```
const Queue = require('bull');
const taskQueue = new Queue('engagement-tasks', {
  redis: { host: '127.0.0.1', port: 6379 }
});
const db = require('./db');

// Simulated platform APIs
const platformApis = {
  instagram: async (task) => simulateEngagement(task, 'Instagram'),
  tiktok: async (task) => simulateEngagement(task, 'TikTok'),
  facebook: async (task) => simulateEngagement(task, 'Facebook')
};

async function simulateEngagement(task, platform) {
  const { task_id, action_type, quantity, target_url } = task;
  console.log(`Simulating ${action_type} on ${platform} for ${target_url}: ${quantity}`);

  // Simulate rate limiting
  await new Promise(resolve => setTimeout(resolve, Math.random() * 2000));

  // Simulate success/failure (30% chance of detection)
  const isDetected = Math.random() < 0.3;
  const status = isDetected ? 'failed' : 'completed';
  const message = isDetected
    ? `Task detected by ${platform} systems (e.g., IP clustering, pattern analysis)`
    : `Task completed (simulated)`;

  // Update task status
  db.run(`UPDATE tasks SET status = ? WHERE id = ?`, [status, task_id]);
  return { status, message };
}

// Process tasks
taskQueue.process(async (job) => {
  const { task_id, platform, action_type, quantity, target_url } = job.data;
  const api = platformApis[platform.toLowerCase()];
  if (!api) throw new Error(`Unsupported platform: ${platform}`);
```

```
  try {
    const result = await api({ task_id, action_type, quantity, target_url });
    console.log(`Task ${task_id}: ${result.message}`);
    return result;
  } catch (error) {
    db.run(`UPDATE tasks SET status = ? WHERE id = ?`, ['failed', task_id]);
    console.error(`Task ${task_id} failed: ${error.message}`);
    return { status: 'failed', message: error.message };
  }
});

// Handle task completion
taskQueue.on('completed', (job, result) => {
  console.log(`Task ${job.id} completed: ${result.message}`);
});

taskQueue.on('failed', (job, err) => {
  console.error(`Task ${job.id} failed: ${err.message}`);
});

console.log('Worker started, waiting for tasks...');
```