

Professor Ape is using AI to create an app.



Here is he:

He wants to use ChatGPT



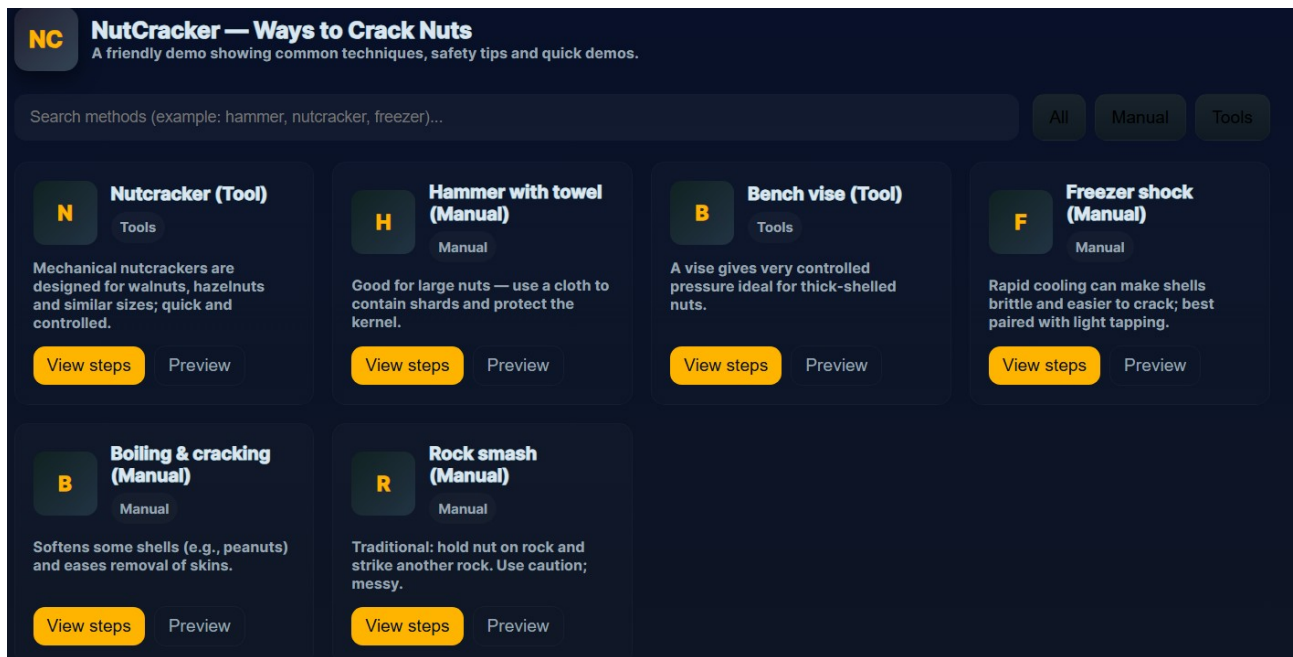
He types here definition of the problem.



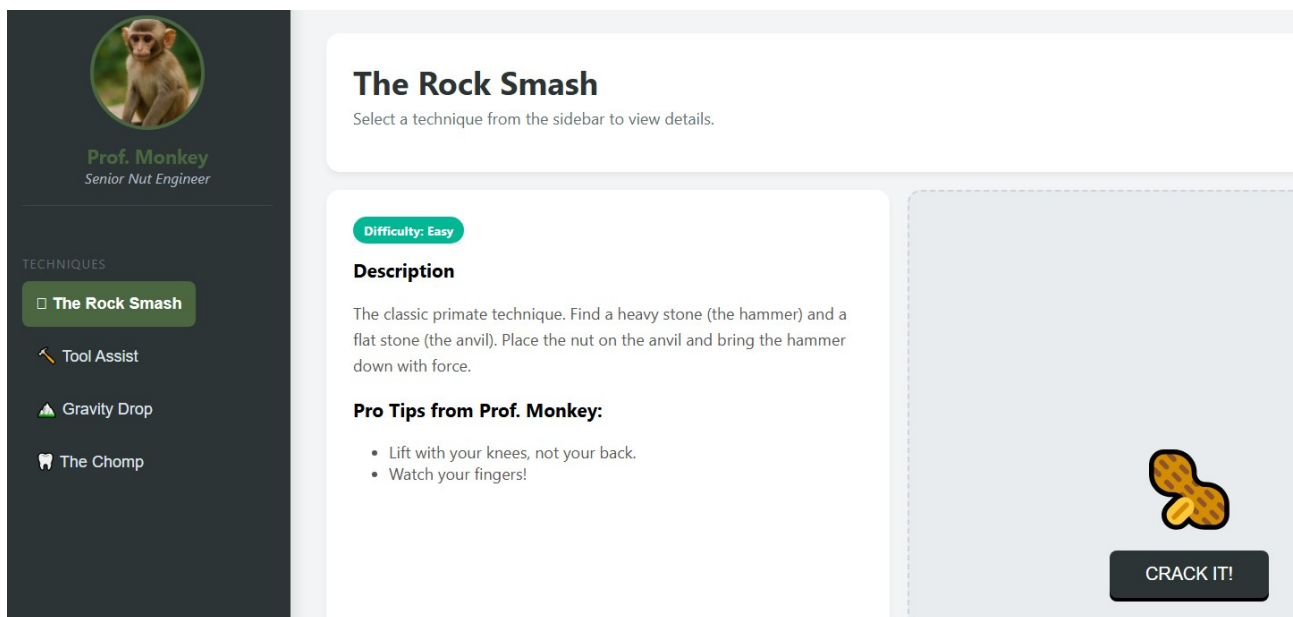
It is about different ways to crack nuts!

