BACKEND WORKFLOW CHECKLIST & GUIDANCE

Objective:

Process user inputs (galaxy catalogs, JWST spectra, cosmology parameters) into scientifically meaningful outputs for frontend display and downloads, integrating multiple modules and producing visualizations, dashboards, and downloadable reports.

---

STEP 0: SETUP & PREPARATION (Shared)

[ ] Set up project structure with separate modules:

- input\_handler.py

- cos\_evo\_module.py

- cluster\_analyzer.py

- jwst\_spectrum.py

- dashboard\_integrator.py

- report\_generator.py

- utils.py

- main.py

[ ] Create virtual/conda environment with required packages:

- py21cmfast

- Bagpipes

- jwst

- numpy, scipy, pandas, matplotlib, plotly

- astropy, h5py

[ ] Prepare `requirements.txt` or `environment.yml`

[ ] Initialize Git repository and define branch/module ownership

[ ] Define standardized output formats (JSON/CSV for data, PNG/Plotly for plots)

[ ] Implement logging and error handling skeleton

---

STEP 1: INPUT HANDLING (Shared)

[ ] Accept and validate uploaded catalogs (CSV/FITS) → columns: RA, Dec, z, photometry

[ ] Accept and validate JWST spectra files (FITS)

[ ] Read and validate cosmology parameters (Ω\_m, H0, z-range)

[ ] Store validated files securely in structured folders:

- inputs/catalogs/

- inputs/jwst/

- outputs/cos-evo/

- outputs/cluster/

- outputs/jwst/

[ ] Return confirmation or error messages to frontend

---

STEP 2: MODULE-SPECIFIC TASKS

Teammate 1 — Cosmic Evolution & Cluster Analysis

[ ] Run 21 cm simulation with py21cmfast

[ ] Generate brightness temperature maps and power spectra

[ ] Plot RA-Dec distribution of galaxies

[ ] Create redshift histogram and identify cluster overdensity peaks

[ ] Separate cluster vs field galaxies

[ ] Run Bagpipes SED fitting (derive stellar mass, SFR, metallicity, age)

[ ] Generate plots: CMD, Mass-SFR diagram, quenched fraction statistics

[ ] Prepare outputs for dashboard (JSON, CSV, plots)

[ ] Test outputs with small sample datasets for correctness

Teammate 2 — JWST Spectrum Analyzer

[ ] Run JWST pipeline: Raw → Detector1 → Spec2 → Spec3

[ ] Extract 1D spectrum (Flux vs Wavelength)

[ ] Fit spectra using Bagpipes → SFH, stellar mass, SFR, metallicity, dust, age ± uncertainties

[ ] Generate plots and tables for dashboard

[ ] Prepare outputs for dashboard (JSON, CSV, plots)

[ ] Test outputs with small JWST demo datasets

---

STEP 3: INTEGRATION & DASHBOARD (Shared)

[ ] Collect outputs from Cos-Evo, Cluster Analyzer, and JWST modules

[ ] Standardize data formats for integration

[ ] Generate comparative statistics:

- Cosmic timeline vs observed galaxies

- Cluster vs field star formation efficiency

- Galaxy evolution storyline across modules

[ ] Prepare data and plots for frontend dashboard visualization

---

STEP 4: REPORT GENERATION & DOWNLOADS (Shared)

[ ] Generate downloadable CSV/JSON with all derived properties

[ ] Save user-generated reports (plots + notes)

[ ] Link report/download functionality to frontend buttons ([Download Results], [Save Report])

---

STEP 5: MAINTENANCE & OPTIMIZATION (Shared)

[ ] Implement error handling for invalid inputs or failed computations

[ ] Queue/manage heavy computational jobs asynchronously

[ ] Cache intermediate results to prevent redundant computations

[ ] Monitor logs for debugging and pipeline performance tracking

---

STEP 6: MODULE OWNERSHIP & COLLABORATION

[ ] Teammate 1: Cos-Evo + Cluster module

[ ] Teammate 2: JWST Spectrum module

[ ] Shared modules: utils.py, dashboard\_integrator.py, report\_generator.py

[ ] Use Git branches:

- cos-evo-module

- cluster-analyzer-module

- jwst-spectrum-module

[ ] Commit frequently with clear messages

[ ] Avoid committing large files (use .gitignore for FITS and intermediate outputs)

[ ] Merge into main branch only after code review and testing

---

STEP 7: LOGGING & DEBUGGING

[ ] Implement info logs for successful pipeline steps

[ ] Implement warning logs for minor issues

[ ] Implement error logs for failed computations

[ ] Ensure logs are saved to a standard folder (logs/) for debugging

---

STEP 8: TESTING & VALIDATION

[ ] Provide small sample datasets for each module

[ ] Define expected output types and sanity checks:

- Number of galaxies in cluster analysis matches catalog

- SED outputs contain stellar mass, SFR, metallicity

- JWST spectra have correct wavelength and flux arrays

[ ] Test modules independently before integration into main.py

---

STEP 9: REFERENCES & NOTES

[ ] py21cmfast Documentation: https://21cmfast.readthedocs.io/

[ ] Bagpipes Documentation: https://bagpipes.readthedocs.io/

[ ] JWST Pipeline Documentation: https://jwst-pipeline.readthedocs.io/

[ ] Include example commands to run each module locally

---

SEQUENTIAL FLOW SUMMARY:

User Input → Input Handler → Cos-Evo → Cluster Analyzer → JWST Spectrum → Dashboard → Report/Download → Maintenance/Optimization