

MCAT Study Schedule - Backend Dev Needed - Test Task to Gauge Competency for Larger Project

Full MCAT Study Schedule Planner — Coding Test Task (Back-end Only)

This task is to help determine a suitable developer for our project.

Successful completion of this will prove competency and will be followed up with a larger offer to help complete the remainder of the project (version-control, integration with front-end, analytics and schedule optimization phase, etc). You will only be responsible for back-end work, we have a separate front-end developer who you would potentially work with in the future.

Goal:

Build one endpoint that returns a complete schedule from start to test (about 3+ months). The plan must respect availability, split study days into Phase 1 → Phase 2 → Phase 3, place 6 AAMC full length practice tests (FLs), and pack each study day to a 5-hour day including 1 hour of written review.

Use Organized_MCAT_Topics.xlsx (access will be given once developer is selected to attempt task) to create a Supabase / Postgres database.

Phase 1 = content review (articles, videos, textbook) + discrete questions + 2 Jack Westin CARS passages + one-hour review

Phase 2 = passage-based questions + discrete questions + 2 Jack Westin CARS passages + one-hour review

Phase 3 = AAMC only - science question sets + 2 AAMC CARS passages + one-hour review

Discrete questions: no reading, student directly answers questions based on knowledge

Passage-based: must do reading, then answer questions related to it

AAMC: the organization that creates the MCAT test, their materials are mandatory and must be completed in 3rd phase

FL's: Full-length practice tests, must be given only on selected day for this (Saturday) and no other work can be assigned that day.

Endpoint (input):

GET /full-plan?start_date=YYYY-MM-DD
 &test_date=YYYY-MM-DD
 &priorities=1A, 1B, 1D, 3A, 3B, 4A, 4B, 5A, 5D, 5E, 6B, 7A, 9B // category order, highest first
 &availability=Mon,Tue,Thu, Fri, Sat // study days; others = break
 &fl_weekday=Sat // day to schedule AAMC FLs (full-length practice tests)
* start_date: first calendar day to plan
* test_date: MCAT exam day (exclusive for scheduling)
* priorities: content categories in priority order (e.g., 1A, 1B, 1D...)
* availability: weekdays you plan (others = break)
* fl_weekday: weekday for Full Lengths (e.g., Sat)

Output (JSON):

Return an array of dated day objects from start to the day before test. Each date is one of:

* Break day:

```
"date":"2025-10-06", "kind":"break"
```

* Full length day:

```
"date":"2025-10-11", "kind":"full_length", "provider":"AAMC", "name":"FL #3"
```

* Study day (Phase 1):

```
"date":"2025-10-07", "kind":"study", "phase":1,  
"blocks":  
  "science_content":["...KA video..., ...KA article, ...Kaplan section...],  
  "science_discretes":["...KA set..., ...ThirdParty set...],  
  "cars":["...passage..., ...passage...],  
  "written_review_minutes": 60,  
  "total_resource_minutes": 240
```

* Study day (Phase 2):

```
"date":"2025-10-08", "kind":"study", "phase":2,  
"blocks":  
  "science_passages":["...", ...],  
  "uworld_set": [...10Q...],  
  "extra_discretes":["...not used in Phase1...],  
  "cars":["...", ...],  
  "written_review_minutes": 60,  
  "total_resource_minutes": 240
```

* Study day (Phase 3):

```
"date":"2025-11-20", "kind":"study", "phase":3,  
"blocks":  
  "aamc_sets":["...20–30Q pack A...", ...20–30Q pack B...],  
  "aamc_CARS_passages":["...", ...],  
  "written_review_minutes": 60,  
  "total_resource_minutes": 240
```

Timeline & day types:

1. Build the calendar from start → test
2. Mark study vs break by availability.
3. Count study days only and split evenly into thirds:
 - * Phase 1 (first third, in order)
 - * Phase 2 (second third)
 - * Phase 3 (final third)
4. Insert 6 AAMC FLs on fl_weekday, evenly spaced from start to test, no FL in last 7 days. On FL days, no other blocks.

Global rules:

Daily time budget:

- * Each study day has 5 hours (300 min).
- * Always reserve 60 min for Written Question Review (end of day).
- * Remaining resource budget = 240 min.
- * Use per-row times from the Excel

High-Yield first

- * A concept key is High-Yield if any row at that concept has high_yield = Yes.
- * Phases 1 & 2: use High-Yield concepts only under priorities.
- * If a required slot can't be filled with High-Yield that day, you may use Low-Yield for that slot only.