Very interesting and important topic, really!

He tries first ChatGPT and gets this kind of UI (codes are shown later)



Then he tries AIstudio and gets this UI:



Very nice, apelike, UI!

So, mr Ape is testing apps and tells later results.



Prof. Ape wants also others be able to try apps and they are published!
Here are links:

<https://brunovossilo.neocities.org/ape1/ape3.html>

<https://brunovossilo.neocities.org/ape2/ape2.html>

CODES

1)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>NutCracker — Ways to Crack Nuts</title>
  <style>
    :root{--bg:#0f1724;--card:#0b1220;--muted:#9aa6b2;--accent:#ffb400;--
glass:rgba(255,255,255,0.04)}
    *{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,#071029 0%,#0f1724 100%);color:#dbe7f0;min-
height:100vh;display:flex;align-items:center;justify-
content:center;padding:32px}
    .app{width:100%;max-width:1100px}

    header{display:flex;align-items:center;gap:16px;margin-bottom:20px}
    .logo{display:flex;align-items:center;gap:12px}
    .logo .mark{width:56px;height:56px;border-
radius:12px;background:linear-gradient(135deg,#223 0%,#345
100%);display:grid;place-items:center;font-weight:700;color:var(--
accent);font-size:20px;box-shadow:0 6px 18px rgba(0,0,0,0.5)}
    h1{font-size:20px;margin:0}
    p.lead{margin:0;color:var(--muted);font-size:13px}

    .controls{display:flex;gap:12px;margin:18px 0}
    .search{flex:1;display:flex}
    .search input{width:100%;padding:12px 14px;border-
radius:10px;border:none;background:var(--glass);color:inherit;font-
size:14px}
    .filter{display:flex;gap:8px}
    .btn{background:linear-
gradient(180deg,#18232b,#0f1a22);padding:10px 14px;border-
radius:10px;border:1px solid rgba(255,255,255,0.03);cursor:pointer}
```

```

.grid{display:grid;grid-template-columns:repeat(auto-
fill,minmax(260px,1fr));gap:16px}
.card{background:linear-
gradient(180deg,rgba(255,255,255,0.02),rgba(255,255,255,0.01));padding:
16px;border-radius:12px;border:1px solid
rgba(255,255,255,0.03);display:flex;flex-direction:column;gap:12px;min-
height:160px}
.card h3{margin:0;font-size:16px}
.chip{display:inline-block;padding:6px 8px;border-
radius:999px;background:rgba(255,255,255,0.03);font-
size:12px;color:var(--muted)}
.card p{margin:0;color:var(--muted);font-size:13px;flex:1}
.actions{display:flex;gap:8px;align-items:center}
.primary{background:var(--accent);color:#06111a;padding:8px
12px;border-radius:10px;border:none;cursor:pointer}
.ghost{background:transparent;border:1px solid
rgba(255,255,255,0.04);padding:8px 12px;border-radius:10px;color:var(--
muted);cursor:pointer}

