GitHub Repository

Repo Link: `https://github.com/PATELOM925/bxtheory`

Exam Study Planner Code Documentation

1) What This System Does

This project is a multi-agent exam study planner built with Google ADK.

It takes multiple course PDFs (midterm overviews, syllabi, textbooks), extracts exam scope, asks Human-in-the-Loop (HITL) questions, estimates required study hours, builds a day-by-day schedule, and exports:

- `study_plan.csv`
- `study_plan.md`

Artifacts are available in:

- ADK Web Artifacts panel (session-scoped)
- Local output folder: `bxtheory/multi_tool_agent/outputs`

2) High-Level Architecture

The system uses a 5-agent architecture with one orchestrator:

OrchestratorAgent
File: bxtheory/multi_tool_agent/agents/orchestrator.py
Owns workflow sequencing, state persistence, and final export.

IngestionAgent
File: bxtheory/multi_tool_agent/agents/ingestion.py
Registers files once per session, parses text, extracts course specs.

HITLAgent
File: bxtheory/multi_tool_agent/agents/hitl.py
Builds intake questions and applies user-driven overrides and profile adjustments.

EstimatorAgent
File: bxtheory/multi_tool_agent/agents/estimation.py
Converts topic/chapter scope into estimated hours.

PlanningAgent + FormattingAgent
Files:
bxtheory/multi_tool_agent/agents/planning.py
bxtheory/multi_tool_agent/agents/formatting.py
Planning builds the schedule and feasibility warnings; formatting exports deterministic CSV/Markdown.

Root ADK setup is in:

- `bxtheory/multi_tool_agent/agent.py`

3) End-to-End Flow

Typical one-shot flow uses `run_study_planner`:

1. `register_files`
2. `extract_course_specs`
3. `set_constraints`
4. `apply_hitl_profile` (optional, or required when `require_hitl_profile=true`)
5. `apply_hitl_edits` (optional)
6. `estimate_topic_hours`
7. `build_plan`
8. `export_plan`

This is implemented in `OrchestratorEngine.run_study_planner`.

4) Session State Contract

State is JSON-serializable and stored in ADK session state.

Keys:

- `files_by_sha`: hash-indexed file references
- `course_specs`: extracted course structure
- `constraints`: normalized planning constraints
- `topic_estimates`: per-topic estimated effort
- `plan_rows`: day-by-day plan rows
- `plan_summary`: totals + feasibility + warnings
- `hitl_history`: capped history of prompts/notes (max 10)
- `hitl_profile`: latest user intake answer payload

5) Upload-Once and Caching Behavior

File deduplication is enforced using SHA-256:

- Hash computed via `sha256_file(...)`
- Existing hash in `files_by_sha` => file reused (no re-upload)
- New hash => optional Gemini Files upload (if enabled), else local ID fallback

Relevant files:

- `bxtheory/multi_tool_agent/tools/hash_cache.py`
- `bxtheory/multi_tool_agent/tools/gemini_files.py`
- `bxtheory/multi_tool_agent/agents/ingestion.py`

6) Parsing and Extraction Logic

Ingestion uses layered extraction:

- Midterm overview (highest confidence)
- Syllabus (fallback)
- Textbook (topic fallback)
- Default topics + fallback exam date if missing

Extraction utilities:

- `extract_text` (pdf text + OCR fallback)
- `extract_exam_date`
- `extract_topics`
- `fallback_exam_date`

Source:

- `bxtheory/multi_tool_agent/tools/pdf_ingest.py`

7) HITL Personalization

HITL collects:

- Ranked course priority
- Familiarity score per course (1 to 5)
- Coverage percent per course (0 to 100)
- Weakness score per course (1 to 5)
- Optional weekday/weekend hours update

These values are blended into `priority_weights` and timing constraints to personalize scheduling.

8) Estimation Model

Per-topic estimation uses:

- textbook page count (when available)
- pages-per-hour baseline
- course difficulty multiplier
- task-mix multiplier
- profile multipliers from HITL (familiarity, weakness, coverage)

Output type:

- `TopicEstimate(course_code, topic_id, estimated_hours, confidence, basis)`

9) Planning Model

Planner behavior:

- Builds schedule from `start_date` to latest exam date
- Reserves pre-exam final-review buffer

- Assigns study in chunks, adds spaced review cadence
- Applies urgency, shortfall pressure, and user profile weighting
- Detects infeasibility and returns exactly 3 mitigation recommendations

Output:

- `PlanRow[]` + `PlanSummary`

10) Exports and Artifacts

`export_plan` creates deterministic outputs:

- `study_plan.csv` (stable headers/order)
- `study_plan.md` (summary, warnings, day-by-day table)

Then orchestrator publishes both files as ADK web artifacts with `tool_context.save_artifact(...)`.

11) Type Contracts

Pydantic schemas:

- `FileRef`
- `TopicSpec`
- `CourseSpec`
- `UserConstraints`
- `TopicEstimate`
- `PlanRow`
- `PlanSummary`

Defined in:

- `bxtheory/multi_tool_agent/models/schemas.py`

12) Testing Coverage

Current tests validate:

- Upload-once cache behavior
- State persistence across stages
- HITL profile application/persistence
- Graceful handling when no files are provided
- `hitl_history` cap behavior
- Warning for unknown file kinds
- Planner infeasible path with exactly 3 options
- Planner balancing across courses when feasible
- Deterministic export contract
- ADK artifact publishing

Tests located in:

- `bxtheory/multi_tool_agent/tests/test_state_cache.py`
- `bxtheory/multi_tool_agent/tests/test_planner.py`
- `bxtheory/multi_tool_agent/tests/test_exports.py`

13) Operator Runbook (Demo + Real Usage)

This section contains the exact commands used for an end-to-end demo.

