Sensing Web App (Task 1) – Process Report

DeviceMotion & DeviceOrientation with Local/Cloud Storage

Luisa Faust

2025

Abstract

This short report documents the end-to-end process of building a browser-based sensing app that collects device orientation, accelerometer, and gyroscope data. The app provides a context input, a privacy-friendly toggle to start/stop collection, and two storage modes: (a) local CSV export and (b) cloud upload via EdgeML. I validate the pipeline in Chrome DevTools with the Sensors emulator and on an iPhone (HTTPS + runtime permission). Representative screenshots and plots are included.

1 Task Brief

- Create a HTTPS web page with a form: text field ("Context") and a toggle switch.
- Enable sensor tracking on toggle (DeviceMotion / DeviceOrientation).
- Connect to EdgeML (`edge-ml/javascript`) with a write key; optionally extend sensors.
- Download and analyze the data with the EdgeML Python client (read key).

2 Stack & Hosting

Frontend: HTML/CSS/JS, Device APIs (Orientation/Motion). **Deployment:** GitHub Pages (automatic SSL).
Cloud: EdgeML JS client for writes; Python `edgeml` for reads.
Key references: CSS switch UI [1], Device APIs [2], [3], Sensors emulator [4], EdgeML JS [5].

3 Implementation Highlights

3.1 UI and Toggle

A clean gradient UI with a context text field and a switch that gates all sensor listeners. The status panel mirrors connection and active state.

3.2 Permission Gating (iOS 13+)

Some browsers (notably iOS Safari) require an explicit user gesture and permission for motion/orientation events. I request permission upon toggle enable:

```
// Guard iOS-style permission for motion/orientation
async function ensureMotionPermission() {
  const needsPerm = typeof DeviceMotionEvent?.requestPermission === '
     function';
  if (needsPerm) {
```

```
    const res = await DeviceMotionEvent.requestPermission();
    if (res !== 'granted') throw new Error('Motion permission denied');
  }
}
```

3.3 Wiring Sensor Events

```
// Start listeners (called when toggle ON)
function startSensors() {
  window.addEventListener('deviceorientation', onOrientation, true);
  window.addEventListener('devicemotion', onMotion, true);
  state.active = true;
}

// Stop listeners (called when toggle OFF)
function stopSensors() {
  window.removeEventListener('deviceorientation', onOrientation, true);
  window.removeEventListener('devicemotion', onMotion, true);
  state.active = false;
}

function onOrientation(e) {
  const { alpha, beta, gamma } = e;
  pushSample('orientation', { alpha, beta, gamma });
}

function onMotion(e) {
  const a = e.accelerationIncludingGravity || e.acceleration || {};
  const g = e.rotationRate || {};
  pushSample('accelerometer', { x: a.x, y: a.y, z: a.z });
  pushSample('gyroscope', { x: g.alpha, y: g.beta, z: g.gamma });
}
```

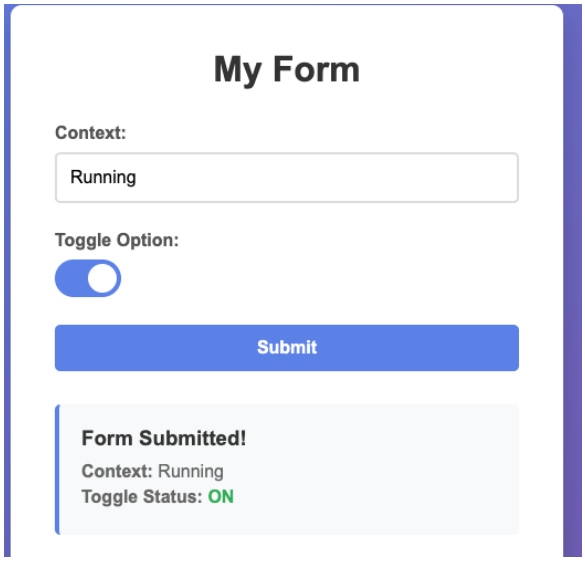
3.4 Dual Storage Modes

Local mode: append samples in memory and expose a CSV download.
EdgeML mode: batch samples and write to cloud using the JS client. Secrets are never hard-coded in public repos; for presentation I keep keys in a separate config and/or redact them.

