



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Automated Data Pipeline: Google Forms to MySQL Integration

The domain of the Project:
Power BI and SQL

Under the guidance of
Ms. Siddhika (Software Engineer)

By:
Ms. Anjali A S (B.Tech CSE Graduate)

Period of the project
May 2024 to August 2025



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Declaration

The project titled “**Automated Data Pipeline: Google Forms to MySQL Integration**” has been mentored by Ms. Siddhika, organised by SURE Trust, from May 2025 to August 2025, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

Name

Ms. Anjali A S

Mentor

Software Engineer—HCL Tech

Prof. Radhakumari

Executive Director & Founder

SURE Trust



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Table of contents

1. Executive summary
2. Introduction
3. Project Objectives
4. Methodology & Results
5. Social / Industry relevance of the project
6. Learning & Reflection
7. Future Scope & Conclusion



Executive Summary

The project, **“Google Form to MySQL Integration”**, demonstrates an automated workflow for collecting and managing data submitted via Google Forms. It enables seamless transfer of form responses from Google Sheets to a structured MySQL database, ensuring accurate storage and easy retrieval. Using Python with the gspread library and Google Cloud service account credentials, the system fetches data, cleans headers, converts timestamps to MySQL-compatible format, and inserts only new entries while avoiding duplicates. This approach eliminates manual data entry, reduces errors, and provides a centralized database for analysis and reporting. The solution is scalable, efficient, and can be scheduled for periodic execution to maintain real-time synchronization of new form submissions.

The project successfully demonstrates the ability to automate the transfer of Google Form responses into a MySQL database with accuracy and efficiency. All form submissions are captured in real-time and stored in a structured format, with timestamps correctly converted for database compatibility. The system effectively avoids duplicate entries by checking existing records, ensuring data integrity. Additionally, the Python-based solution proved scalable, handling multiple responses without errors, and provides a reliable foundation for future enhancements such as automated scheduling or integration with other reporting tools. Overall, the project confirms that Google Forms can be seamlessly integrated with MySQL to streamline data collection and management processes.



Introduction

In today's digital era, organizations and institutions frequently rely on online forms to collect data efficiently from participants, employees, or students. Google Forms is a popular tool for this purpose due to its simplicity, accessibility, and real-time data collection capabilities. However, managing and analyzing large volumes of responses manually in Google Sheets can be cumbersome, error-prone, and inefficient. This project, **"Google Form to MySQL Integration"**, addresses this challenge by automating the process of transferring form responses into a structured MySQL database. The primary goal is to ensure accurate, reliable, and centralized data storage while eliminating manual intervention. The scope of the project includes fetching data from Google Sheets, cleaning and formatting it, converting timestamps to MySQL-compatible format, and inserting only new entries to maintain data integrity. The innovative component of the project lies in its automated synchronization mechanism, which can be scheduled to update the database periodically, allowing near real-time data management. This solution not only streamlines the workflow but also enables organizations to efficiently analyze and report data, providing a scalable framework for future extensions such as real-time dashboards or advanced analytics.

This project is particularly relevant in scenarios where timely and accurate data collection is critical, such as tracking attendance, gathering feedback, or conducting surveys. Manual methods of transferring data from Google Sheets to a relational database are not only labor-intensive but also prone to errors that can compromise the quality of the data. By integrating Google Forms with MySQL using Python, the project provides a seamless and automated approach to manage data efficiently. The system ensures consistency, reduces redundancy, and offers a centralized repository that can be accessed for reporting, analysis, or decision-making. Furthermore, the project demonstrates an innovative use of cloud-based credentials and Python automation to bridge the gap between online forms and structured databases, laying the foundation for more advanced data-driven applications in the future.

Overall, this project showcases how automation can simplify data management by connecting cloud-based tools like Google Forms with traditional databases such as MySQL. It highlights the importance of accurate data handling, efficiency, and scalability, providing a practical solution for organizations that rely on frequent data collection and analysis.



Project Objectives

- **Objectives and Goals:**

The main objective of this project is to automate the process of transferring Google Form responses into a MySQL database to improve efficiency, accuracy, and data integrity. The detailed goals include:

1. **Establish Secure Connection with Google Sheets:**

- Utilize a Google Cloud service account and JSON credentials to authorize Python scripts.
- Ensure secure and authorized access to the Google Sheet containing form responses.

2. **Fetch and Clean Data from Google Sheets:**

- Retrieve all form responses using the gspread Python library.
- Clean and standardize column headers by removing extra spaces and converting them to lowercase for consistency.

3. **Insert Data into MySQL Database:**

- Establish a connection to the MySQL database (meeting_feedback) and insert data into the table form_responses.
- Implement a mechanism to check existing records (using unique identifiers such as email and timestamp) to avoid duplicate entries.

