**🧾 Session Summary: HRIS Take-Home Project (Late Night Work Log)**

🕐 Time Block

**Started:** ~6:30PM  
**Wrapped:** ~11:45PM  
**Total Duration:** ~5 hours  
**Goal:** Jump-start and validate ingestion and transformation pipeline

✅ Work Completed Tonight

1. **Data Ingestion and Cleaning**

* Loaded all sheets from HRIS\_TAKE\_HOME\_PROJECT\_DATA.xlsx
* Normalized column headers across Employees, Applicants, EmploymentType
* Converted date columns using pd.to\_datetime() with defensive errors='coerce'
* Removed legacy Excel-origin conversion logic

2. **Deduplication + Audit Logging**

* Implemented duplicate filtering:
  + Employees → deduplicated by name, hire\_date, department
  + Applicants → deduplicated by name, role, application\_date
* Logged all dropped records to:
* logs/dropped\_employees.csv
* logs/dropped\_applicants.csv

3. **Database Creation**

* Created fresh SQLite DB: hris\_project.db
* Exported cleaned tables:
* employees
* applicants
* employment\_types

4. **SQL View Construction**

* Built two views:
  + time\_to\_hire: calculates days between application\_date and hire\_date
  + status\_summary: counts applicants by status
* Filtered invalid timelines (hire before application)

5. **Error Exclusion & Logging**

* Identified applicant records with invalid time\_to\_hire\_days
* Logged excluded rows to logs/invalid\_hires.csv with reason codes
* e.g., Steven Avila: “Hire date precedes application date”

6. **Unit Testing (pytest)**

* Validated structure and logic of SQL views:
  + test\_time\_to\_hire\_view\_structure ✅
  + test\_status\_summary\_view\_structure ✅
  + test\_time\_to\_hire\_null\_flags ✅
* All tests passed

7. **Checklist + Deliverable Mapping**

* Compared original project instructions to current state
* Calculated completion percentage (~60–65%)
* Identified remaining deliverables for Tuesday–Wednesday:
* API endpoints
* Cron automation
* Visualizations
* Documentation

🧠 Thoughtful Additions (Above and Beyond)

* Logged temporal inconsistencies with error reason for HR accountability
* Modularized logic with data\_loader.py and transform.py separation
* Strategically prepared for downstream processes with future-proof naming and schema