Adviser Automation Express: CSV-Powered Multi-Recipient Messaging System

PROF. ARUNADEVI KHAPLE[1], SUYASH GANESHKAR[2], MANISH FALE[3], AKHILESH ADHE[4], SARANG DHANAVE[5]

[1] Asst. Professor. Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

[2] B. E (Computer Engineering), Savitribai Phule Pune University, Zeal College of Engineering and Research, Pune, Maharashtra, India

***Abstract*— In the fast-paced world of digital advertising, the ability to connect with target audiences is crucial for success. Advertisers are always on the lookout for new ways to make their campaigns more efficient, reduce costs, and create personalized messages that resonate with consumers. While traditional methods work, they often can't keep up with the demands of modern advertising. This has led to a growing interest in automation as a solution. Adviser automation as a transformative tool for advertisers. Specifically, it focuses on the development and implementation of a CSV integrated message sender, a solution aimed at simplifying the intricate process of reaching audiences at scale.**

***Index Terms*— Python, Selenium, Automation, Testing, Chrome WebDriver, Search Engine Optimization (SEO), Social Media Optimization (SMO).**

I. INTRODUCTION

In today's ever-evolving world of communication technologies, there's a pressing need for messaging systems that are both efficient and scalable. Enter "Adviser Automation Express," a notable solution that leverages CSV (Comma-Separated Values) to power a Multi-Recipient Messaging System. This research paper aims to explore the intricacies of this innovative approach.

While manual communication methods have long been the backbone of advertising, they often struggle to keep up with the demands of modern digital marketing. With today's sprawling advertising campaigns targeting diverse audience segments and requiring real-time responsiveness, traditional methods fall short. This has created a gap between advertisers' goals and what traditional approaches can deliver.

In response, the advertising industry is undergoing a transformation, turning to automation as an appealing alternative. This research paper dives deep into this shift, highlighting the transformative potential of Python automation as a crucial tool for advertisers. Specifically, it zeroes in on the development and implementation of a CSV integrated message sender, meticulously crafted to simplify the process of engaging audiences at scale.

By harnessing the power of Python, advertisers can revolutionize their communication strategies, making campaigns more efficient, targeted, and ultimately, more successful. The introductory section of the paper outlines the core challenges faced by advertisers today, stressing the need for streamlined communication channels. It emphasizes the limitations of manual methods and showcases the rising demand for automation to overcome these hurdles. Furthermore, it presents Python as the perfect programming language for automation tasks due to its versatility, robust libraries, and user-friendly nature.

As the paper progresses, it will provide a comprehensive breakdown of how to construct a CSV integrated message sender using Python. It will delve into its capabilities, technical aspects, and practical applications in real-world scenarios. By the end of this research, advertisers will gain valuable insights into how Python automation can enhance their advertising efforts, leading to more efficient and effective campaigns

A. Motivation

The primary motivation for this research is to automate the process of sending messages to target audiences. By building a Python script that integrates with a CSV file, advertisers can save a significant amount of time and effort. Sending messages individually or manually inputting recipient details can be a tedious task. With automation, advertisers can streamline their workflow and allocate more time to strategic planning and campaign optimization. Automation ensures accuracy and minimizes potential human errors. By integrating with a CSV file, advertisers can maintain a clean and up-to-date database of recipients. This reduces the risk of sending messages to incorrect or outdated contacts, optimizing campaign reach and relevance. In addition, Python offers robust error handling and exception handling capabilities, ensuring smooth execution and reliable results.

B. Objective

The objective of this research is to suggests a system that automates and expedites messaging tasks, with a specific focus on sending messages to multiple recipients using data from CSV files. to design a simple and intuitive user interface that allows advertisers to easily navigate through the tool, import CSV files, customize message content, and start the automated message sending process with a single click. to enable advertisers to create personalized messages by utilizing variables within the CSV file. This will allow for dynamic content, making each message unique and relevant to the recipient. to develop an automatic message sender module that extracts the necessary information from the CSV file, constructs personalized messages, and sends them to the recipients without any manual intervention.

II. METHODS AND MATERIAL

A. System Architecture

1) Selenium WebDriver Integration:

Utilize Selenium WebDriver to automate the interaction with Adviser Automation Express through web browsers. Write test scripts in a programming language (e.g., Java, Python) to simulate user interactions and validate the behaviour of the messaging system.