footer{margin-top:18px;color:var(--muted);font-size:13px;text-
align:center}

/* Modal */
.modal-backdrop{position:fixed;inset:0;display:none;place-
items:center;background:linear-
gradient(180deg,rgba(2,6,23,0.7),rgba(2,6,23,0.85));z-index:60}
.modal{width:100%;max-width:760px;background:#0b1220;border-
radius:12px;padding:20px;border:1px solid rgba(255,255,255,0.03)}
.modal h2{margin:0 0 8px 0}
.steps{display:flex;flex-direction:column;gap:10px;margin-top:12px}
.step{background:rgba(255,255,255,0.02);padding:12px;border-
radius:10px}
.safety{background:#2b1100;color:#ffd8b3;padding:10px;border-
radius:10px;font-size:13px}

/* Responsive */
@media (max-width:640px){header{flex-direction:column;align-
items:flex-start}.controls{flex-direction:column}}
</style>

```

```
</head>
<body>
  <div class="app">
    <header>
      <div class="logo">
        <div class="mark">NC</div>
        <div>
          <h1>NutCracker — Ways to Crack Nuts</h1>
          <p class="lead">A friendly demo showing common techniques,
safety tips and quick demos.</p>
        </div>
      </div>
    </header>

    <section class="controls" aria-label="controls">
      <div class="search">
        <input id="search" placeholder="Search methods (example: hammer,
nutcracker, freezer)..." />
      </div>
      <div class="filter">
        <button class="btn" id="showAll">All</button>
        <button class="btn" id="showManual">Manual</button>
        <button class="btn" id="showTools">Tools</button>
      </div>
    </section>

    <main>
      <div id="grid" class="grid"></div>
    </main>

    <footer>Click any method to view steps. This is an educational demo —
follow safe, food-grade techniques only.</footer>
  </div>

  <!-- Modal -->
  <div id="backdrop" class="modal-backdrop">
    <div class="modal" role="dialog" aria-modal="true">
```

```

    <button id="closeModal"
style="float:right;background:transparent;border:none;color:var(--
muted);cursor:pointer">Close ✕</button>
    <h2 id="modalTitle">Method</h2>
    <p id="modalDesc" style="color:var(--muted)"></p>
    <div class="steps" id="modalSteps"></div>
    <div style="margin-top:12px" class="safety" id="safetyBox"></div>
    <div style="display:flex;gap:8px;margin-top:12px">
        <button class="primary" id="copyBtn">Copy steps</button>
        <button class="ghost" id="closeBtn">Close</button>
    </div>
</div>
</div>

```

```

<script>
// Lightweight dataset of methods. Each entry includes title, category,
description and ordered steps.
const METHODS = [
    {
        id: 'nutcracker',
        title: 'Nutcracker (Tool)',
        category: 'Tools',
        desc: 'Mechanical nutcrackers are designed for walnuts, hazelnuts and
similar sizes; quick and controlled.',
        steps: ['Place nut in the jaw of the nutcracker.', 'Squeeze steadily until
the shell cracks.', 'Remove shell pieces and extract the kernel.'],
    },
    {
        id: 'hammer',
        title: 'Hammer with towel (Manual)',
        category: 'Manual',
        desc: 'Good for large nuts — use a cloth to contain shards and protect
the kernel.',
        steps: ['Wrap the nut in a towel on a sturdy surface.', 'Tap gently with
a hammer — start light, increase force as needed.', 'Pick out the nut meat
carefully.'],
    },
    {
        id: 'benchvise',

```

```

    title: 'Bench vise (Tool)',
    category: 'Tools',
    desc: 'A vise gives very controlled pressure ideal for thick-shelled
nuts.',
    steps: ['Place the nut in the vise jaws.', 'Tighten slowly until the shell
fractures.', 'Open the vise and remove the nut.'],
  },
  {
    id: 'freezer',
    title: 'Freezer shock (Manual)',
    category: 'Manual',
    desc: 'Rapid cooling can make shells brittle and easier to crack; best
paired with light tapping.',
    steps: ['Freeze the nuts for several hours.', 'Remove and allow a
minute to acclimate.', 'Tap on a surface or use a nutcracker to open.'],
  },
  {
    id: 'boiling',
    title: 'Boiling & cracking (Manual)',
    category: 'Manual',
    desc: 'Softens some shells (e.g., peanuts) and eases removal of skins.',
    steps: ['Boil nuts for 3–10 minutes depending on size.', 'Drain and
cool slightly.', 'Use thumbs or a light press to open the shell.'],
  },
  {
    id: 'rock',
    title: 'Rock smash (Manual)',
    category: 'Manual',
    desc: 'Traditional: hold nut on rock and strike another rock. Use
caution; messy.',
    steps: ['Place nut on a stable lower rock.', 'Strike with an upper rock
with a measured blow.', 'Separate shell and kernel.'],
  }
];

```

// DOM references

