Developer Recruitment Tasks - Full Solutions Report

Task 1: Build an Automation in Make.com

Scenario:
You need to create random contacts in Airtable, generate company summaries with ChatGPT, store the reports in Google Docs, and send them to Slack daily at 7:00 AM.

Implementation Steps:
1. Create an Airtable base with columns: Name, Email, Company.
2. In Make.com, add a Scheduler module set to run daily at 7:00 AM.
3. Add an Airtable module → 'Create Record' → Use Random User API to generate random names and emails, and assign companies.
4. Add OpenAI module → Send prompt like: 'Summarize the company "{Company Name}" in 5 sentences.'
5. Add Google Docs module → Create a document with the format:
Company Name: {Company Name}
Summary: {Generated Text}
6. Add Slack module → Send message with Google Doc link.

Outcome: A daily automation that creates random contacts, summarizes company information, generates a Google Doc, and sends it to Slack automatically.

Task 2: Build a YouTube Scraper

Scenario:
You need to extract YouTube video details (Title, Channel, Description, Publish Date, View Count) and store them in Google Sheets.

Implementation Options:
1. Use YouTube Data API v3.
2. Or use Python with libraries like googleapiclient, pandas, and gspread.

Python Example Code:

```
from googleapiclient.discovery import build
import pandas as pd

# API setup
api_key = "YOUR_API_KEY"
youtube = build('youtube', 'v3', developerKey=api_key)

# Search for videos
request = youtube.search().list(
    q="technology",
    part="snippet",
    maxResults=5
)
response = request.execute()

videos = []
for item in response['items']:
    if item['id']['kind'] == 'youtube#video':
        video_id = item['id']['videoId']
        title = item['snippet']['title']
        channel = item['snippet']['channelTitle']
        description = item['snippet']['description']
        publish_date = item['snippet']['publishedAt']
        stats = youtube.videos().list(part="statistics", id=video_id).execute()
        views = stats['items'][0]['statistics']['viewCount']

        videos.append([title, channel, description, publish_date, views])

# Save to DataFrame
df = pd.DataFrame(videos, columns=["Title", "Channel", "Description", "Publish Date", "Views"])
print(df)

# Next: Push df to Google Sheets using gspread or Make.com Google Sheets module
```

Task 3: Build a Slack Bot and Connect to Make.com

Scenario:
You need a Slack bot that can be triggered in Make.com to send weekly reports into a private channel.

Implementation Steps:
1. Go to https://api.slack.com/apps and create a new Slack App.
2. Add a Bot User and assign permissions like 'chat:write'.
3. Install the bot to your workspace and copy the Bot Token.
4. In Make.com, add the Slack module and authenticate with the bot token.
5. Add a data source (e.g., Airtable, CRM, Google Sheets) to fetch weekly metrics.
6. Format the report:
Weekly Report:
Leads Contacted: 2345
Replied: 56
Meetings Booked: 4
Positive Replies: 12
7. Schedule the scenario weekly (e.g., every Monday at 9 AM).

Outcome: Every week, your Slack bot posts an automated weekly report into a private Slack channel.

Example Weekly Report Table:

Leads Contacted	Replied	Meetings Booked	Positive Replies
2345	56	4	12