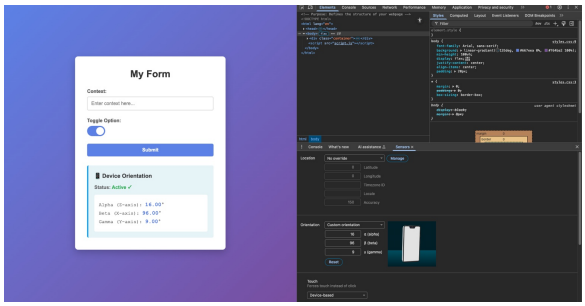
4 Validation & Debugging

4.1 Chrome DevTools Sensors

I emulate device orientation to validate the pipeline before testing on real hardware (Figure 1).



(a) Console stream while moving the virtual device.



(b) Sensors panel: custom orientation angles & 3D phone.

Figure 1: Chrome DevTools debugging of orientation/motion events.

4.2 UI Flow on Desktop

Form submission and status panels (Figure 2).

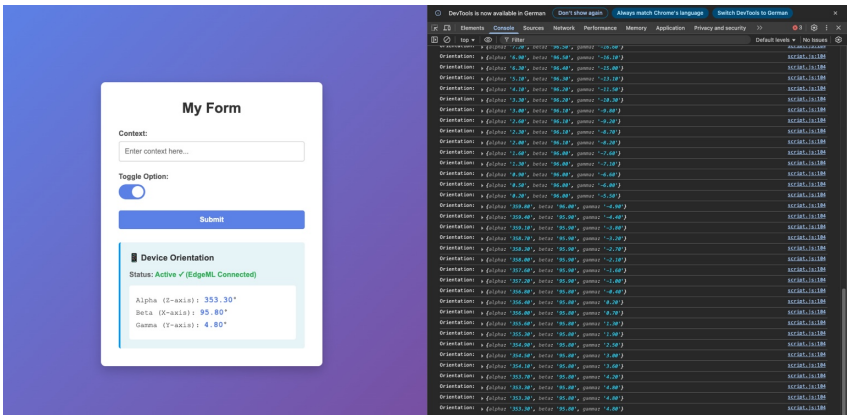
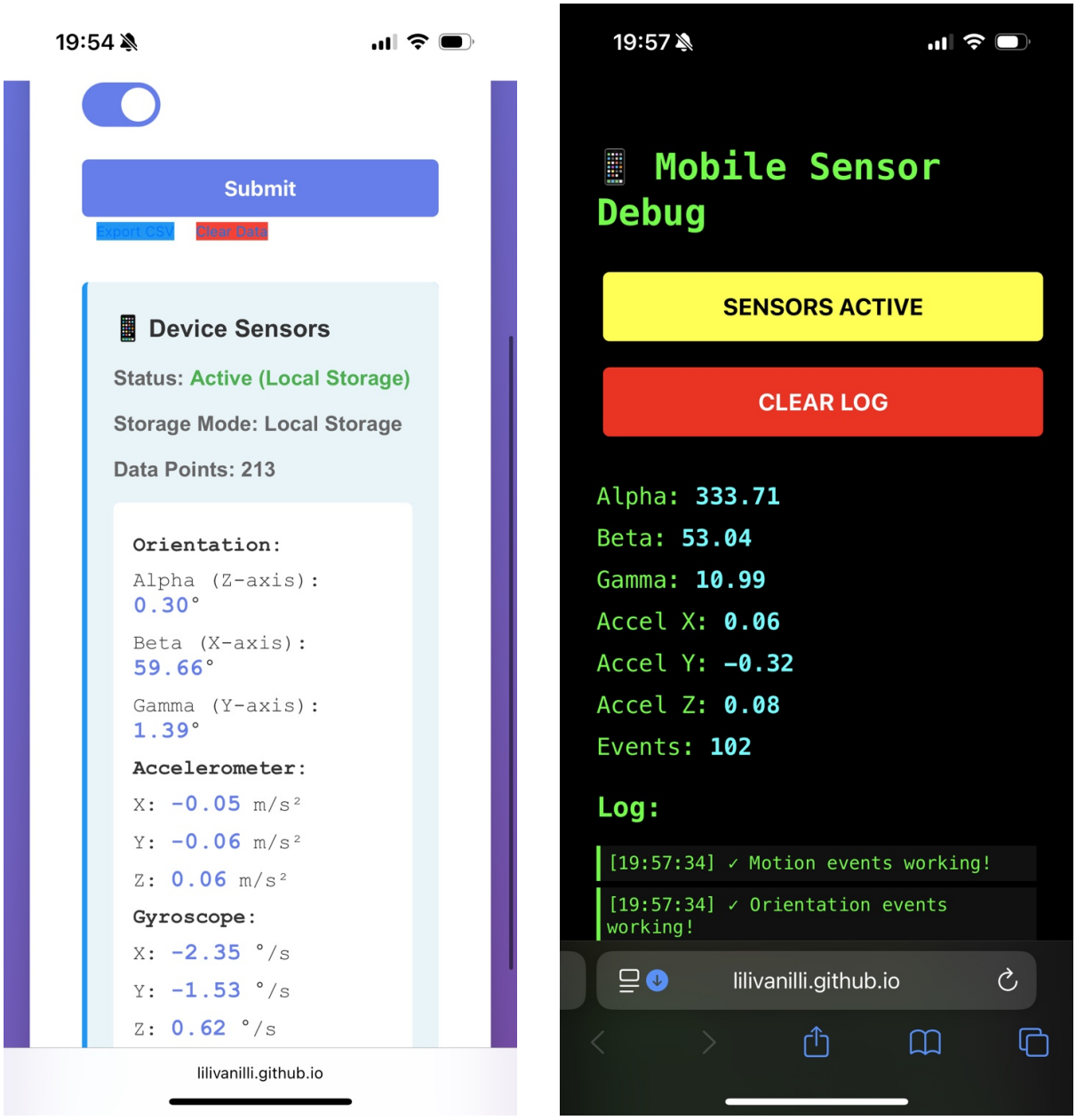