```

const grid = document.getElementById('grid');
const searchInput = document.getElementById('search');
const showAll = document.getElementById('showAll');

```

```
const showManual = document.getElementById('showManual');
const showTools = document.getElementById('showTools');
```

```
const backdrop = document.getElementById('backdrop');
const modalTitle = document.getElementById('modalTitle');
const modalDesc = document.getElementById('modalDesc');
const modalSteps = document.getElementById('modalSteps');
const safetyBox = document.getElementById('safetyBox');
const copyBtn = document.getElementById('copyBtn');
const closeBtn = document.getElementById('closeBtn');
const closeModal = document.getElementById('closeModal');
```

```
// Render cards
```

```
function render(items){
  grid.innerHTML = '';
  items.forEach(m => {
    const el = document.createElement('article');
    el.className = 'card';
    el.innerHTML = `
      <div style="display:flex;gap:12px;align-items:center">
        <div style="width:56px;height:56px;border-
radius:10px;background:linear-gradient(135deg,#122 0%,#234
100%);display:grid;place-items:center;font-weight:700;color:var(--
accent)">${m.title.split(' ')[0][0]}</div>
        <div style="flex:1">
          <h3>${m.title}</h3>
          <div style="margin-top:6px"><span class="chip">${m.category}
</span></div>
        </div>
      </div>
      <p>${m.desc}</p>
      <div class="actions">
        <button class="primary" data-id="${m.id}">View steps</button>
        <button class="ghost" data-id="${m.id}">Preview</button>
      </div>
    `;
    grid.appendChild(el);
  });
}
```

```

// Initial render
render(METHODS);

// Filtering helpers
function filterByCategory(cat){
  if(cat === 'All') render(METHODS);
  else render(METHODS.filter(m => m.category === cat));
}

showAll.addEventListener('click', ()=> filterByCategory('All'));
showManual.addEventListener('click', ()=> filterByCategory('Manual'));
showTools.addEventListener('click', ()=> filterByCategory('Tools'));

searchInput.addEventListener('input', () => {
  const q = searchInput.value.trim().toLowerCase();
  if(!q) return render(METHODS);
  const out = METHODS.filter(m => (m.title + ' ' + m.desc + ' ' +
m.steps.join(' ')).toLowerCase().includes(q));
  render(out);
});

// Delegate clicks on grid for buttons
grid.addEventListener('click', (ev) => {
  const btn = ev.target.closest('button');
  if(!btn) return;
  const id = btn.getAttribute('data-id');
  if(!id) return;
  const method = METHODS.find(m=>m.id===id);
  if(!method) return;
  // If primary: show modal with detailed steps
  if(btn.classList.contains('primary')){
    openModal(method);
  } else {
    // Preview: show quick alert-style small modal
    openModal(method, {preview:true});
  }
});

```

```

function openModal(method, opts={}){
  modalTitle.textContent = method.title;
  modalDesc.textContent = method.desc;
  modalSteps.innerHTML = "";
  const steps = method.steps;
  steps.forEach((s,i)=>{
    const el = document.createElement('div');
    el.className = 'step';
    el.innerHTML = `<strong>Step ${i+1}</strong> <span
style="color:var(--muted)">${s}</span>`;
    modalSteps.appendChild(el);
  });
  safetyBox.textContent = 'Safety tip: Wear eye protection, work on a
stable surface, and avoid excessive force. Dispose of shells safely.';
  backdrop.style.display = 'grid';

