AI Student Intelligence

Internal How-To Guide (Run From Scratch)

This guide explains **exactly** how to download, configure, and run the AI Student Intelligence system on your own machine, using **your own Google Sheets data**.

Read fully once before starting.

---

0. What This Project Is (In One Line)

A full academic intelligence system that:

- Validates raw exam data
- Generates analytics and explainable insights
- Uses AI (local or cloud) to generate readable summaries
- Shows everything in a clean UI

---

1. What You Need on Your System (MANDATORY)

1.1 Software Requirements

You **must** have the following installed:

1. **Python 3.10 or later**
    - Check:
    - python --version
    - If not installed: install from python.org
2. **Git**
    - Check:
    - git --version
    - If not installed: install Git for your OS

3. **Google Chrome / any modern browser**
    - Required for Streamlit UI

1.2 Optional but Recommended (LLM Options)

You have **two AI modes**:

Option A — Local AI (Recommended, Free)

- **Ollama**
- Runs fully on your machine
- No API key needed

Install Ollama:

- https://ollama.com
- After install, open terminal and run:
- ollama pull mistral

This is the **default mode**.

Option B — Cloud AI (Optional)

If you want live AI generation:

- OpenAI / Claude / Gemini / DeepSeek
- Requires API keys

You can skip this initially.

---

2. Clone the Project From GitHub

Open terminal / PowerShell and run:

git clone https://github.com/KoushikTalluri09/ai-student-intelligence.git

cd ai-student-intelligence

You should now be inside the project folder.

---

3. Create and Activate Virtual Environment

Windows:

python -m venv venv

venv\Scripts\activate

Mac / Linux:

python3 -m venv venv

source venv/bin/activate

You should see (venv) in your terminal.

---

4. Install Python Dependencies

Run:

pip install -r requirements.txt

If this fails:

- Upgrade pip:
- pip install --upgrade pip
- Then retry.

---

5. Google Sheets Setup (MOST IMPORTANT STEP)

5.1 Create a Google Sheet

Create a Google Sheet that will act as your **database**.

Name it exactly (or change config later):

AI_Student_Intelligence

5.2 Required Sheets and Columns

Sheet 1: raw_student_scores

This is where **your raw data lives**.

You MUST use these column names exactly:

student_id

grade

subject

exam_type

score

exam_date

Rules:

- score must be numeric
- exam_date must be a valid date
- exam_type can be anything (mock / real / quiz)

5.3 Create Google Service Account (ONE TIME)

1. Go to **Google Cloud Console**
2. Create a new project
3. Enable:
    - Google Sheets API
    - Google Drive API
4. Create **Service Account**
5. Generate **JSON key file**

Download the JSON file.

5.4 Share Google Sheet With Service Account

Open your Google Sheet → Share → Add:

service-account-name@project-id.iam.gserviceaccount.com

Give **Editor access**. This is mandatory.

---

6. Configure Environment Variables (.env)

In the project root, create a file called:

.env

Add:

GOOGLE_SHEETS_CREDENTIALS=path/to/service_account.json

GOOGLE_SHEETS_DB_NAME=AI_Student_Intelligence

Optional AI keys

OPENAI_API_KEY=...

ANTHROPIC_API_KEY=...

GEMINI_API_KEY=...

DEEPSEEK_API_KEY=...

LLM_PROVIDER=ollama

Notes:

- .env is NOT committed to GitHub
- Path can be absolute or relative

---

7. Run the Full Pipeline (DATA → INSIGHTS → AI)

This step generates **everything**.

Run:

python pipeline_runner.py ollama 20

What happens:

1. Phase 0 — Data validation
2. Phase 1 — Analytics

3. Phase 2 — Explainable insights
4. Phase 3 — AI summaries
5. Phase 4 — Student consolidation

After success, your Google Sheet will contain:

- validated_results
- subject_analytics
- subject_insights
- subject_summaries
- student_consolidated_latest
- student_consolidated_history

---

8. Start the Backend API

In the same terminal (or new one with venv active):

python -m uvicorn pipeline_server:app --host 0.0.0.0 --port 8000

If successful, you'll see:

Uvicorn running on http://0.0.0.0:8000

Do NOT close this terminal.

---

9. Start the UI (Streamlit)

Open a new terminal (activate venv again) and run:

streamlit run ui_app.py

Browser will open automatically.

---

10. Using the UI

1. Enter Student ID (example: S001)
2. Choose AI Engine:
    - Ollama → Cached (fast)
    - Others → Live
3. Click **Generate Academic Report**

What you'll see:

- Overall academic assessment
- Color-coded subject cards
- Trend indicators
- Subject-level insights
- Explainability (evidence)
- Recommended next steps

---

11. Using Your Own Data

To use **your own dataset**:

1. Replace data in raw_student_scores
2. Keep column names unchanged
3. Rerun:
4. python pipeline_runner.py ollama 20
5. Refresh UI

No code changes needed.

---

12. Common Errors & Fixes

Error: “storage module not found”

Run scripts using:

python -m <module>

or ensure project root is current directory.

Error: NaN / JSON errors

Already handled in pipeline.
If you see it, your sheet has invalid values.

UI shows empty data

Check:

- Backend is running
- Pipeline was run successfully
- Student ID exists in sheet

---

13. Important Rules (DO NOT BREAK)

- Do not rename columns
- Do not delete headers
- Do not edit generated sheets manually
- Always rerun pipeline after data change

---

14. Final Notes

This system is:

- Deterministic
- Explainable
- Auditable
- Production-style

It is **not** a toy demo.

If anything breaks, it’s almost always:

- Google Sheet permission issue
- Column name mismatch
- Backend not running