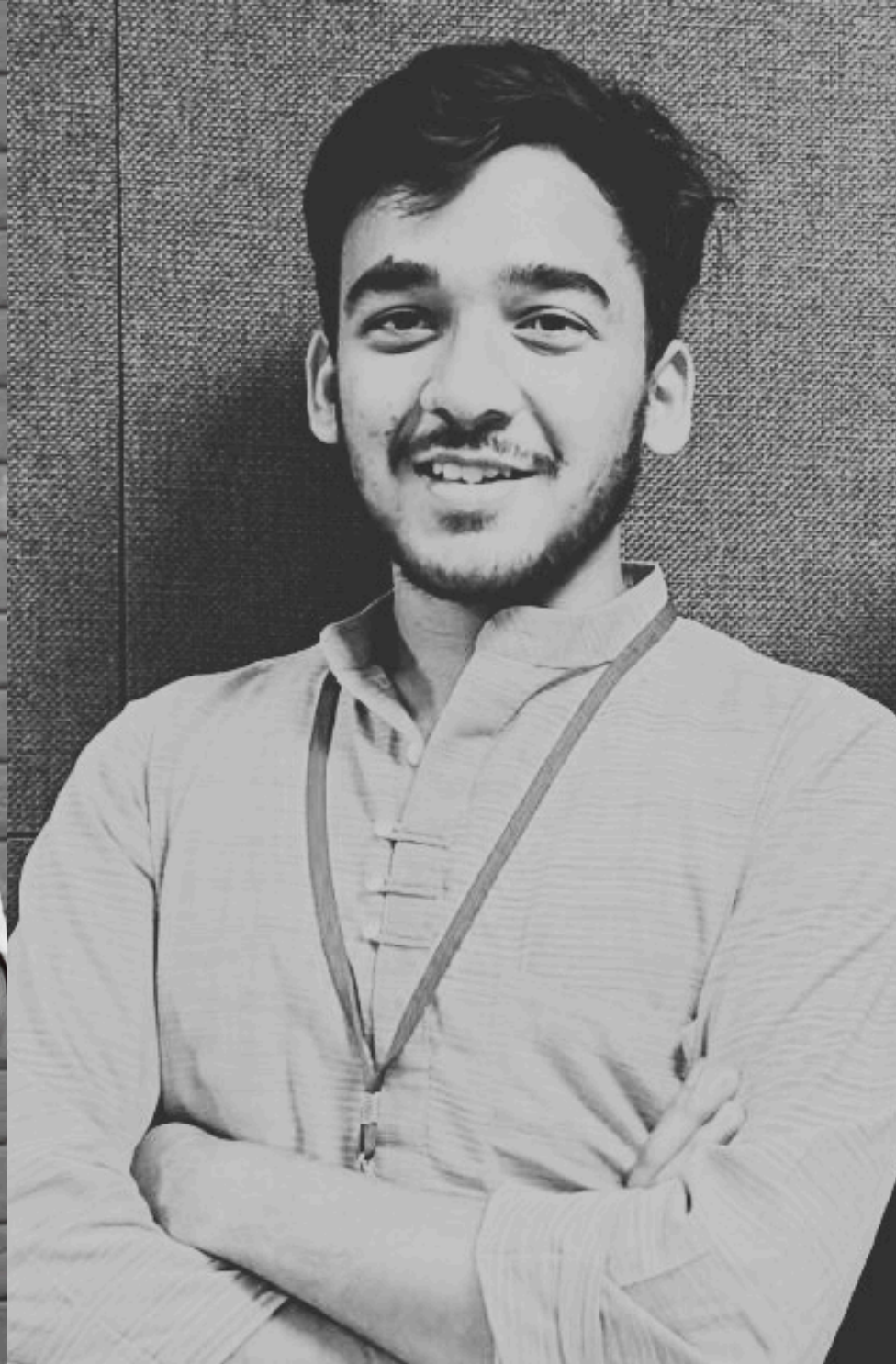


Team Helium

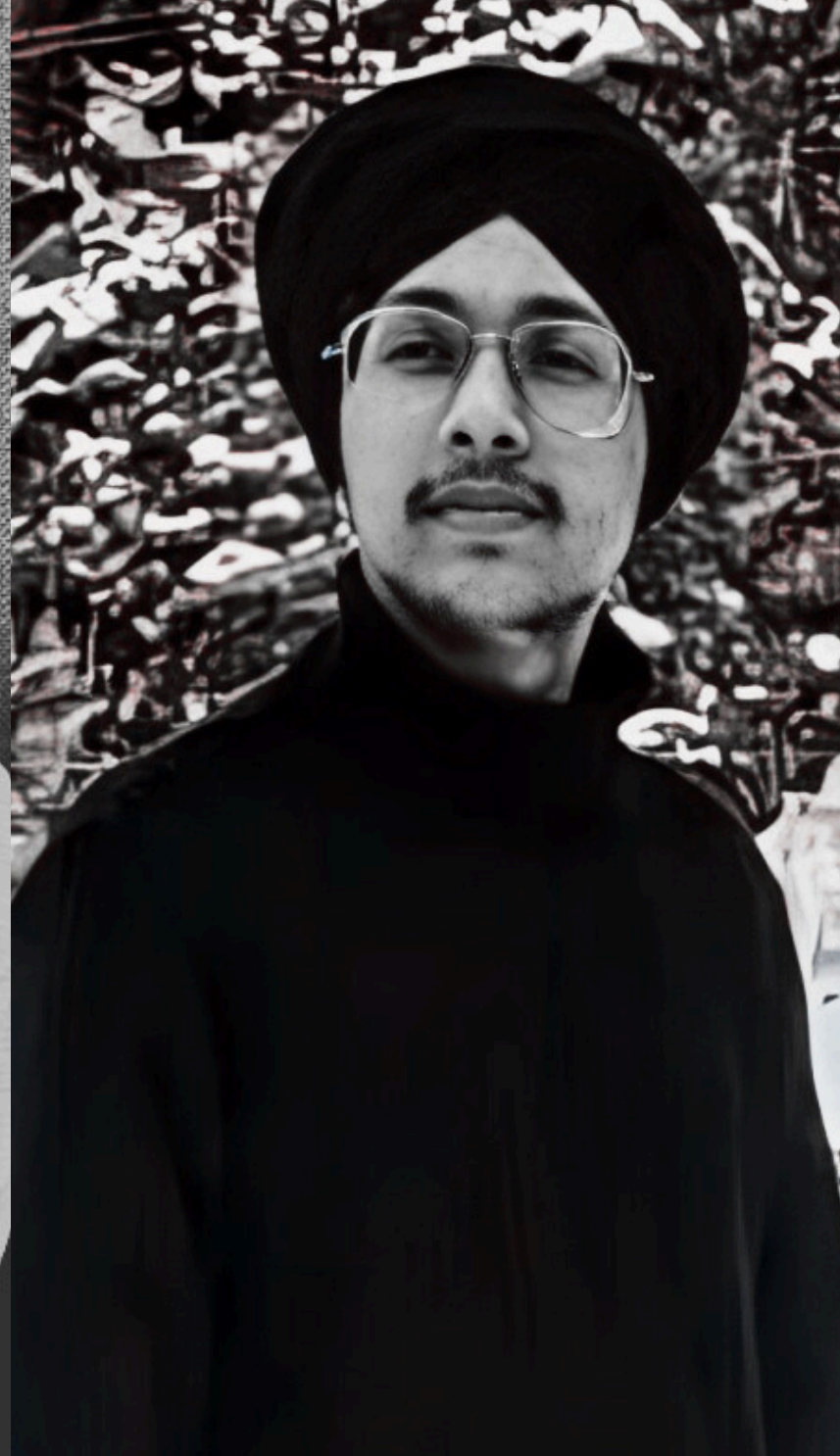
Progress Report Automated Resume Screening and Microsoft Form Submission with RPA



//aadityavij



//parasbisht



//devkaransingh



//raunakmalhotra

The Team

Objectives

2

He

Helium
4.0026

1

Automate Resume Screening: Extract candidate details from emails with PDF attachments titled "STGi | New Hire(s)", reducing manual effort and improving accuracy.

