
Capstone Project - Student Data Manipulation

---

1. CRUD + Database Module (Basics)

Goal: Build a simple student database and perform CRUD operations. **Skills:** SQL, Python, FastAPI (optional)

Data Preparation — `students.csv`:

```
StudentID,Name,Age,Course
101,Neha,21,AI
102,Arjun,22,ML
103,Sophia,20,Data Science
104,Ravi,23,AI
105,Meena,21,ML
```

Exercise:

- Create a `students` table.
- Write Python code or SQL queries for:
    - Insert new students
    - Update a course
    - Delete a student
    - Fetch students enrolled in “AI”

---

2. REST API Module

Goal: Expose CRUD operations as a REST API. **Skills:** FastAPI, Pydantic, GET/POST/PUT/DELETE

Exercise:

- Create endpoints:
    - `GET /students` – fetch all
    - `POST /students` – add
    - `PUT /students/{id}` – update
    - `DELETE /students/{id}` – remove

Data: Use the same `students.csv` as above.

---

3. ETL & Pandas Processing Module

Goal: Read marks data, transform, and generate analytics. **Skills:** Pandas, Data Cleaning, Derived Columns

Data — `marks.csv`:

```
StudentID,Maths,Python,ML
101,78,85,90
102,65,70,60
103,88,92,95
```

```
104,50,60,45
105,80,88,84
```

Exercise:

- Read `marks.csv`
- Add `TotalMarks` and `Percentage`
- Add `Result` column: Pass if Percentage ≥ 50
- Save final CSV: `student_results.csv`

---

4. Queue Processing Module

Goal: Asynchronously process uploaded results. **Skills:** Python Queue or RabbitMQ

Data: Use the `marks.csv` from above.

Exercise:

- Producer script: pushes CSV path into a queue
- Consumer script: reads CSV, runs ETL, saves results
- Print logs for each step

---

5. Logging & Error Handling Module

Goal: Log every operation. **Skills:** Python `logging`, exception handling

Exercise:

- Log API calls (`GET /students`, etc.)
- Log ETL execution time and errors
- Log queue consumption status

---

6. Automation & Scheduling Module

Goal: Run ETL daily. **Skills:** `cron` (Linux/macOS) or Task Scheduler (Windows)

Exercise:

- Schedule the ETL script to run every morning at 8 AM
- Save a timestamped file (`daily_report_YYYYMMDD.csv`)

---

7. Advanced: Message Queues & Background Processing

Goal: Split producer & consumer logic for real-world systems. **Skills:** RabbitMQ, Async Processing

Exercise:

- Producer accepts an API upload of marks CSV
- Consumer processes it and updates database

---

8. Final Analytics Module

Goal: Merge students + marks → Generate insights **Skills:** Pandas Joins, Aggregations, GroupBy

Exercise:

- Join `students.csv` and processed `student_results.csv`
- Show top 3 students by percentage
- Calculate average marks per course