  // Set copy button action
  copyBtn.onclick = ()=>{
    const text = method.title + '\n' + method.desc + '\n\n' +
method.steps.map((s,i)=>`${i+1}. ${s}`).join('\n');
    navigator.clipboard?.writeText(text).then(()=>{
      copyBtn.textContent = 'Copied!';
      setTimeout(()=>copyBtn.textContent='Copy steps',1200);
    }).catch(()=>{
      copyBtn.textContent = 'Copy failed';
      setTimeout(()=>copyBtn.textContent='Copy steps',1200);
    });
  };

  closeBtn.onclick = closeModalFunc;
  closeModal.onclick = closeModalFunc;
  backdrop.onclick = (e)=>{ if(e.target === backdrop)
closeModalFunc(); };
}

function closeModalFunc(){ backdrop.style.display='none'; }

// Keyboard accessibility
document.addEventListener('keydown', (e)=>{

```

```

    if(e.key === 'Escape') closeModalFunc();
  });

  // Small accessibility: allow Enter on focused search to focus grid
  searchInput.addEventListener('keydown', (e)=>{ if(e.key === 'Enter')
{ const first = grid.querySelector('button'); first && first.focus(); }});
</script>
</body>
</html>

```

2)

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>NutCracker Pro - Knowledge Base</title>
  <style>
    :root {
      --primary: #4a6741; /* Jungle Green */
      --accent: #d4a373; /* Wood color */
      --bg: #f3f4f6;
      --sidebar-bg: #2d3436;
      --text-light: #dfe6e9;
    }

    body {
      margin: 0;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background-color: var(--bg);
      height: 100vh;
      display: flex;
      overflow: hidden;
    }

    /* --- SIDEBAR (The Monkey's Profile) --- */

```

```
.sidebar {  
  width: 280px;  
  background-color: var(--sidebar-bg);  
  color: var(--text-light);  
  display: flex;  
  flex-direction: column;  
  padding: 20px;  
  box-shadow: 4px 0 15px rgba(0,0,0,0.1);  
}
```

```
.profile-section {  
  text-align: center;  
  margin-bottom: 40px;  
  border-bottom: 1px solid #444;  
  padding-bottom: 20px;  
}
```

```
.avatar {  
  width: 120px;  
  height: 120px;  
  border-radius: 50%;  
  object-fit: cover;  
  border: 4px solid var(--primary);  
  margin-bottom: 10px;  
  background: #555;  
}
```

```
.profile-name {  
  font-size: 1.2rem;  
  font-weight: bold;  
  color: var(--primary);  
}
```

```
.profile-role {  
  font-size: 0.9rem;  
  color: #b2bec3;  
  font-style: italic;  
}
```

```
/* Navigation Buttons */
.nav-btn {
  background: transparent;
  border: none;
  color: var(--text-light);
  padding: 15px;
  text-align: left;
  cursor: pointer;
  border-radius: 8px;
  margin-bottom: 5px;
  transition: 0.2s;
  font-size: 1rem;
  display: flex;
  align-items: center;
  gap: 10px;
}

.nav-btn:hover {
  background: rgba(255,255,255,0.1);
}

.nav-btn.active {
  background: var(--primary);
  color: white;
  font-weight: bold;
}

/* --- MAIN CONTENT AREA --- */
.main-content {
  flex: 1;
  padding: 40px;
  overflow-y: auto;
  display: flex;
  flex-direction: column;
  gap: 20px;
}

.header-card {
  background: white;
```

```
padding: 30px;
border-radius: 15px;
box-shadow: 0 4px 6px rgba(0,0,0,0.05);
}
```

```
h1 { margin: 0; color: #2d3436; }
p.subtitle { color: #636e72; margin-top: 5px; }
```

```
/* The Simulation Grid */
.simulation-area {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 20px;
  flex: 1;
}
```

```
/* Info Card */
.info-card {
  background: white;
  padding: 30px;
  border-radius: 15px;
  box-shadow: 0 4px 6px rgba(0,0,0,0.05);
}
```

```
.difficulty-badge {
  display: inline-block;
  padding: 5px 12px;
  border-radius: 20px;
  font-size: 0.8rem;
  font-weight: bold;
  color: white;
  margin-bottom: 15px;
}
```

```
.diff-easy { background: #00b894; }
.diff-med { background: #fdbc6e; color: #333; }
.diff-hard { background: #d63031; }
```

```
/* The Interactive Stage */
```

```
.stage-card {
  background: #e9ecef;
  border-radius: 15px;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  position: relative;
  border: 2px dashed #cbd5e0;
  overflow: hidden;
}

.stage-items {
  position: relative;
  height: 200px;
  width: 100%;
  display: flex;
  align-items: flex-end;
  justify-content: center;
}

.nut {
  font-size: 80px;
  z-index: 1;
  transition: transform 0.1s;
}

.tool {
  font-size: 80px;
  position: absolute;
  top: -50px; /* Start high */
  opacity: 0;
  z-index: 2;
}

.crack-btn {
  margin-top: 20px;
  padding: 15px 40px;
  background: var(--sidebar-bg);
```

```

    color: white;
    border: none;
    border-radius: 8px;
    font-size: 1.1rem;
    cursor: pointer;
    box-shadow: 0 4px 0 #000;
    transition: transform 0.1s;
}

.crack-btn:active {
    transform: translateY(4px);
    box-shadow: 0 0 0 #000;
}

/* Animations */
@keyframes smash {
    0% { top: -50px; opacity: 1; transform: rotate(0deg); }
    40% { top: -80px; transform: rotate(-10deg); }
    100% { top: 60px; opacity: 1; transform: rotate(0deg); } /* Hit
position */
}

@keyframes shake {
    0% { transform: translateX(0); }
    25% { transform: translateX(-5px) rotate(-5deg); }
    75% { transform: translateX(5px) rotate(5deg); }
    100% { transform: translateX(0); }
}

@keyframes fadeOut {
    0% { opacity: 1; }
    100% { opacity: 0; }
}

/* Crumbled nut state */
.nut.cracked {
    content: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100"><text
y="80" font-size="80">🥜</text></svg>'); /* Fallback */