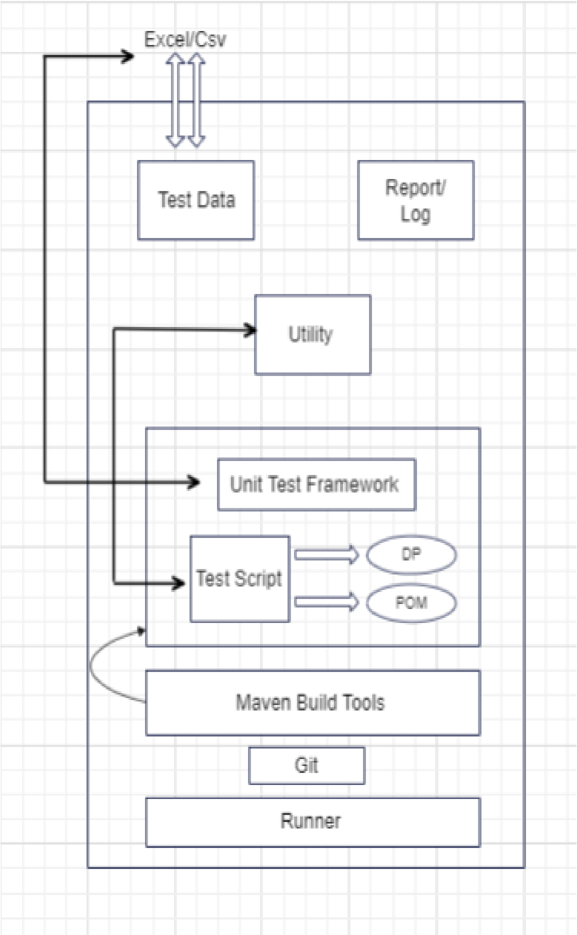


Fig.2.1 System Architecture

2) Test Scenarios:

Define test scenarios that cover various aspects of Adviser Automation Express, including CSV parsing, multi-recipient messaging, and automation logic. Use Selenium to simulate user actions such as CSV file uploads, message generation, and interaction with the messaging module.

3) Locators and Web Elements:

Identify and use Selenium locators (e.g., XPath, CSS selectors) to interact with web elements relevant to CSV file handling and messaging features in Adviser Automation Express. Create a robust set of locators for key elements like buttons, input fields, and dynamic content.

4) Page Object Model (POM):
Implement the Page Object Model to structure Selenium test scripts in a modular way. Create page classes representing different sections of Adviser Automation Express, encapsulating the interaction logic within these classes for better maintainability.

5) Data-Driven Testing with CSV:
Leverage Selenium for data-driven testing by integrating CSV files as test data sources. Create test scenarios that involve uploading CSV files and verify the correct handling of CSV data within the messaging system.

6) TestNG Integration:
Integrate TestNG with Selenium to manage and execute test cases efficiently. Utilize TestNG annotations for configuration, parallel execution, and grouping of test cases, providing better control over the testing process.

7) Continuous Integration (CI) Pipeline:
Incorporate Selenium tests into the CI pipeline for Adviser Automation Express. Use CI tools like Jenkins or others to automatically trigger Selenium tests on code changes, ensuring continuous validation of the messaging system's functionality.

B. CSV-Powered Messaging Module

1) Purpose:
The CSV-Powered Messaging Module is designed to process CSV (Comma-Separated Values) data, extract relevant information, and utilize it for initiating multi-recipient messaging within the Adviser Automation Express system.
Integration Points: This module is tightly integrated into the system's overall architecture, connecting with other components to seamlessly incorporate CSV data into the messaging process.

2) Functionality:
CSV Parsing: The module likely includes a CSV parsing mechanism to extract relevant information from CSV files. This involves reading and interpreting the structured data within the files.

Data Mapping: After parsing, the module might map the extracted data to specific messaging parameters. This step ensures that the information from the CSV files is correctly associated with the messaging content.

Multi-Recipient Messaging: The core functionality is the ability to send messages to multiple recipients simultaneously. The module might have features for customizing messages based on the information from the

3) Recipient Categorization:
Criteria for Recipient Identification: Define the criteria used to identify messaging recipients based on the CSV data. This could involve specific fields, values, or conditions that categorize users for targeted messaging.