Figure 2: Form with context and toggle; submission feedback.

4.3 Mobile Validation (iPhone)

A dedicated mobile debug page verifies API support, permissions, and live values (Figure 3).



(a) Main sensing UI on iPhone (HTTPS). (b) Mobile sensor debugger confirms events.

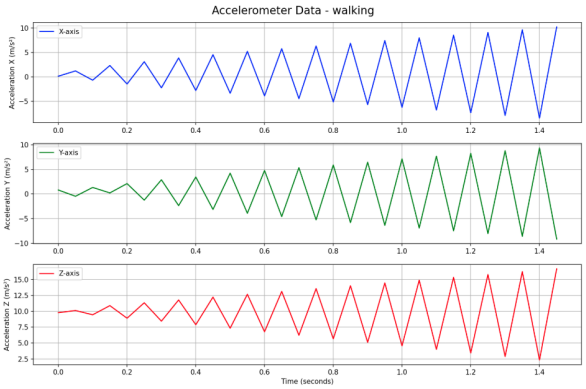
Figure 3: On-device validation and debugging.

5 Data & Analysis

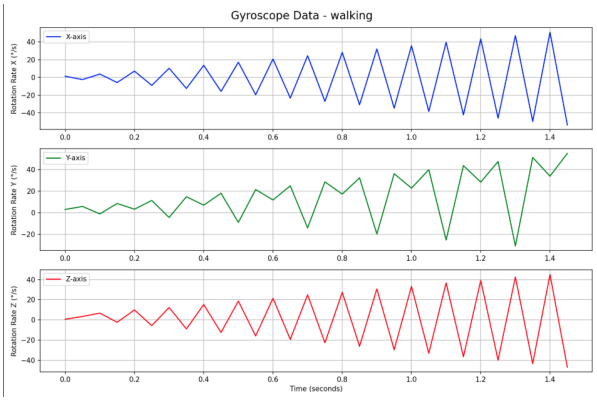
Samples contain timestamp, relative time, sensor type, axis values, and context. For local mode I export CSV; for cloud mode I later download with the Python client. Using the included analysis script, I generate time-series and magnitude plots. Representative outputs are shown in Figure 4.

```
Timestamp,Relative Time (s),Sensor Type,X,Y,Z,Context
1697288400000,0.000,orientation,45.23,12.45,5.67,walking
1697288400050,0.050,orientation,46.89,13.12,-2.34,walking
1697288400100,0.100,orientation,48.45,14.78,8.91,walking
1697288400150,0.150,orientation,44.12,11.23,3.45,walking
1697288400200,0.200,orientation,49.67,15.89,12.34,walking
1697288400250,0.250,orientation,43.89,10.45,-1.23,walking
1697288400300,0.300,orientation,50.23,16.34,15.67,walking
1697288400350,0.350,orientation,42.45,9.67,-4.56,walking
1697288400400,0.400,orientation,51.78,17.23,18.91,walking
1697288400450,0.450,orientation,41.23,8.89,-7.89,walking
1697288400500,0.500,orientation,53.12,18.45,21.23,walking
1697288400550,0.550,orientation,40.45,8.12,-11.45,walking
1697288400600,0.600,orientation,54.89,19.78,23.67,walking
1697288400650,0.650,orientation,39.12,7.45,-14.89,walking
1697288400700,0.700,orientation,56.23,21.12,25.34,walking
1697288400750,0.750,orientation,38.45,6.78,-18.12,walking
1697288400800,0.800,orientation,57.89,22.45,27.89,walking
1697288400850,0.850,orientation,37.67,6.12,-21.67,walking
1697288400900,0.900,orientation,59.12,23.89,29.45,walking
1697288400950,0.950,orientation,36.89,5.45,-25.23,walking
1697288401000,1.000,orientation,60.45,25.23,31.78,walking
1697288401050,1.050,orientation,35.23,4.78,-28.91,walking
1697288401100,1.100,orientation,61.78,26.67,33.12,walking
1697288401150,1.150,orientation,34.56,4.12,-32.45,walking
1697288401200,1.200,orientation,63.12,28.12,35.67,walking
1697288401250,1.250,orientation,33.89,3.45,-36.12,walking
1697288401300,1.300,orientation,64.45,29.45,37.23,walking
1697288401350,1.350,orientation,32.45,2.89,-39.78,walking
1697288401400,1.400,orientation,65.89,30.89,39.89,walking
1697288401450,1.450,orientation,31.23,2.23,-43.45,walking
1697288400000,0.000,accelerometer,0.15,0.82,9.78,walking
1697288400050,0.050,accelerometer,1.23,-0.45,10.12,walking
1697288400100,0.100,accelerometer,-0.67,1.34,9.45,walking
1697288400150,0.150,accelerometer,2.34,0.23,10.89,walking
1697288400200,0.200,accelerometer,-1.45,2.12,8.91,walking
```