```

```

    filter: grayscale(100%) blur(1px);
    transform: scaleY(0.5);
}

```

```

/* Responsive */
@media (max-width: 800px) {
  body { flex-direction: column; overflow: auto; }
  .sidebar { width: 100%; flex-direction: row; align-items: center;
padding: 10px; height: auto; }
  .profile-section { border: none; margin: 0; display: flex; align-
items: center; gap: 10px; padding: 0; }
  .avatar { width: 50px; height: 50px; margin: 0; }
  .nav-list { display: none; } /* Hide nav on mobile for simplicity in
this demo */
  .simulation-area { grid-template-columns: 1fr; }
}
</style>
</head>
<body>

```

```

<!-- SIDEBAR -->
<div class="sidebar">
  <div class="profile-section">
    <!-- Your Monkey Image -->
    
    <div>
      <div class="profile-name">Prof. Monkey</div>
      <div class="profile-role">Senior Nut Engineer</div>
    </div>
  </div>

  <div style="flex:1">
    <p style="text-transform: uppercase; font-size: 0.8rem; letter-
spacing: 1px; color:#636e72; margin-bottom: 10px;">Techniques</p>
    <button class="nav-btn active" onclick="loadMethod(0)"> The
Rock Smash</button>
    <button class="nav-btn" onclick="loadMethod(1)">🔧 Tool
Assist</button>