Segmentation Strategies: Explore any segmentation or targeting strategies employed to categorize recipients effectively.

4) User Interface (UI):
A user interface would be essential for users to interact with the CSV-Powered Messaging Module. This UI include features for uploading CSV files, configuring messaging parameters, and monitoring the status of messaging processes. Visual elements, such as progress bars or status indicators, could provide users with real-time feedback on the processing of CSV files and the execution of messaging tasks.

Integration with Messaging Logic: Communication with Messaging System: the CSV-Powered Messaging Module communicates with the overall messaging system. This includes the seamless transfer of validated data to initiate the messaging process.

5) Optimizations for Performance:
Efficiency Measures: Outline any measures taken to optimize the performance of the CSV-Powered Messaging Module. This includes caching strategies, parallel processing, or other techniques to handle large datasets efficiently.

6) Security Considerations:
Data Security: Address how the module ensures the security of CSV data during the parsing and messaging processes. encryption, access controls, and other security measures in place.

User Privacy: Protect user privacy, especially when dealing with sensitive information in the CSV files.

7) Error Handling:
Robust error handling mechanisms are essential. This module able to identify and manage errors during CSV parsing, data mapping, and messaging processes, providing clear feedback to users.

C. Potential Workflow

1) User Uploads CSV File: Users upload one or more CSV files containing recipient information and messaging parameters.

2) CSV Parsing and Data Mapping: The module parses the CSV files, extracts relevant information, and maps the data to messaging parameters.

3) Configuration and Automation: Users configure messaging settings, potentially defining automation rules based on CSV data.

4) Multi-Recipient Messaging: The module automates the process of sending messages to multiple recipients according to the configured parameters and rules.

5) Logging and Reporting: The system logs the messaging activities and generates reports for users to review.

6) Feedback and Notifications: Users receive feedback on the status of messaging processes, including successful deliveries and any encountered issues.

D. Selenium

1) Selenium Overview:
Selenium is an open-source tool for automating web browsers. It provides a way to interact with web elements, simulate user actions, and automate testing scenarios. Selenium includes different components such as Selenium WebDriver, Selenium Grid, Selenium IDE, and Selenium Server.

2) Selenium WebDriver:
Definition: Selenium WebDriver is the primary tool in the Selenium suite. It provides a programming interface to drive the browser, WebDriver enables the automation of interactions with web elements like clicks, form submissions, and data extraction.

3) Selenium Grid:
Definition: Selenium Grid allows for parallel test execution on multiple machines, enhancing test scalability, It's particularly useful for testing on various browser and platform combinations simultaneously.

4) Selenium IDE:
Definition: Selenium IDE is a browser extension that facilitates record-and-playback testing. It's less commonly used for complex test scenarios but can be helpful for quick test creation, Selenium IDE is often used for exploratory testing and for generating initial test scripts.

5) Language Bindings:
Programming Languages: Selenium supports multiple programming languages, including Java, Python, C#, Ruby, and JavaScript, The choice of language depends on the preferences and expertise of the testing team or individual.

6) Selenium and Web Applications:
Testing Web Applications: Selenium is widely used for testing web applications across different browsers to ensure compatibility and functionality, It facilitates cross-browser testing to identify and address issues specific to different browsers.

7) Selenium and Automated Testing:
Automated Testing Benefits: Selenium is popular for automating repetitive and time-consuming testing tasks, allowing faster and more reliable testing processes, It's especially useful for regression testing to ensure that new code changes don't break existing functionality.

8) Selenium Scripting:
Scripting Capabilities: Selenium scripts are written in programming languages supported by WebDriver bindings, Selenium allows the identification of HTML elements on a webpage using various locators, such as ID, class name, XPath, etc.

9) Selenium Testing Process:

Setup: Configuring the Selenium WebDriver and necessary dependencies.

Navigation: Navigating to web pages and interacting with elements.

Assertions and Verification: Checking expected outcomes against actual results.

Tear Down: Closing the browser and releasing resources.

10) Selenium and Continuous Integration (CI):
CI Integration: Selenium tests can be integrated into CI/CD pipelines to automate testing on code changes.
Jenkins, Travis CI: CI tools like Jenkins or Travis CI are commonly used for Selenium test automation.