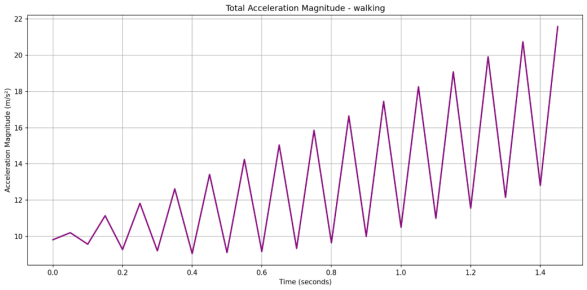
(a) Accelerometer axes over time.



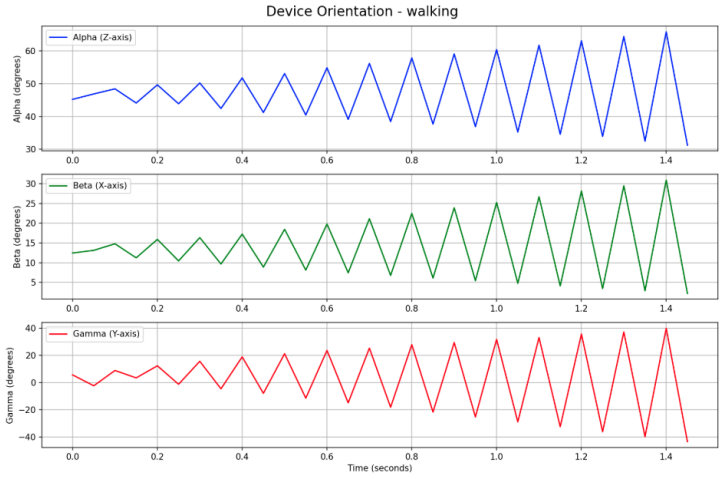
(b) Total acceleration magnitude.



(c) Gyroscope axes over time.



(d) Orientation angles over time.



(e) Excerpt of exported CSV (structure).

Figure 4: Analysis outputs generated from the collected data.

6 Lessons Learned

- **HTTPS and permissions** are mandatory on mobile; gate listeners behind explicit user actions.
- The **Sensors emulator** accelerates development; still verify on-device (sampling cadence, noise).
- Keep **storage modes decoupled** (local CSV vs. EdgeML) to enable offline testing.

References

[1] W3Schools. "Css toggle switch." Accessed 2025. [Online]. Available: https://www.w3schools.com/howto/howto_css_switch.asp.

[2] MDN Web Docs. "Deviceorientationevent - web apis." Accessed 2025. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/DeviceOrientationEvent.

[3] MDN Web Docs. "Devicemotionevent - web apis." Accessed 2025. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/DeviceMotionEvent.

[4] C. H. et al. "Device orientation events." Archived tutorial; Accessed 2025-10-14. [Online]. Available: https://www.html5rocks.com/en/tutorials/device/orientation/.

[5] EdgeML. "Edge-ml/javascript." Accessed 2025. [Online]. Available: https://github.com/edge-ml/javascript.