13.1 Start Environment and ADK Web

```
# Go to your repo root (folder that contains bxtheory/) cd <repo-root> # Activate project virtual environment (adjust if your venv path is different) source bxtheory/bin/activate # Enter agent module cd bxtheory/multi_tool_agent # Launch ADK web UI adk web .
```

What this does:

- Starts the app in ADK web mode at `http://127.0.0.1:8000`.
- Uses your configured `.env` model/provider settings.

13.2 In ADK Chat: One-Shot Plan Generation (with HITL Required)

Paste this in the ADK chat box:

```
Call run_study_planner with: file_paths=[ "data/input/midterm_topics/SYSD 300 - Midterm 1 Overview.pdf", "data/input/midterm_topics/PHYS 234 - Midterm 1 Overview.pdf", "data/input/midterm_topics/HLTH 204 - Midterm 1 Overview.pdf", "data/input/syllabi/SYSD 300 - Syllabus.pdf", "data/input/syllabi/PHYS 234 - Syllabus.pdf", "data/input/syllabi/HLTH 204 - Syllabus.pdf", "data/input/textbooks/John D Sterman - Business Dynamics Systems Thinking and Modeling for a Complex World-McGraw-Hill Higher Education 2000.pdf", "data/input/textbooks/David H McIntyre_ Corinne A Manogue_ Janet Tate_ Oregon State Un - Quantum mechanics _ a paradigms approach (2012, Pearson ).pdf", "data/input/textbooks/Marc M. Triola, Mario F. Triola, Jason Roy - Biostatistics for the Biological and Health Sciences (2nd Edition) (2017, Pearson).pdf" ] constraints_json='{"start_date":"2026-02-09","hours_weekday":3,"hours_weekend":6}' require_hitl_profile=true
```

What this does:

- Registers all files with upload-once SHA caching.
- Extracts course specs and blocks final plan until HITL profile is provided.
- Sets baseline daily study-time constraints.

How to tweak:

- Change `start_date` to shift the entire plan horizon.
- Increase/decrease `hours_weekday` and `hours_weekend` for feasibility.
- Set `require_hitl_profile=false` if you want a quick non-personalized first draft.

13.3 In ADK Chat: Apply HITL Profile

Paste this after the first call:

```
Call apply_hitl_profile with: answers_json='{ "ranked_courses":["PHYS234","SYSD300","HLTH204"],
"familiarity_by_course":{"PHYS234":2,"SYSD300":3,"HLTH204":4},
"coverage_by_course":{"PHYS234":25,"SYSD300":40,"HLTH204":60},
"weakness_by_course":{"PHYS234":5,"SYSD300":3,"HLTH204":2},
"hours":{"hours_weekday":3,"hours_weekend":6} }'
```

What this does:

- Updates priority and effort multipliers based on learner profile.
- Persists profile in session state for subsequent runs.

How to tweak (HITL knobs):

- `ranked_courses`: reorder to prioritize a different exam.
- `familiarity_by_course`: lower value => more planned effort.
- `coverage_by_course`: higher value => reduced remaining effort.
- `weakness_by_course`: higher value => more effort allocation.
- `hours`: override day capacity without changing other settings.

13.4 In ADK Chat: Regenerate with Explicit Override

Paste this to force a final prioritization tweak:

```
Call run_study_planner with: file_paths=[ "data/input/midterm_topics/SYSD 300 - Midterm 1
Overview.pdf", "data/input/midterm_topics/PHYS 234 - Midterm 1 Overview.pdf",
"data/input/midterm_topics/HLTH 204 - Midterm 1 Overview.pdf", "data/input/syllabi/SYSD 300 -
Syllabus.pdf", "data/input/syllabi/PHYS 234 - Syllabus.pdf", "data/input/syllabi/HLTH 204 -
Syllabus.pdf", "data/input/textbooks/John D Sterman - Business Dynamics Systems Thinking and Modeling
for a Complex World-McGraw-Hill Higher Education 2000.pdf", "data/input/textbooks/David H McIntyre_
Corinne A Manogue_ Janet Tate_ Oregon State Un - Quantum mechanics _ a paradigms approach (2012,
Pearson ).pdf", "data/input/textbooks/Marc M. Triola, Mario F. Triola, Jason Roy - Biostatistics for
the Biological and Health Sciences (2nd Edition) (2017, Pearson).pdf" ]
constraints_json='{"start_date":"2026-02-09","hours_weekday":3,"hours_weekend":6}'
hitl_overrides_json='{"priority_weights":{"PHYS234":1.35,"SYSD300":1.00,"HLTH204":0.90}}'
hitl_note='Prioritize PHYS due to low familiarity and high weakness.'
```

What this does:

- Reuses cached files and profile in the same session.
- Applies explicit final priority weights before planning.
- Regenerates both outputs and artifacts.

How to tweak:

- Increase `PHYS234` above `1.35` for stronger priority boost.
- Keep values around `0.55` to `1.90` for stable weighting behavior.
- Use `hitl_note` to leave an audit trail in session history.

13.5 Verify Outputs

```
# Local output files ls -lh outputs # Quick CSV preview head -n 12 outputs/study_plan.csv
```

Also check ADK web Artifacts panel for:

- `study_plan.csv`
- `study_plan.md`

13.6 Stop the App

```
# In the terminal running adk web Ctrl + C
```