# **Automation System Documentation**

This folder (src/automation/) contains automation runners built on **Playwright** that handle Facebook login, session persistence, and automated job posting in Facebook groups.

## **Files Overview**

### **1. runner.js**

Core automation runner that manages:

* Launching a Playwright browser.
* Managing per-user browser contexts.
* Ensuring Facebook login.
* Running automation tasks (default or custom).

**Key Responsibilities**

* Initializes a global browser (chromium) with safe configs.
* Creates and switches **contexts** per user (so multiple accounts can be used safely).
* Ensures the user is logged in (via ensureLoggedIn).
* Runs automation tasks:  
  + Either a provided function (automationTasks).
  + Or default: verify Facebook homepage.
* Provides cleanup logic to gracefully close browser and contexts.
* Includes a **defaultAutomationTasks** example that checks profile name.

**Exports**

* runAutomation(credentials, automationTasks?) → Runs automation for a given user.
* cleanup() → Closes browser & contexts.
* defaultAutomationTasks(page, context) → Sample profile-checking task.

### **2. facebook-login.js**

Handles **secure login** and **session persistence** for Facebook.

**Key Responsibilities**

* Manages cookie storage under cookies/ folder.
* Loads valid cookies on startup to avoid repeated logins.
* Detects expired sessions and falls back to fresh login.
* Handles **2FA / checkpoint challenges** with manual completion fallback.
* Normalizes cookie data for Playwright compatibility.

**Functions**

* saveCookiesToStorage(context, email) → Saves session cookies.
* loadCookiesFromStorage(context, email) → Restores cookies if valid.
* hasFacebookSession(context) → Checks if user has a valid FB session.
* detectChallenges(page) → Detects if login hit 2FA/checkpoint.
* ensureLoggedIn({ page, context }) → Main login flow with cookie validation + fallback login.
* ensureLoggedInWithStore(...) → Wrapper for compatibility.

### **3. job-post-runner-enhanced.js**

Automates **job posting to Facebook groups** with robust error handling.

**Key Responsibilities**

* Manages a separate browser + context for job posting.
* Navigates to group pages with retries & crash recovery.
* Formats job post content (title, location, description, requirements, perks).
* Generates **unique Messenger links** for applications.
* Creates posts in groups using multiple selectors for robustness.
* Adds **immediate self-replies** to engage candidates (optional).
* Uses **Prisma** to store job data & maintain consistency.

**Functions**

* initializeJobPostBrowser() → Launches a lightweight browser.
* getJobPostContextForUser(browser, email) → Creates per-user posting context.
* navigateToGroupSafely(page, groupUrl) → Resilient navigation to FB groups.
* isPageLikelyRecovered(page) → Detects if page crashed or recovered.
* generateMessengerLink(jobData, jobPostId) → Encodes job info into contextual link.
* formatJobPost(job, jobPostId) → Creates post text with job details.
* createGroupPost(page, jobContent) → Posts formatted job to group.
* addImmediateSelfReply(page, jobData, jobPostId) → Adds comment under the post.

## **How It All Works**

1. **Runner Bootstraps Browser**
   * runner.js starts a Chromium instance and user context.
2. **Login Flow**
   * facebook-login.js ensures session is active:  
     + Loads cookies (if available).
     + Otherwise, logs in with credentials.
     + Saves refreshed cookies.
3. **Automation Tasks**
   * Default: verify Facebook home/profile.
   * Custom tasks: Passed into runAutomation.
4. **Job Posting Flow**
   * job-post-runner-enhanced.js launches its own browser.
   * Navigates to FB group safely.
   * Creates a **job post with contextual Messenger link**.
   * Optionally adds self-reply for engagement.