4. **Timestamp Conversion for Database Compatibility:**

- Convert Google Sheet timestamps from DD/MM/YYYY HH:MM:SS format to MySQL YYYY-MM-DD HH:MM:SS format.
- Ensure all timestamps are consistent and compatible with MySQL's DATETIME data type for accurate reporting and analysis.

- **Expected Outcomes and Deliverables:**

The project is expected to provide a reliable and efficient automated system that delivers the following outcomes:

1. **Accurate and Complete Data Transfer:**

- All new Google Form responses are inserted into MySQL accurately.
- Rows with missing critical information (e.g., email) are skipped with proper logging.

2. **Centralized Database for Analysis:**

- A structured MySQL database containing all responses, enabling easy querying, filtering, and reporting.
- Eliminates the need for manual data entry from Google Sheets to the database.



Methodology and Results

1. Methods / Technology Used

The project leverages a combination of Python programming and database management technologies to automate the transfer of Google Form responses into MySQL. The key methods include:

- Python Automation: Used to write scripts for fetching data from Google Sheets, cleaning and formatting it, and inserting it into MySQL.
- Google Sheets API (gsread): Used to read responses from Google Forms stored in Google Sheets.
- Data Cleaning and Validation: Headers are standardized, missing fields are handled, and timestamps are converted to MySQL-compatible DATETIME format.
- MySQL Database Management: Data is inserted into a structured relational database with mechanisms to prevent duplication using unique identifiers (email + timestamp).

2. Tools / Software Used

- Python 3.13: Main programming language for automation and data processing.
- MySQL Workbench: For database creation, management, and querying.
- Google Sheets & Google Forms: To collect form responses and store them online.
- Google Cloud Console: To create a service account and generate JSON credentials for secure API access.
- Libraries:
 - gsread for Google Sheets interaction
 - oauth2client for Google API authentication
 - mysql-connector-python for MySQL database operations

3. Data Collection Approach

- Source: Google Form responses submitted by users.
- Storage: Responses are automatically stored in a linked Google Sheet.
- Access: Python script accesses the Google Sheet using API credentials.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

- Validation: Each row is checked for critical fields like email and timestamp before insertion.
- Insertion Logic: Only new responses (not already in the MySQL table) are inserted to avoid duplicates.

4. Project Architecture

The project follows a simple yet efficient architecture:

1. User Submission: Users submit responses via Google Form.
2. Data Storage: Responses are saved in a Google Sheet automatically.
3. Python Script Execution:
 - Connects to Google Sheet via API.
 - Fetches all rows and cleans data.
 - Converts timestamps into MySQL format.
 - Checks existing rows in MySQL to avoid duplicates.
 - Inserts new rows into the database.
4. Database Storage: Data is stored in the meeting_feedback MySQL database in the form_responses table.
5. Result Access: Stored data can be queried, analyzed, or exported for reporting.

5. Results

The system successfully automated the insertion of Google Form responses into MySQL.

New submissions were correctly identified and inserted without duplicating existing data.

Timestamp conversion was successful for all entries.

The database became a centralized repository for analysis and reporting.



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

- Google form responses are updated in the google sheets

Attendance form (Responses)						
File Edit View Insert Format Data Tools Extensions Help						
100% 123 Roboto 10 B I A						
A1 Timestamp						
	A	B	C	D	E	F
1	Timestamp	email	meeting_id	rating	feedback	
2	31/08/2025 19:13:05	gopikang2020@gmail.cr	6859	4	Good	
3	31/08/2025 19:42:37	anjalias2004@gmail.cor	5346346	5	good	
4	31/08/2025 21:13:26	Aswini as	Aswini	5	Nice	
5	01/09/2025 00:03:52	nipunonepiece@gmail.c	Monkeydluffy	5	It was great class	
6	01/09/2025 10:35:00	anjalisuresh5237@gmm	56578	5	good	
7	01/09/2025 17:07:08	Rs45@gmail.com	Rohit sharma	5	It was a clearly understa	
8	01/09/2025 17:13:06	dominic@gmail.com	ghw567g	5	nice	
9	01/09/2025 20:32:35	maya123@gmail.com	ugh356i	3	Less content	
10	01/09/2025 20:33:09	anu2009@gmail.com	hhsegyuw76	3	Satisfied	
11	01/09/2025 20:34:38	karthik308@gamil.com	bgdsgy889	2	Not bad	
12	01/09/2025 20:35:35	tim2020@gmail.com	abc123	3	Not bad	
13	01/09/2025 20:36:03	manoj123@gmail.com	abc123	4	Nice	
14	01/09/2025 20:36:44	aswinias2005@gmail.cc	abc123	5	Nice	
15	01/09/2025 20:38:46	syam2001@gmail.com	abc123	5	Loved it	
16						

- Google sheets are connected to the MySQL using Google Sheets API integration (through a Google Cloud Service Account)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\anjali\sqlproject> python form_to_mysql.py
Connected to MySQL
14 rows inserted/updated successfully.
MySQL connection closed
PS C:\Users\anjali\sqlproject> |
```



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

- Finally the google from responses are stored in the MySQL .

