Mobile Details Checker :root{--bg:#0f1724;--card:#0b1220;--muted:#9aa4b2;--accent:#06b6d4} *{box-sizing:border-box} body{margin:0;font-family:Inter, ui-sans-serif, system-ui, -apple-system, 'Segoe UI', Roboto, 'Helvetica Neue', Arial;background:linear-gradient(180deg,#071022 0%, #0b1220 100%);color:#e6eef6;min-height:100vh;display:flex;align-items:center;justify-content:center;padding:20px} .container{width:100%;max-width:980px} header{display:flex;align-items:center;gap:12px;margin-bottom:18px} .logo{width:52px;height:52px;border-radius:12px;background:linear-gradient(135deg,var(--accent),#7c3aed);display:flex;align-items:center;justify-content:center;font-weight:700} h1{margin:0;font-size:20px} p.lead{margin:4px 0 0;color:var(--muted);font-size:13px} <pre><code>.card{background:rgba(255,255,255,0.02);padding:16px;border-radius:12px;box-shadow:0 6px 18px rgba(2,6,23,0.6)} .grid{display:grid;grid-template-columns:1fr 360px;gap:14px} @media(max-width:880px){.grid{grid-

template-columns:1fr}} .section{margin-bottom:12px} table{width:100%;border-collapse:collapse} tr{border-bottom:1px dashed rgba(255,255,255,0.03)} td{padding:10px 6px} td.key{width:45%;color:var(--muted);font-size:13px} td.value{font-weight:600} .controls{display:flex;gap:8px;flex-wrap:wrap} button{background:linear-gradient(90deg,var(--accent),#7c3aed);border:none;padding:8px 12px;border-radius:8px;color:#021226;font-weight:600;cursor:pointer} .btn-ghost{background:transparent;border:1px solid rgba(255,255,255,0.06);color:var(--muted);font-weight:600} .small{padding:6px 8px;font-size:13px} .right-card{position:relative} .notice{font-size:13px;color:var(--muted);margin-bottom:8px} pre{white-space:pre-wrap;word-break:break-word;background:rgba(0,0,0,0.2);padding:10px;border-radius:8px;color:#dbeafe} footer{margin-top:10px;color:var(--muted);font-size:13px} </code></pre>
MD

Mobile Details Checker

Ek simple page jo aapke mobile / browser details detect karta hai. Kuch features permission maangenge.

```
<div class="card grid"> <div> <div
class="section"> <div
style="display:flex;justify-content:space-
between;align-items:center;margin-
bottom:8px"> <strong>Detected details</
strong> <div class="controls"> <button
id="refreshBtn" class="small">Refresh</
button> <button id="copyBtn" class="small
btn-ghost">Copy JSON</button> <button
id="downloadBtn" class="small btn-
ghost">Download JSON</button> </div> </
div> <table id="infoTable" aria-live="polite">
<!-- rows added by JS --> </table> </div>
<div class="section"> <strong>User agent
parse</strong> <p class="notice">Exact
model detection is limited — we try to infer
from the user agent string.</p> <pre
id="uaParse"></pre> </div> <div
class="section"> <strong>Raw data
(debug)</strong> <pre id="rawData"></
pre> </div> </div> <aside class="card right-
card"> <div class="section"> <strong>Quick
actions</strong> <p class="notice">Kuch
features permission maang sakte hain
(Battery, Geolocation).</p> <div
style="display:flex;flex-
direction:column;gap:8px;margin-top:8px">
<button id="batteryBtn" class="small">Get
Battery Info</button> <button id="geoBtn"
class="small">Get Location</button>
```

<button id="cameraBtn" class="small">Test Camera (permission)</button> <button id="clearBtn" class="small btn-ghost">Clear Location</button> </div> </div> <div class="section"> <strong>Notes</strong> <p class="notice">- Browser APIs vary by device and privacy settings.

- For accurate handset model, OEM apps or native code are required.