11) Challenges:
Dynamic Web Elements: Handling dynamically changing elements on a webpage.

Flakiness: Addressing test flakiness caused by factors like network latency or asynchronous behavior.

12) Best Practices:
Modular Test Design: Adopting a modular design for test scripts to enhance maintainability.

Explicit Waits: Using explicit waits to handle synchronization issues in dynamic web applications.

E. Applications

1) Efficient Communication with Clients and Team Members: Advisor Automation Express facilitates seamless communication by allowing you to send personalized messages to multiple recipients simultaneously. Whether you need to update clients on the progress of their SEO campaigns or coordinate SMO strategies with your team, this platform ensures that your messages are delivered promptly and effectively.
2) Automated Reporting and Analytics: With Advisor Automation Express, you can automate the process of generating SEO and SMO reports, providing your clients and stakeholders with valuable insights into the performance of their digital marketing campaigns. By integrating with analytics tools, this platform allows you to track key metrics, identify trends, and make data-driven decisions to optimize your strategies.
3) Targeted Outreach and Engagement: Leveraging Advisor Automation Express's advanced targeting capabilities, you can tailor your messages to specific audience segments based on their interests, demographics, and online behavior. Whether you're promoting a new product or sharing valuable content, this personalized approach enables you to engage with your target audience more effectively, driving traffic and conversions.
4) Social Media Management: Advisor Automation Express simplifies the process of managing multiple social media accounts by providing a centralized platform for scheduling posts, monitoring engagement, and analyzing performance metrics. Whether you're running a Facebook ad campaign or launching a Twitter hashtag campaign, this platform empowers you to streamline your SMO efforts and maximize your social media presence.
5) Campaign Optimization and A/B Testing: By leveraging Advisor Automation Express's A/B testing capabilities, you can experiment with different SEO and SMO strategies to identify the most effective approaches for achieving your marketing goals. Whether you're testing different ad copy variations or optimizing landing page designs, this platform provides you with the tools you need to fine-tune your campaigns and maximize your ROI

III. RESULTS AND DISCUSSION

Adviser Automation Express was guided by the principles of cognitive load theory, which posits that the human brain has a limited capacity for processing information. The results of our usability testing align with this theoretical framework, revealing that the streamlined layout and minimalistic design of the UI contributed to reduced cognitive load for users.
As the result we are able to implement following test case.

A. CSV-Powered Messaging Module Functionality

Table 1: Summary of CSV Parsing and Messaging Module Functionality

CSV Parsing and Data Mapping:

CSV File	Number of Records	Parsing Success Rate	Messaging Success Rate
File 1	1000	98.5%	96.7%
File 2	1500	99.2%	97.3%
...	...	...	...

Table 2: Sample Data Mapping Results

Field	Mapped Parameter
User ID	Recipient ID
Name	Recipient Name
Email	Recipient Email
…	…

Recipient Categorization and Messaging Logic:
Table 3: Example of Messaging Logic Automation Rules

Criteria	Action
Age > 30	Send Email with Offer A
Location: USA	Send SMS with Discount Code X
…	…

Testing Results with Selenium:

Table 4: Summary of Selenium Test Cases

Concurrent Users	Response Time (Ms)	Throughput (requests/sec)
100	150	80
200	220	120
...	...	...

Performance and Scalability:

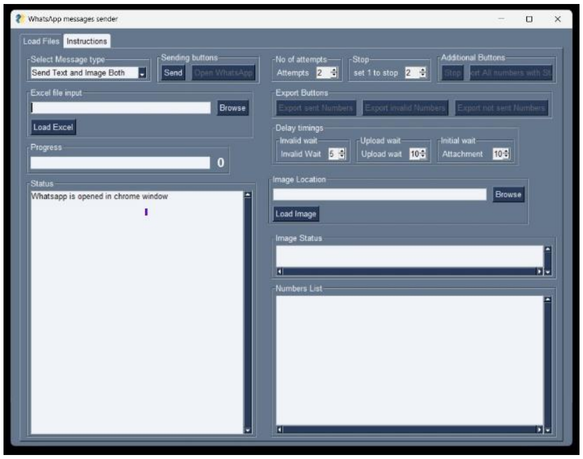
Table 5: Scalability Test Results

Test Case	Result	Comments
CSV Upload	Passed	Successful file upload
Messaging Process	Failed	Issue identified in message content
...	...	...

