Keyboard Web AR Lenses

Camera Kit Web SDK · Snapchat Augmented Reality Platform

THE PROBLEM

When Snapchat's Camera Kit launched on the web (https://developers.snap.com/camera-kit/integrate-sdk/web/web-configuration), there was a fundamental gap: AR Lenses that required text input simply didn't work. Keyboard-driven Lenses — experiences that let users type into an AR scene — existed on native iOS and Android, but the web SDK had no equivalent. Developers building web AR experiences with Camera Kit had no way to unlock this entire category of interactive Lenses.

THE SOLUTION

I led the design and implementation of keyboard support for the Camera Kit Web SDK. The core engineering challenge wasn't just adding a text input, it was that the web has no single keyboard model. iOS Safari, Android Chrome, and desktop browsers each handle virtual keyboard events, viewport resizing, and focus behavior differently, with no consistent API across them. I had to design a solution that abstracted those differences into a unified interface, and coordinate closely with the iOS and Android SDK teams to ensure the web implementation matched the behavior contract that Lens creators already depended on.

The result exposed two integration paths: a ready-made <textarea> element the SDK manages automatically, and a lower-level event API for developers who want to build their own input UI, both of which correctly route text into the active Lens regardless of the browser or device.

THE PROOF

The feature shipped to production and is documented in the public Camera Kit Web SDK, where it is available to every developer building web AR experiences on Snapchat's platform.

Stack: TypeScript · Camera Kit Web SDK · Cross-platform browser APIs (iOS Safari, Android Chrome, macOS, Windows)