Never repeat (Phases 1–2)

- * The same resource from Khan Academy, Kaplan, or Jack Westin may appear once total across P1+P2 (schedule-wide).
- * Keep used_resources in a separate table and keep track of them.
- * UWorld question sets can repeat while sets remain (# of questions)
- * Phase 2 discretized must be not used in Phase 1.

Specificity priority (for matching):

- * Concept (preferred), otherwise Subtopic, then Category
- * Deliver more specific → more general

*Time-fit

CARS Provider rank (use only when relevant):

- * CARS passages: Phases 1 + 2 = Jack Westin, Phase 3 = AAMC only

Same-day de-duplicate:

- * Do not place the same resource twice on the same day (across blocks).
- * Phase 3: the two AAMC sets must be from different packs unless nothing else is left.

What to schedule each study day:

Phase 1 (science first → CARS → review):

Goal: always pair Kaplan with matching Khan Academy content.

1. Science content (matched to same anchor key):
 - * 1 Kaplan section, plus matching Khan Academy content items (videos + articles)
2. Science discretely (same anchor key):
 - * 1 Khan Academy or Jack Westin discrete set
3. CARS: 2 Jack Westin passages
4. Written Question Review: 60 min (outside the 240)

Phase 2 (science first → CARS → review):

1. Science passages: 2 third-party (same category/subtopic as anchor is fine)
2. UWorld: 1 set (10Q)
3. Extra discretely not used in Phase 1: 1–2 × discrete set (KA or Jack Westin)
4. CARS: 2 Jack Westin passages
5. Written Question Review: 60 min

If tight: keep 2 science passages + UWorld 10Q, then add 1 new discrete, then CARS; drop a second discrete first if needed.

Phase 3 (AAMC only → review):

1. AAMC sets: 2 × (20–30Q) from different packs
2. AAMC CARS passages: 2
3. Written Question Review: 60 min

Anchor & candidate selection (Phases 1–2):

1. Pick an anchor concept for the day from the High-Yield pool in the highest-priority category with High-Yield remaining.
2. Phase 1 science picks must match the same anchor; if any science slot can't fill, widen together: Concept → Subtopic (AA.B.x) → Category (AA.x.x).
3. Phase 2 science may use the anchor (same subtopic OK). CARS is independent.
4. For each slot, build the candidate pool from the anchor via fallback, filter by slot type, never-repeat, supply, then sort by:
 - * Specificity
 - * Numeric key order
 - * Time-fit
 - * Provider rank (if applicable)
 - * Title A→Z

Packing the 240 minutes: add items in the required order for the phase; skip an item if it would exceed 240, then continue.

Deliverables:

GitHub repository containing:

- * Small Node/TypeScript service exposing /full-plan.
- * Supabase / Postgres DB using XLSX
- * Clear documentation and three sample calls.

How we'll evaluate:

- * ☒ Availability respected (correct break vs study days).
- * ☒ Phases split by study-day count in order: P1 → P2 → P3.
- * ☒ Six AAMC FLs: on fl_weekday, evenly spaced, none in last 7 days.
- * ☒ Phase 1: Kaplan is matched with relevant KA content + 2 Jack Westin CARS passages + review; KA/Kaplan/Jack Westin never repeat across P1+P2
- * ☒ Phase 2: includes passages, UWorld set, and discretely not used in P1 + 2 Jack Westin CARS passages + review
- * ☒ Phase 3: AAMC only; two sets from different science packs per day + 2 AAMC CARS passages
- * ☒ Daily time budget honored: 240 min resources + 60 min review.
- * ☒ High-Yield-first enforced (Low-Yield only if High-Yield can't fill a slot that day).
- * ☒ Tie-breaks followed: Concept, Subtopic, Category, numeric key order, time-fit, provider rank.
- * ☒ Deterministic: same inputs → same plan.
- * ☒ Clean JSON + DB + documentation.

The pay for successful completion of the task is \$100. The task will be given to multiple developers to compare results.

Once a developer is selected that completes the task satisfactorily, the total budget for the backend development suitable for production is \$3000 among multiple phases and credited as a co-founder of the software.

Please let us know if you'd like more information / clarification and we'd like to set up a 15 minute call to discuss the task before starting.