```

```
        <button class="nav-btn" onclick="loadMethod(2)">🏔 Gravity  
Drop</button>
```

```
        <button class="nav-btn" onclick="loadMethod(3)">🦷 The  
Chomp</button>
```

```
    </div>
```

```
</div>
```

```
<!-- MAIN CONTENT -->
```

```
<div class="main-content">
```

```
    <div class="header-card">
```

```
        <h1 id="method-title">The Rock Smash</h1>
```

```
        <p class="subtitle">Select a technique from the sidebar to view  
details.</p>
```

```
    </div>
```

```
<div class="simulation-area">
```

```
    <!-- Details Column -->
```

```
    <div class="info-card">
```

```
        <span id="badge" class="difficulty-badge diff-easy">Difficulty:  
Easy</span>
```

```
        <h3 style="margin-top:0">Description</h3>
```

```
        <p id="method-desc" style="line-height: 1.6; color: #555;">
```

```
            The classic primate technique. Find a heavy stone (the  
hammer) and a flat stone (the anvil). Place the nut on the anvil and bring  
the hammer down with force.
```

```
        </p>
```

```
        <h3>Pro Tips from Prof. Monkey:</h3>
```

```
        <ul id="method-tips" style="color: #555;">
```

```
            <li>Lift with your knees, not your back.</li>
```

```
            <li>Watch your fingers!</li>
```

```
        </ul>
```

```
    </div>
```

```
<!-- Interactive Stage Column -->
```

```
<div class="stage-card">
```

```
    <div class="stage-items">
```

```
        <div class="tool" id="tool-emoji"> </div>
        <div class="nut" id="target-nut"> 🌰 </div>
    </div>
    <button class="crack-btn" onclick="runSimulation()">CRACK
IT!</button>
    <div id="status-msg" style="height: 20px; margin-top: 10px;
color: var(--primary); font-weight: bold;"></div>
</div>
```

```
</div>
</div>
```

```
<script>
    // Data for different methods
    const methods = [
        {
            title: "The Rock Smash",
            desc: "The most ancient and reliable method. Utilizing a heavy
stone to apply blunt force trauma to the shell. Requires good aim.",
            difficulty: "Easy",
            diffClass: "diff-easy",
            tips: ["Lift with your knees.", "Don't smash your own fingers."],
            tool: " ",
            nut: " 🌰 "
        },
        {
            title: "Tool Assist",
            desc: "Using advanced human technology (a hammer) to breach
the hull. Highly effective but requires finding a hardware store.",
            difficulty: "Medium",
            diffClass: "diff-med",
            tips: ["Hold the handle at the end for leverage.", "Wear safety
goggles."],
            tool: " 🛠️ ",
            nut: " 🍂 " // Chestnut
        },
        {
            title: "Gravity Drop",
```

```

        desc: "Climb a tree or a cliff. Drop the nut onto a hard surface.
Let physics do the work. Energy efficient!",
        difficulty: "Medium",
        diffClass: "diff-med",
        tips: ["Ensure no one is standing below.", "Harder shells need
higher drops."],
        tool: "↓ ", // Represents falling
        nut: "🥥" // Coconut
    },
    {
        title: "The Chomp",
        desc: "Using pure jaw strength. Only recommended for softer
shells or monkeys with dental insurance.",
        difficulty: "Hard",
        diffClass: "diff-hard",
        tips: ["Brush twice a day.", "Don't try this on a coconut."],
        tool: "🦷",
        nut: "🍌"
    }
];

```

```

let currentIndex = 0;
const toolEl = document.getElementById('tool-emoji');
const nutEl = document.getElementById('target-nut');
const statusMsg = document.getElementById('status-msg');

```

```

function loadMethod(index) {
    currentIndex = index;
    const data = methods[index];

    // Update UI Text
    document.getElementById('method-title').innerText = data.title;
    document.getElementById('method-desc').innerText = data.desc;

    // Update Badge
    const badge = document.getElementById('badge');
    badge.className = 'difficulty-badge ' + data.diffClass;
    badge.innerText = 'Difficulty: ' + data.difficulty;
}

```

```

// Update Tips
const tipsList = document.getElementById('method-tips');
tipsList.innerHTML = "";
data.tips.forEach(tip => {
    const li = document.createElement('li');
    li.innerText = tip;
    tipsList.appendChild(li);
});

// Update Visuals
toolEl.innerText = data.tool;
nutEl.innerText = data.nut;
resetStage();

// Update active button state
const btns = document.querySelectorAll('.nav-btn');
btns.forEach((btn, i) => {
    if(i === index) btn.classList.add('active');
    else btn.classList.remove('active');
});
}

function resetStage() {
    nutEl.style.transform = "scaleY(1)";
    nutEl.style.filter = "none";
    nutEl.innerText = methods[currentIndex].nut;
    toolEl.style.opacity = "0";
    toolEl.style.animation = "none";
    statusMsg.innerText = "";
}

function runSimulation() {
    resetStage();

    // 1. Animate Tool
    toolEl.style.opacity = "1";
    // Trigger reflow to restart animation
    void toolEl.offsetWidth;
    toolEl.style.animation = "smash 0.6s ease-in forwards";

```

```
// 2. Animate Nut Impact
setTimeout(() => {
    nutEl.style.animation = "shake 0.2s linear";

    // 3. Crack Result
    setTimeout(() => {
        nutEl.style.transform = "scaleY(0.4) rotate(10deg)";
        nutEl.style.filter = "grayscale(100%)";
        nutEl.innerText = "💣"; // Explosion or broken bits

        // Hide tool
        toolEl.style.animation = "fadeOut 0.2s forwards";

        // Success Message
        statusMsg.innerText = "CRACKED SUCCESSFULLY!";
    }, 200);

}, 350); // Timing matches the 'hit' of the smash animation
}

// Initialize
loadMethod(0);

</script>
</body>
</html>
```