2

Populate Microsoft Form: Automatically fill a Microsoft Form with extracted data using RPA tools and dynamic selectors for varied form fields.

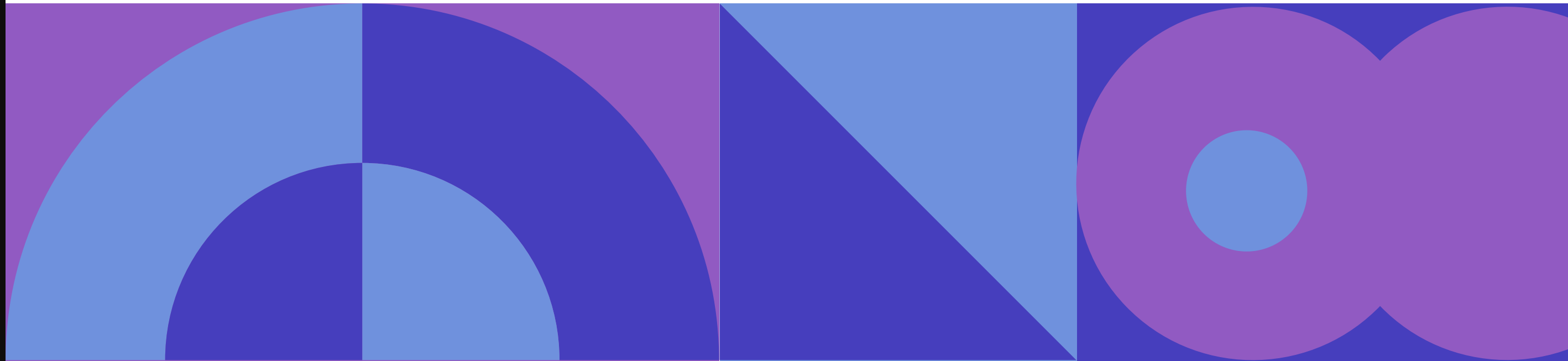
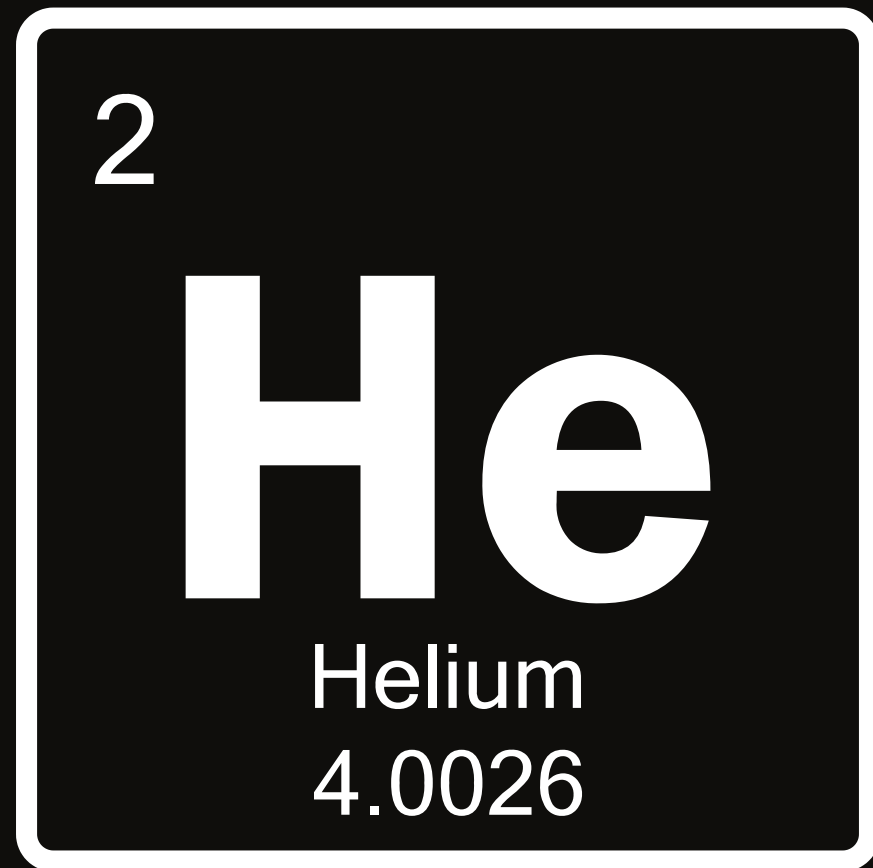
3