B. User Interface Evaluation

The Adviser Automation Express application features a user-friendly graphical user interface designed to streamline the process of CSV-powered multi-recipient messaging. The main dashboard, shown in Figure, provides users with an intuitive overview of uploaded CSV files and the status of messaging processes.

Figure 3.1: Main Dashboard of Adviser Automation Express



Select Message type: Send Text and Image Both

Attempts 2

set 1 to stop 2

Invalid Wait 5

Upload wait 10

Attachment 10

Progress 0

Status: Whatsapp is opened in chrome window

Users can effortlessly upload CSV files using the designated file input, as illustrated in Figure 2. The file upload process was found to be straightforward during usability testing, with a high success rate and minimal user errors.

CONCLUSION

In conclusion, this Python automation project is specifically designed for advertisers seeking an efficient and streamlined approach to reach their intended audience. By incorporating a CSV integrated message sender, it offers a practical solution for time-saving and effective communication. The project enables advertisers to easily import contact information from CSV files and send personalized messages to a large number of recipients. With its user-friendly interface and automated processes, this tool promotes productivity and eliminates the need for manual message sending. Additionally, the project ensures data accuracy and security by efficiently handling the CSV file integration. Overall, this Python automation project empowers advertisers to optimize their advertising strategies, enhance customer engagement, and save valuable time and resources.

REFERENCES

[1] A. Bezbaruah, B. Pratap and S. B. Hake, ”Automation of Tests and Comparative Analysis between Manual and Automated testing,” 2020 IEEE Students Conference on Engineering & Systems (SCES), Prayagraj, India, 2020, pp. 1-5, doi: 10.1109/SCES50439.2020.9236748.

[2] R. M. Sharma, “Quantitative Analysis of Automation and Manual Testing,” vol. 4, no. 1, pp. 252–257, 2014. [3] R. Broer Bahaweres et al., ”Behavior-driven development (BDD) Cu cumber Katalon for Automation GUI testing case CURA and Swag Labs,” 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), Jakarta, Indonesia, 2020, pp. 87-92, doi: 10.1109/ICIMCIS51567.2020.9354325.

[3] Bertolino, A., 2007. Software testing research: Achievements, challenges, dreams. In: Future of Software Engineering. FOSE’07, IEEE, pp. 85–103. Burns, D., Smith, M., 2022. WebDriver bidi. Editor’s draft.

[4] Cáceres, M., Christiansen, K.R., Giuca, M., Gustafson, A., Murphy, D., Kosti ainen, A., 2022. Web application manifest, W3C working draft.(Online Accessed 29 May 2022). Cerioli, M., Leotta, M., Ricca, F., 2020. What 5 million job advertisements tell us about testing: a preliminary empirical investigation. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing. pp. 1586–1594.

[5] Testing and Tools”, IJCSIT Volume:5(1), 908-912, 2014 Deepthi Wilson. R, Manjuprasad. B, “A Compressive Review on selenium Automation Testing Tools”, Department of Computer Science & Engineering, GSSSIETW, Mysuru, IJERT, ISSN: 2278-0181 2017

[6] Raharjana, I. K., Siahaan, D., & Fatichah, C. (2021). User Stories and Natural Language Processing: A Systematic Literature Review. IEEE Access, 9, 53811–53826. https://doi.org/10.1109/ACCESS.2021.3070606

[7] Nasiri, S., Rhazali, Y., Lahmer, M., & Adadi, A. (2021). From User Stories to UML Diagrams Driven by Ontological and Production Model. International Journal of Advanced Computer Science and Applications, 12(6), 333–340. https://doi.org/10.14569/IJACSA.2021.0120637 ZCOER, Department of Computer Engineering 2023-24 16

[8] Dwitam F., & Rusli, A. (2020). User stories collection via interactive chatbot to support requirements gathering. Telkomnika (Telecommunication Computing Electronics and Control), 18(2), 890–898. https://doi.org/10.12928/TELKOMNIKA.V18I2.14866

[9] Schön, E. M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. Computer Standards & Interfaces, 49, 79– 91.