

Social Media Automation Simulator

This project simulates the structure and behavior of illegitimate social media automation tools for educational purposes. It does NOT interact with real platforms and is designed to help study how such tools operate and why they are detected.

Setup

1. **Install Node.js** (v16 or higher).
2. **Install Redis** (required for task queuing).
 - On macOS: `brew install redis`
 - On Ubuntu: `sudo apt-get install redis-server`
 - On Windows: Download from <https://redis.io/download>
3. Clone the repository and navigate to the project folder.
4. Run `npm install` to install dependencies.
5. Start the server: `node server.js`
6. Start the worker: `node worker.js`
7. Access the app at `http://localhost:3000`.

Features

- **Frontend**: User and admin dashboards to submit and manage simulated tasks.
- **Backend**: Express server with mock APIs for task submission and user management.
- **Database**: SQLite (in-memory) for storing users and tasks.
- **Task Queue**: Bull queue with Redis to simulate asynchronous task processing.
- **Worker**: Simulates engagement tasks with random success/failure to mimic platform detection.

Usage

- **Login**: Use `admin/admin123` for admin access or `user1/pass123` for user access.
- **Submit Tasks**: Select a platform, action type, and quantity to simulate engagement.
- **Monitor Tasks**: View task status and simulated outcomes in the dashboard.
- **Admin Dashboard**: Manage all tasks and update their status.

Educational Insights

The simulator demonstrates:

- How task queues manage asynchronous actions.
- Mock API calls with rate limiting and failure simulation.
- Platform detection mechanisms (simulated as random failures).
- Credit-based systems for task submission.

****Warning****: Real automation tools violate platform Terms of Service and are unethical. This simulator is for study only.

Dependencies

- Express: Web server
- SQLite3: Mock database
- Bull: Task queuing
- Bootstrap: Frontend styling