

- Accessibility Score: ~60/100
- Missing ARIA labels
- Poor keyboard navigation
- No screen reader support

After Implementation

- Accessibility Score: 95+/100
- Full ARIA compliance
- Complete keyboard support
- Excellent screen reader experience

Lessons Learned

Accessibility from the start - Easier to build in than retrofit

Test with real users- Actual screen reader usage reveals issues

Web Speech API challenges

Browser support varies

Background noise handling

Permission management