

Mission Guide: GigSniper

Automated Freelance Income Generator – Step-by-Step Build Plan

- Assigned to: Jimmy
- Skills Used: Python, Web Scraping, GPT, Notion API, CLI Automation
- Bonus: Unity & C# background opens up possibilities for desktop wrappers later.

1. Environment Setup

- 1 Install Python 3.11+ and pip on your machine.
- 2 Install required libraries: ``pip install openai requests beautifulsoup4 notion-client``
- 3 Create a folder for the project: ``gig-sniper`` and set up a basic Git repo.
- 4 Create a ``.env`` file to store your API keys and private config safely.
- 5 Use Notion to create a Kanban DB with columns: Job Title, Platform, Status, Outcome, Earnings.

2. Scraper Bot Core

- 1 Use BeautifulSoup or requests to pull latest jobs from Upwork and Freelancer.
- 2 Filter jobs by payout \geq \$50 and proposals $<$ 20 (scrape job cards & job detail pages).
- 3 Save each matching gig to a local cache or JSON file with its ID to avoid duplicates.
- 4 Create a 'watchdog' loop that runs every 5–10 mins and updates the cache.
- 5 Log the number of jobs found per scan and alert if no results are found for 3+ cycles.

3. GPT-Generated Cover Letters

- 1 Use GPT-4 API to generate short, tailored proposals using job title, skills required, and budget.
- 2 Use a prompt template like: 'Write a friendly but professional Upwork proposal for this job: [description]'.
- 3 Save the generated text alongside each job in the JSON cache for review.
- 4 Later, allow the bot to auto-approve and submit if it passes certain filters.
- 5 C# Tip: You can later build a Unity frontend app to review/edit letters and track earnings visually.

4. Bid Submission Logic

- 1 Use browser automation (like Selenium) to log into your Upwork/Freelancer accounts.
- 2 Navigate to the gig page, paste the GPT cover letter, and set bid price (fixed or hourly).
- 3 Track which jobs were submitted and the time of submission.
- 4 Use C# or Unity later to make a desktop dashboard for click-to-submit or batch-schedule logic.
- 5 Optional: rotate between accounts or proxies if you scale later.

5. Notion Sync & Analytics

- 1 Use Notion's API to push every job into a Kanban board: Submitted, Waiting, Replied, Hired.
- 2 Log job earnings when hired and push that to a 'Revenue by Day' Notion view.
- 3 Set up a daily Notion page update summarizing new jobs, earnings, wins/losses.
- 4 Optional: auto-generate graphs using Charty or pull into Google Sheets for deeper insight.

Let's build something legendary. Start this today, and we'll automate income by tomorrow.