```
1 • CREATE DATABASE IF NOT EXISTS meeting_feedback;
2 • USE meeting_feedback;
3
4 • CREATE TABLE IF NOT EXISTS form_responses (
5     id INT AUTO_INCREMENT PRIMARY KEY,
6     email VARCHAR(255) NOT NULL,
7     meeting_id VARCHAR(100) NOT NULL,
8     rating TINYINT NULL,
9     feedback TEXT NULL,
10    submitted_at DATETIME NOT NULL,
11    is_processed TINYINT DEFAULT 0,
12    UNIQUE KEY uq_response (email, meeting_id, submitted_at)
13 );
14
15 • select * from form_responses ;
```

id	email	meeting_id	rating	feedback	submitted_at	is_processed
1	gopikang2020@gmail.com	6859	4	Good	2025-08-31 19:13:05	0
2	anjalias2004@gmail.com	5346346	5	good	2025-08-31 19:42:37	0
3	Aswini as	Aswini	5	Nice	2025-08-31 21:13:26	0
7	nipunonepiece@gmail.com	Monkeyduffy	5	It was great class	2025-09-01 00:03:52	0
16	anjalisuresh5237@gmail.com	56578	5	good	2025-09-01 10:35:00	0
22	Rs45@gmail.com	Rohit sharma	5	It was a clearly understanding session	2025-09-01 17:07:08	0
29	dominic@gmail.com	ghw567g	5	nice	2025-09-01 17:13:06	0
37	maya123@gmail.com	ugh356i	3	Less content	2025-09-01 20:32:35	0
38	anu2009@gmail.com	hhsegyuw76	3	Satisfied	2025-09-01 20:33:09	0
39	karthik308@gmail.com	hndenv889	2	Nnt had	2025-09-01 20:34:38	0

form_responses 6 x

Output

Action Output

#	Time	Action	Message
5	10:35:24	select * from form_responses LIMIT 0, 1000	4 row(s) returned



Learning and Reflection

This project provided me with valuable hands-on experience in bridging data between cloud-based platforms and relational databases. On the **technical side**, I gained proficiency in working with the **Google Sheets API** through a Google Cloud Service Account, learning how to authenticate securely, fetch real-time form responses, and integrate them seamlessly into a **MySQL database** using Python. I also deepened my understanding of database operations such as insertions, updates, and handling errors like missing values or incorrect data formats.

From an **analytical and problem-solving perspective**, I learned how to debug integration errors, identify discrepancies in field names, resolve datetime format mismatches, and ensure smooth data flow. Each challenge improved my ability to analyze the root cause of issues and apply systematic fixes, enhancing both my troubleshooting and logical thinking skills.

Additionally, I developed **soft management skills** such as planning the workflow, structuring the project into smaller stages (API setup, database connectivity, testing, and automation), and managing time effectively. Documenting the process, collaborating for feedback, and iteratively improving the code also helped me strengthen communication and project management abilities.

Overall, the project enriched my knowledge of **cloud integration, database management, and Python programming**, while simultaneously building confidence in solving real-world data engineering challenges.



Conclusion and Future Scope

The project successfully demonstrated how Google Forms can be integrated with MySQL to create a seamless data collection and storage pipeline. By leveraging the Google Sheets API through a Google Cloud Service Account and automating the data transfer with Python, form responses were securely and efficiently stored in a structured relational database. This integration not only simplified the data management process but also ensured reliability, scalability, and accuracy of the collected information.

In terms of future scope, the project can be further enhanced by introducing automation techniques such as scheduling the Python script with cron jobs (Linux) or Task Scheduler (Windows), or by deploying it as a cloud function for real-time updates. Additional features like data visualization dashboards (e.g., with Power BI, Tableau, or Python libraries like Matplotlib/Plotly) can be implemented to gain deeper insights from the collected responses. The system can also be extended to handle large-scale datasets, integrate with other cloud services (such as BigQuery), or incorporate machine learning models for predictive analysis.

Thus, this project not only meets its initial objectives but also lays the groundwork for future innovations in automated data pipelines and intelligent feedback management systems.