

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>GRID RUNNER | High Voltage Crash</title>
  <style>
    body { background: #050505; color: #00d4ff; font-family: 'Segoe UI', sans-serif; text-align: center; margin:
0; overflow: hidden; }

    #loadingScreen { position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: #000; z-index:
10000; display: flex; justify-content: center; align-items: center; transition: 0.5s; }

    .loader-content { width: 80%; max-width: 400px; }

    .progress-bar { width: 100%; height: 8px; background: #222; border-radius: 4px; overflow: hidden; margin-
top: 15px; }

    #progressFill { width: 0%; height: 100%; background: #fbff00; transition: 0.2s; }

    .stats-bar { background: #111; padding: 15px; border-bottom: 2px solid #00d4ff; display: flex; justify-
content: space-around; }

    #multiplier { font-size: 80px; font-weight: bold; margin: 40px 0; text-shadow: 0 0 20px #00d4ff; }

    .game-area { height: 200px; position: relative; border-bottom: 1px solid #333; margin-bottom: 20px; }

    #voltageLine { fill: none; stroke: #fbff00; stroke-width: 3; filter: drop-shadow(0 0 5px #fbff00); }

    .controls { background: #111; padding: 20px; border-radius: 20px; display: inline-block; border: 1px solid
#333; }

    input { background: #000; border: 1px solid #00d4ff; color: #fff; padding: 10px; width: 80px; border-
radius: 5px; text-align: center; }

    button { padding: 15px 40px; font-size: 18px; font-weight: bold; border-radius: 10px; cursor: pointer;
border: none; transition: 0.2s; margin-top: 10px; width: 100%; }

    .btn-bet { background: #00d4ff; color: #000; }

    .btn-cashout { background: #fbff00; color: #000; }

    .history { display: flex; justify-content: center; gap: 8px; margin-top: 20px; }

    .h-item { padding: 4px 10px; border-radius: 15px; font-size: 12px; font-weight: bold; background: #222; }

    .low { color: #ff4444; } .high { color: #00ff00; }

    .modal { position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.9); display: flex;
justify-content: center; align-items: center; z-index: 9000; }

    .modal-content { background: #111; padding: 25px; border: 1px solid #00d4ff; border-radius: 15px; max-
width: 300px; }
  </style>
</head>
<body>

  <div id="loadingScreen">
    <div class="loader-content">
      <h1 style="color:#fbff00; letter-spacing:4px;">GRID RUNNER</h1>
      <div class="progress-bar"><div id="progressFill"></div></div>
    </div>
  </div>
</body>
</html>
```

```

        <p style="font-size:12px; color:#555;">STABILIZING VOLTAGE...</p>
    </div>
</div>

<div id="helpModal" class="modal">
    <div class="modal-content">
        <h3 style="color:#fbff00;">SYSTEM BRIEFING</h3>
        <p style="font-size:14px; text-align:left;">1. Place your bet.<br>2. Watch the multiplier rise.<br>3.
DISCONNECT before the grid trips (Blackout)!</p>
        <button class="btn-bet" onclick="closeHelp()">INITIALIZE</button>
    </div>
</div>

<div class="stats-bar">
    <div>BALANCE<br><span id="balanceDisplay" style="color:#fff;">R1000.00</span></div>
</div>

<div class="game-area">
    <svg width="100%" height="100%"><polyline id="voltageLine" points="0,200"></polyline></svg>
    <div id="multiplier">1.00x</div>
</div>

<div class="controls">
    <div style="display:flex; justify-content: space-between; gap: 10px;">
        <div>Bet: <input type="number" id="betAmount" value="10"></div>
        <div>Auto: <input type="number" id="autoCash" value="2.00"></div>
    </div>
    <button id="mainBtn" class="btn-bet" onclick="handleAction()">PLACE BET</button>
</div>

<div id="historyLog" class="history"></div>

<script>
    let balance = 1000;
    let isRunning = false;
    let currentMultiplier = 1.0;
    let crashPoint = 0;
    let gameLoop;
    let history = [];

    // LOAD SCREEN
    window.onload = () => {
        let p = 0;
        let inv = setInterval(() => {
            p += 5;

```

```
document.getElementById('progressFill').style.width = p + "%";
if(p >= 100) {
    clearInterval(inv);
    setTimeout(() => document.getElementById('loadingScreen').style.display = 'none', 500);
}
}, 50);
};
```

```
function closeHelp() { document.getElementById('helpModal').style.display = 'none'; }
```

```
function handleAction() {
    if (!isRunning) startRound();
    else cashOut();
}
```

```
function startRound() {
    let bet = parseFloat(document.getElementById('betAmount').value);
    if (bet > balance || bet <= 0) return alert("Invalid Bet");

    balance -= bet;
    updateUI();

    isRunning = true;
    currentMultiplier = 1.0;
    crashPoint = 0.97 / (1 - Math.random());
    if (crashPoint < 1) crashPoint = 1.02;

    document.getElementById('mainBtn').innerText = "DISCONNECT";
    document.getElementById('mainBtn').className = "btn-cashout";
    document.getElementById('voltageLine').setAttribute('points', "0,200");

    runEngine();
}
```

```
function runEngine() {
    if (!isRunning) return;

    gameLoop = setTimeout(() => {
        currentMultiplier += 0.01 * Math.pow(currentMultiplier, 0.5);
        document.getElementById('multiplier').innerText = currentMultiplier.toFixed(2) + "x";

        // Draw Line
        let x = (currentMultiplier - 1) * 60;
        let y = 200 - (Math.random() * 20 + (currentMultiplier * 5));
        let pts = document.getElementById('voltageLine').getAttribute('points');
```

```

document.getElementById('voltageLine').setAttribute('points', pts + ` ${x},${y}`);

// Auto Cashout
let auto = parseFloat(document.getElementById('autoCash').value);
if (auto > 1 && currentMultiplier >= auto) { cashOut(); return; }

if (currentMultiplier >= crashPoint) blackout();
else runEngine();
}, 80);
}

function cashOut() {
  let bet = parseFloat(document.getElementById('betAmount').value);
  balance += bet * currentMultiplier;
  endRound(true);
}

function blackout() {
  document.body.style.background = "#ff000033";
  setTimeout(() => document.body.style.background = "#050505", 200);
  document.getElementById('multiplier').innerText = "TRIPPED";
  endRound(false);
}

function endRound(win) {
  isRunning = false;
  clearTimeout(gameLoop);
  updateHistory(currentMultiplier);
  updateUI();
  document.getElementById('mainBtn').innerText = "PLACE BET";
  document.getElementById('mainBtn').className = "btn-bet";
}

function updateUI() {
  document.getElementById('balanceDisplay').innerText = "R" + balance.toFixed(2);
}

function updateHistory(val) {
  history.unshift(val);
  if (history.length > 5) history.pop();
  let log = document.getElementById('historyLog');
  log.innerHTML = history.map(v => `<span class="h-item ${v >= 2 ? 'high' : 'low'}">${v.toFixed(2)}x</span>`).join("");
}
</script>

```

</body>

</html>