Generate Daily Audit Reports: Create and email a daily audit report of all processed forms, including robust error handling and detailed logging for troubleshooting and auditing.



SECTION 1:

The Approach and Workflow



Workflow #1

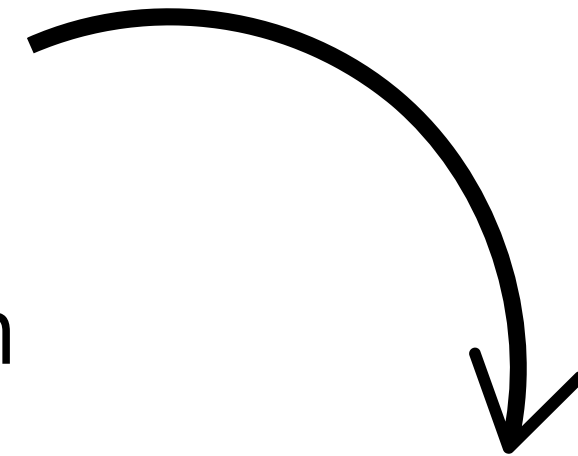
Get IMAP Mail Message

Helps to fetch
messages on gmail



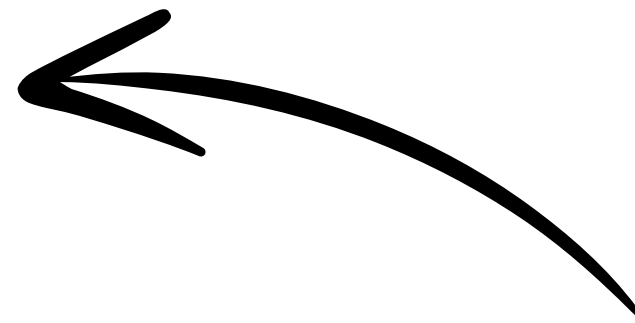
FOR EACH function

Filters out mails from
inbox as per the
selection criteria given



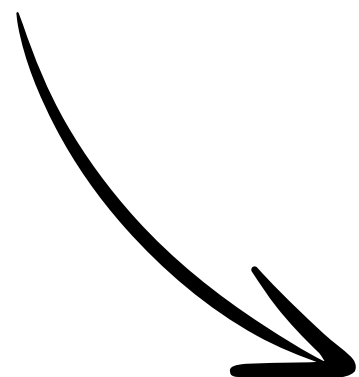
SAVE ATTACHMENT

Resumes from the selected
emails are transferred to
the allocated folder



VALIDATE WORKFLOW

To eliminate any warnings



RUN THE WORKFLOW

check if respective resumes have been saved in the allocated folder

Workflow #2

Initialize folder path and build a DataTable for storing extracted data.

Loop through each file (resume) and check the file type (PDF)

Extract the text content from each resume.

Use regex to extract fields (Name, Email, Phone, etc.) from the text.

Add the extracted data to a DataTable and write it to an Excel file.