<div class="section"> <strong>Export</strong> <p class="notice">Download or copy detected details as JSON for sharing.</p> </div> </aside> Made for you — refresh to update. Some values may be unavailable depending on browser. function safe(v){return v===undefined|| v===null?'-':String(v)} async function gather(){ const nav = navigator || {}; const screen = window.screen || {}; const conn = nav.connection || nav.mozConnection || nav.webkitConnection || {}; const data = { userAgent: navigator.userAgent || '-', platform: navigator.platform || '-', vendor: navigator.vendor || '-', language: navigator.language || navigator.languages || '-', cookiesEnabled: navigator.cookieEnabled ?? '-', deviceMemory: navigator.deviceMemory ?? '-', hardwareConcurrency: navigator.hardwareConcurrency ?? '-',

```
product: navigator.product || '-',
productSub: navigator.productSub || '-',
screenWidth: screen.width || '-',
screenHeight: screen.height || '-',
availWidth: screen.availWidth || '-',
availHeight: screen.availHeight || '-',
pixelRatio: window.devicePixelRatio || '-',
colorDepth: screen.colorDepth || '-',
connectionEffectiveType:
conn.effectiveType || '-',
connectionDownlink: conn.downlink ?? '-',
connectionRtt: conn.rtt ?? '-', touchPoints:
navigator.maxTouchPoints ?? '-', online:
navigator.onLine ?? '-', timeZone:
Intl.DateTimeFormat().resolvedOptions().ti
meZone || '-', timestamp: new
Date().toISOString() }; // try battery
(permission may be needed) try{
if(navigator.getBattery) { const batt = await
navigator.getBattery(); data.battery =
{charging:batt.charging, level:batt.level,
chargingTime:batt.chargingTime,
dischargingTime:batt.dischargingTime}; } }
catch(e){data.battery='unavailable'} // try
geolocation (will prompt) data.location =
null; // parse UA for possible model hints
data.uaModelGuess =
parseUserAgentForModel(data.userAgent);
 return data; } function
parseUserAgentForModel(ua){ const s =
ua || ''; // quick heuristics if(/
iPhone/.test(s)){ const m = s.match(/
iPhone; CPU iPhone OS ([\d_]+)/); return
{os:'iOS', device:'iPhone', osVersion: m?
m[1].replace(/_/g,'.'):'unknown'}; } if(/
Android/.test(s)){ // try to catch model
after Build/ const buildMatch = s.match(/;
\s*(?:[A-Za-z0-9\- ]+?)\s*Build\/([A-Za-
z0-9\-_.]+)/); // many UA include model like
"SM-G991B" or "Pixel 6" const modelMatch
= s.match(/\b([A-Za-z0-9\-]+)\s+Build\//) ||
s.match(/\b(Pixel\s\d+|Pixel_\d+)/); return
{os:'Android', modelGuess: modelMatch?
modelMatch[0]: (buildMatch?
buildMatch[1]:'unknown') }; } return
{os:'Unknown', raw: s.substring(0,120)}; }
function render(data){ const table =
document.getElementById('infoTable');
table.innerHTML=''; const rows = [ ['User
agent', data.userAgent], ['Platform',
data.platform], ['Vendor', data.vendor],
['Language(s)',
Array.isArray(data.language)?
data.language.join(', '):data.language],
['Cookies enabled', data.cookiesEnabled],
['Device memory (GB)',
data.deviceMemory], ['CPU threads',
data.hardwareConcurrency], ['Screen (w ×
h)', data.screenWidth + ' × ' +
data.screenHeight], ['Pixel ratio',
data.pixelRatio], ['Color depth',
data.colorDepth], ['Touch points',
data.touchPoints], ['Online', data.online],
['Connection type',
data.connectionEffectiveType +
(data.connectionDownlink?(' • ' +
data.connectionDownlink + ' Mbps'):'')],
['Time zone', data.timeZone], ['Timestamp',
data.timestamp] ]; rows.forEach(([k,v])=>{
const tr=document.createElement('tr');
tr.innerHTML=`<td class="key">${k}</
td><td class="value">${safe(v)}</td>`;
table.appendChild(tr); });
document.getElementById('uaParse').textC
ontent =
JSON.stringify(data.uaModelGuess, null,
2);
document.getElementById('rawData').textC
ontent = JSON.stringify(data, null, 2); }
async function doGatherAndRender(){
const d = await gather(); window._lastData
= d; render(d); }
document.getElementById('refreshBtn').ad
dEventListener('click',
()=>doGatherAndRender());
document.getElementById('copyBtn').addEv
entListener('click', async ()=>{ try{ await
navigator.clipboard.writeText(JSON.stringif
y(window._lastData || {}, null, 2));
alert('JSON copied to clipboard'); }catch(e)
```

```
{alert('Copy failed: ' + e.message)} });
document.getElementById('downloadBtn').
addEventListener('click', ()=>{ const blob =
new
Blob([JSON.stringify(window._lastData || {},
null, 2)], {type:'application/json'}); const url
= URL.createObjectURL(blob); const a =
document.createElement('a'); a.href = url;
a.download = 'mobile-details.json';
a.click(); URL.revokeObjectURL(url); }); //
battery button
document.getElementById('batteryBtn').add
EventListener('click', async ()=>{ try{ if(!
navigator.getBattery) return alert('Battery
API not supported here'); const batt =
await navigator.getBattery(); alert('Battery:
charging=' + batt.charging + ' level=' +
batt.level); window._lastData =
window._lastData || {};
window._lastData.battery =
{charging:batt.charging, level:batt.level};
render(window._lastData); }catch(e)
{alert('Battery read failed: '+e.message)} });
// geolocation let lastPos = null;
document.getElementById('geoBtn').addEv
entListener('click', ()=>{ if(!
navigator.geolocation) return
alert('Geolocation not supported');
navigator.geolocation.getCurrentPosition(p
os=>{ lastPos = {lat: pos.coords.latitude,
lon: pos.coords.longitude, accuracy:
pos.coords.accuracy}; window._lastData =
window._lastData || {};
window._lastData.location = lastPos;
render(window._lastData); alert('Location
saved (check Raw data)'); }, err=>{
alert('Location error: ' + err.message); },
{maximumAge:60000,timeout:15000}); });
document.getElementById('clearBtn').addE
ventListener('click', ()=>{
if(window._lastData)
{window._lastData.location=null;
render(window._lastData);} }); // camera
test (will prompt)
document.getElementById('cameraBtn').ad
dEventListener('click', async ()=>{ try{
const s = await
navigator.mediaDevices.getUserMedia({vid
eo:true}); // stop tracks immediately; this
just tests permission
s.getTracks().forEach(t=>t.stop());
alert('Camera access granted (permission
test)'); }catch(e){alert('Camera test failed:
'+e.message)} }); // initial
doGatherAndRender();
```