

# UTSAV V CHAUDHARY

utsavmaan28@gmail.com | portfolio | github/UttU28

## Projects

---

### LinkedIn Reverse Search: Client Project

**Developed a LinkedIn Scraping Bot:** Utilized **Python**, **Selenium**, and **Requests** for efficient LinkedIn data scraping.

**Excel Integration:** Designed a pipeline to process input Excel sheets with headers (**firstName**, **lastName**, **companyName**) and output enriched data with additional details.

**LinkedIn Profile URL Extraction:** Automated the extraction and validation of LinkedIn profile URLs for individuals listed in the Excel file using **Proxies** and **Routing in Selenium**, enabling seamless data collection.

**Scraping:** Extracted experience details, including **Company Name**, **Position Title**, **Start Date**, & **End Date** for each job.

**SalesQL API Integration:** Implemented **SalesQL API** to retrieve verified contact email addresses for each individual.

**Data Enrichment Pipeline:** Built a **pipeline** and **threading** to manage **parallel data processing**, for LinkedIn and SalesQL.

**Database Optimization:** Maintained a database of previously scraped data to minimize redundant computations and speed up processing for future requests. On new Excel sheet, it scans database for existing data improving performance.

**Next.js Web Application:** Developed a **client-facing web application** for **user authentication** and **Excel uploads**, integrating server-side scraping and database query pipelines using **Firestore DB** for login & data.

**Email Delivery via SMTP:** Set up a **local SMTP server** for secure & automated email delivery of Excel file to the recipient.

**Comprehensive Data Output:** Produced Excel sheets enriched with fields including **firstName**, **lastName**, **linkedInURL**, **contactEmail**, **companyTitle**, **companyName**, and **companyURL**.

### Eventbrite & LUMA Scraping: Client Project

**Automated Event Scraping:** Developed a **Python bot** to scrape new events from **Eventbrite & LUMA platforms** every hour.

**Comprehensive Data Collection:** Extracted key event details, including **Event Name**, **Description**, **Title**, **startDate**, **endDate**, **eventUrl**, **registrationUrl**, and **speakersList**.

**Data Cleaning & Preprocessing:** Automated data **cleaning**, **filtering** & **preprocessing** in **Python** for relevant data extraction.

**SQL Database Design:** Designed an efficient **SQL schema** to store and manage structured event data seamlessly.

**Custom Keyword Filtering:** Integrated a dynamic **filtering system** to identify events related to specific client-required topics such as **Cryptocurrency**, **RWA**, and **Tokenization**, and filtering based on **Date Time** and **Location**.

**Web Dashboard:** Created a **website** to display event data, adding **CRM functionalities** for event tracking & management.

**Event Tracking Automation:** Automated the **ingestion pipeline** to handle continuous updates from event sources efficiently.

**Client-Specific Features:** Delivered tailored solutions for **keyword-based event categorization** and management to meet client-specific business goals.

### Instagram Teaching Reels' Automation - Personal Project

**Developed a Python-based pipeline** to dynamically **create and upload English teaching reels from scratch** on **Instagram**.

**PlayPhrase.me Integration:** Automated raw video collection and downloading by leveraging **PlayPhrase.me API**, storing initial raw videos on **Google Drive** for free storage.

**Advanced Video Editing:** Utilized **FFMPEG** for video editing, including overlays, image insertion, and transitions, combined with **Photoshop** for logo and overlay design. Creating a **preset** for all the videos to use.

**AI-Powered Subtitles:** Integrated **WhisperAI** to generate accurate subtitles with precise timestamps for **audio captions**.

**Cloud Optimization:** Stored processed video URLs in **Azure Blob Storage** for efficient & scalable video management.

**End-to-End Workflow Automation:** Orchestrated the entire video creation and deployment pipeline using **Azure DevOps** and **CRON jobs** for a fully human-less, automated bot workflow.

**Automated Instagram Uploads:** Leveraged **Selenium** to automate Instagram video uploads, employing **Docker containers** on **Ubuntu OS** to run scripts seamlessly. Marking **English Words used** and created a **Preview Dashboard** for personal viewing.

**Session Management:** **Session persistence** by storing & reusing Chrome data to avoid repeated logins across all sessions.

### Movie Controller Application - Personal Project

**Cross-Platform Media Controller:** Developed a **Python-based client-server application** enabling remote control of a PC for streaming services like **YouTube**, **Netflix**, **Prime Video**, and **FMovies** to simplify media control through **Phone** or **Smart Watch**.

**Client-Side Interface:** Designed a **Smart TV remote-style interface** using **Vite** and **React** for intuitive interaction.

**Remote Communication:** Configured **POST requests** from the **React client** to a **Flask** for seamless command transmission.

**Device Automation:** Integrated **Python libraries** like **PyAutoGUI** for emulating PC controls, including typing, searching, and browser manipulation, allowing control of these streaming platforms' websites.

**Screen Object Detection:** Leveraged **OpenCV (CV2)** for image recognition to identify screen objects and perform targeted actions like click, drag, hotkey presses, and typing. Performs actions as it pops (e.g., skipping intro, popup & redirecting ads).

**Ease of Access:** Made the **React client** accessible via web browsers, allowing users to control their PC from anywhere within the network.

**Innovative Remote Experience:** Enabled users to control movies and media streaming platforms with convenience and automation directly from their **personal devices**.