Workflow#3

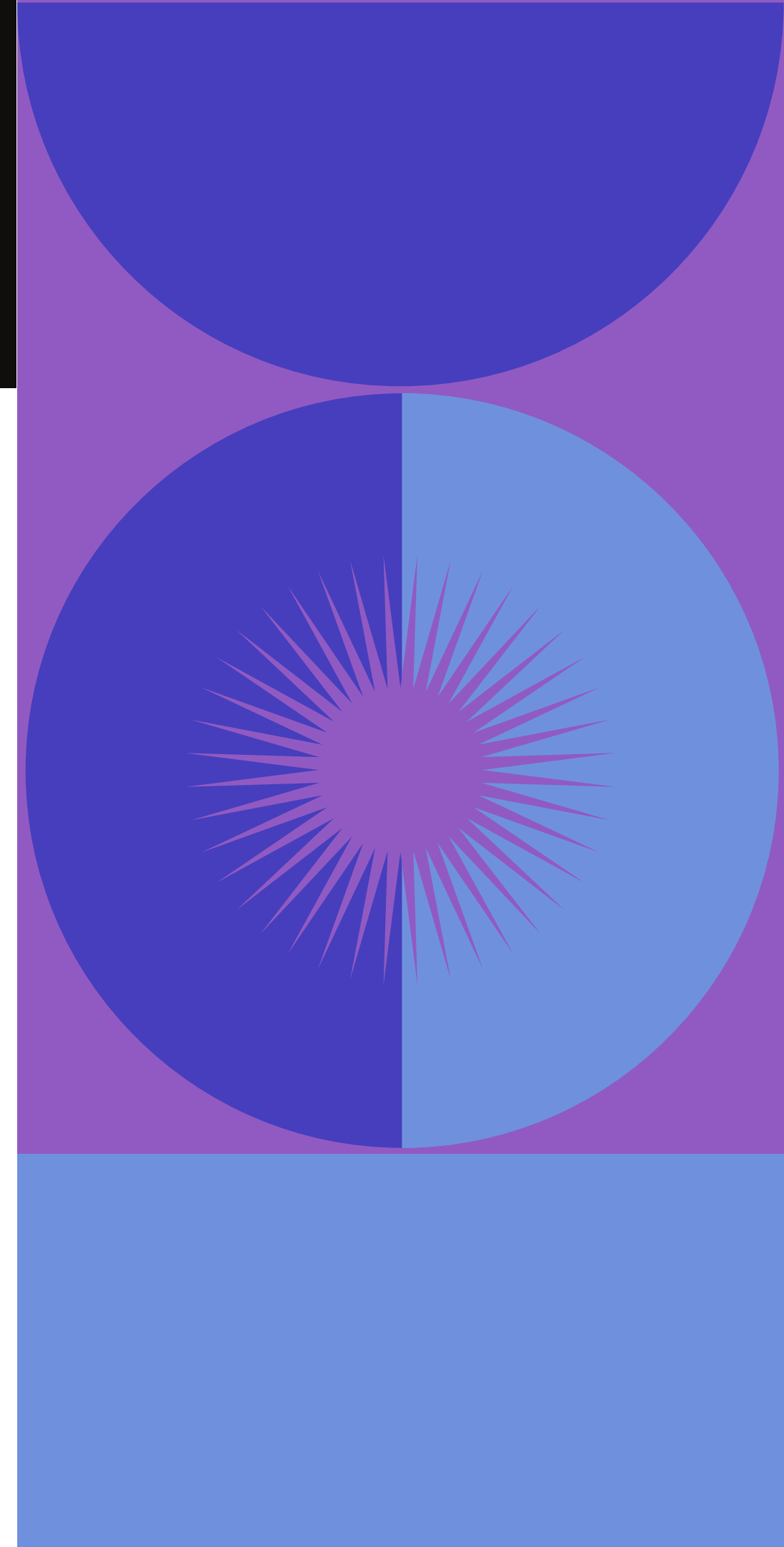
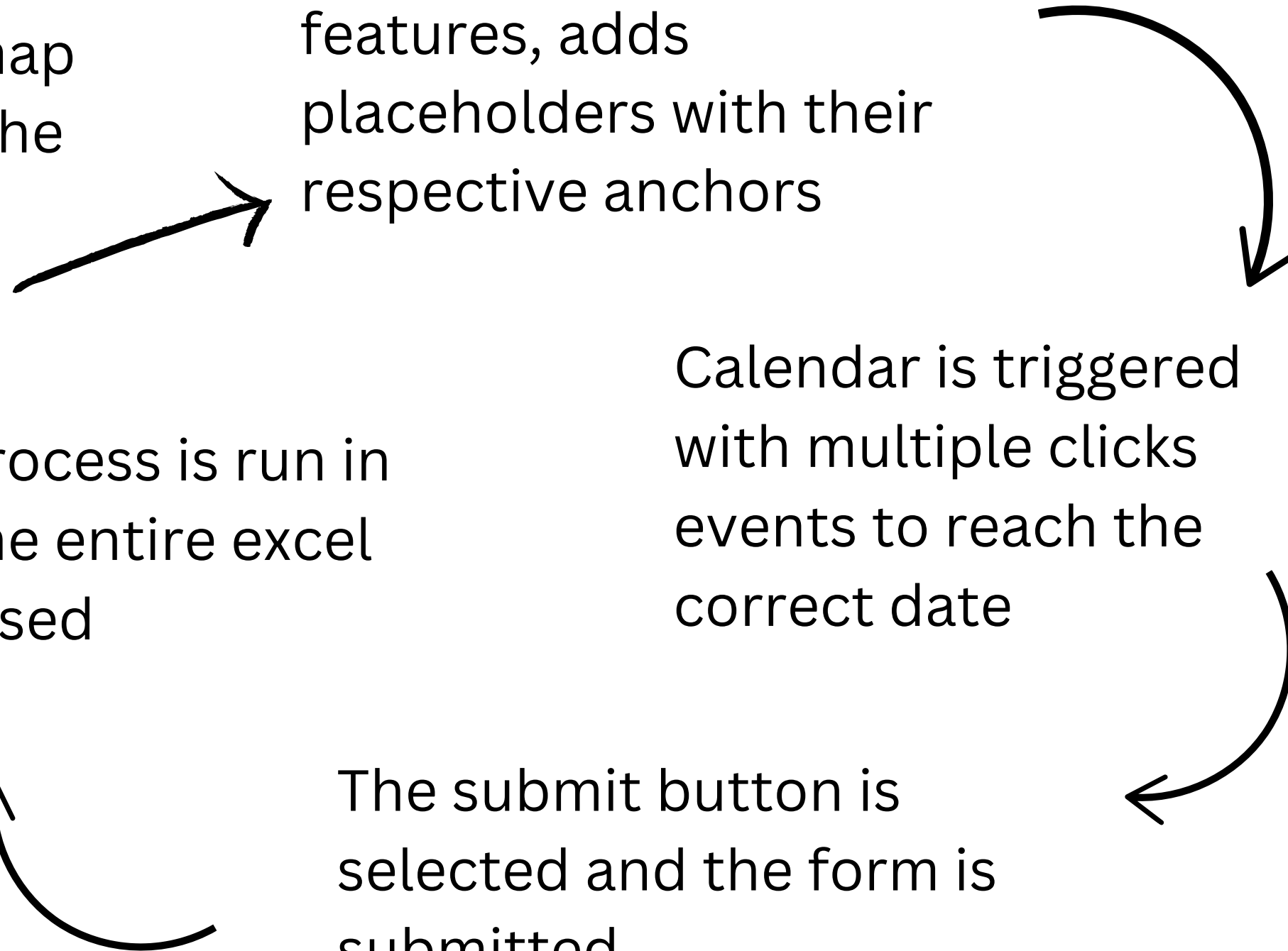
Loop through each row in the excel file and map each placeholder to the form using target - anchor pairs

App/Web recorder features, adds placeholders with their respective anchors

The same process is run in loop until the entire excel file is traversed

Calendar is triggered with multiple clicks events to reach the correct date

The submit button is selected and the form is submitted



24hrs

What went

We ended up wasting (utilising?) too much time to extract data from the pdf file into an **excel sheet** when it wasn't that necessary

wrong

We wasted too much time using classical **anchor methods** when we could have saved time by using **TYPE INTO** and **APP Recorder**