



# StoryPoints AI Internship Program

---

## Week 1 Assignment — Data Extraction & Basic Transformation

### Useful Links:

Before you begin your tasks, take some time to go through the following resources. These will give you help you connect with mentors and peers.

### Mentor's Profiles

Success Coach Nilesh's LinkedIn: [Link](#)

Arun Bonam's LinkedIn: [Link](#)

### Community

WhatsApp Group: [Link](#) → Stay updated, ask questions, and share learning. We're currently sharing 101 Hacks to help you grow in Life, Career, Leadership, and Business. These will also support your journey as a professional tester.

### Share Your First Steps:

Before diving into the Week 1 tasks, we'd love for you to share this new beginning with your network. Create a short LinkedIn post announcing that you've started your Internship with London Success Academy.

This not only celebrates your milestone but also starts building your professional brand right from day one. 🚀

# Welcome to the StoryPoints AI Data Engineering Internship!

In Week 1, you will learn how to fetch data from multiple sources, perform basic transformations, and store it in a production-ready format on Google Cloud Storage (GCS). This assignment mimics a real-world scenario where data engineers extract, transform, and manage large datasets.

## ♦ Prerequisites

Set up Google Cloud using the following resources:

Create a Free GCP Account: <https://cloud.google.com/free>

Set Up a Project in GCP:

<https://cloud.google.com/resource-manager/docs/creating-managing-projects>

Create a GCS Bucket: <https://cloud.google.com/storage/docs/creating-buckets>

Install Google SDK CLI: <https://cloud.google.com/sdk/docs/install>

BigQuery Introduction: <https://cloud.google.com/bigquery/docs/introduction>

## Scenario

StoryPoints AI processes large-scale clickstream, transaction, and currency conversion data daily. As a data engineer, your Week 1 task is to extract data from multiple sources (CSV files and APIs), apply basic transformations, and store the cleaned datasets into a structured, partitioned GCS bucket.

## Datasets & API

Dataset	Format	Rows	Description
clickstream.csv	CSV	~200k	Website clickstream data
transactions.csv	CSV	~100k	Customer purchase transactions
ExchangeRate-API	JSON (API)	Live	Fetch real-time currency conversion rates

Download files from the GitHub repository:

<https://github.com/storypointsai/dataengineering-internship>

Use the free ExchangeRate-API to fetch real-time USD-based conversion rates. You need to register for an API key.

## Using the ExchangeRate-API

Step 1: Register for a Free API Key: <https://www.exchangerate-api.com/>

Step 2: API Endpoint

[https://v6.exchangerate-api.com/v6/YOUR\\_API\\_KEY/latest/USD](https://v6.exchangerate-api.com/v6/YOUR_API_KEY/latest/USD)

Step 3: Example Python Code

```
import os
import requests

API_KEY = "YOUR_API_KEY" # Replace with your key
url = f"https://v6.exchangerate-api.com/v6/{API_KEY}/latest/USD"

response = requests.get(url)
data = response.json()

if response.status_code == 200 and data["result"] == "success":
    rates = data["conversion_rates"]
    print("USD to INR:", rates["INR"])
else:
    print("API Error:", data)
```

Step 4: Sample JSON Response

```
{
  "result": "success",
  "base_code": "USD",
  "time_last_update_utc": "Fri, 05 Sep 2025 00:00:01 +0000",
  "conversion_rates": {
    "EUR": 0.92,
    "INR": 83.1,
    "GBP": 0.78,
    "JPY": 145.3
  }
}
```

⚡ Store raw JSON responses into: data/raw/api\_currency/YYYY-MM-DD/

## Assignment Tasks

1. Task 1: Explore datasets and document schemas, null values, and duplicates in README.md.
2. Task 2: Extract data — read CSVs in chunks (e.g., 50,000 rows) and fetch currency rates via API.
3. Task 3: Basic transforms — standardize column names, convert timestamps to UTC, deduplicate, and enrich transactions with amount\_in\_usd using conversion rates.
4. Task 4: Load — write cleaned outputs to GCS paths partitioned by ingest\_date=YYYY-MM-DD/ for each dataset.
5. Task 5: Logging & alerts — log record counts and failures; print warnings if inputs are missing or API fails.
6. Task 6: Architecture diagram — draw your pipeline (sources → ETL scripts → partitioned GCS) and include it in README.

## Deliverables

- Python scripts for ingestion, transformation, and GCS loading.
- README.md with dataset understanding, approach, assumptions, and architecture diagram.
- Partitioned outputs stored in GCS.
- Proper Documentation with the results of the tasks that you achieved (ex: screenshots)
- 

## Submission Workflow

- Fork the starter GitHub repository (link will be provided).
- Create a PRIVATE repository in your GitHub and push your code there.
- Add StoryPoints AI reviewers as collaborators. ( arun.bonam@storypointsai.com)
- Work in a feature branch (e.g., week1-task-<name>) and commit frequently.
- Create a Pull Request to your main branch and assign reviewers.
- Submit your repository link.

## Pro Tips

- Use pandas chunking: pd.read\_csv(file, chunksize=50000) for large CSVs.
- Implement retry with exponential backoff for API calls.
- Partition outputs by ingest\_date=YYYY-MM-DD/ for efficiency.
- Use UTC timestamps to avoid timezone issues.
- Keep commits small and meaningful.