

API Endpoints

READ

Schedule

1. Get all for this week
 - a. param - start date, end date
2. Get all for a particular date
 - a. Get for today
3. Get all for a particular assignment
 - a. get remaining

Assignment

1. List all assignments
 - a. filters: course, due date, priority, status
 - b. sorting
2. List overdue assignments

// READ APIs

READ

R1 — List Assignments

Goal: View assignments with basic status.

Trigger: Student opens "All Assignments."

Preconditions: None.

Main Success: Filter/sort; compute on-track/behind per item; return lightweight tasks count.

Postconditions: None.

IDs: GET /assignments?status=&courseId=&dueFrom=&dueTo=

R2 — Get Assignment Detail (+Tasks)

Goal: Inspect one assignment and its planned/actual tasks.

Main Success: Return assignment, future/past tasks, remaining effort, risk.

IDs: GET /assignments/{id}?includeTasks=true

R3 — Schedule Overview (Dashboard)

Goal: Snapshot of next 7–14 days.

Main Success: Return ordered task timeline, capacity per day, risks, suggested next actions.

IDs: GET /schedule/overview

CREATE

Assignment

1. Create a new assignment
 - a. title, description, course, due-date, priority, MEDIA (Image or PDF)
 - i. AI controller calculates estimated hours and complexity
2. Create assignments in bulk
3. Empty Slot
 - a. duration, date, time

b. update assignment/schedules automatically

C1 — Create Assignment (Manual)

Goal: Add a new assignment and generate its task plan.

Trigger: Student submits create form.

Preconditions: Student exists; valid due_at.

Main Success:

1. Validate payload (title, due_at, optional est_effort_min, course_id).
2. Infer estimate if missing; normalize timezones.
3. Generate task plan (split into blocks within work hours before due_at).
4. Merge into calendar; resolve conflicts by shifting lower-priority future tasks.
5. Persist Assignment, Task[]; record ScheduleVersion.

Postconditions: Assignment is planned; tasks appear in schedule.

Alternates: Tight window ⇒ flag risk and compress plan; missing estimate ⇒ placeholder task + prompt.

IDs: POST /assignments

C2 — Create Assignment (LMS Import → Confirm)

Goal: Turn an imported item into an active assignment.

Trigger: LMS Importer creates draft; Student taps "Add."

Main Success: Same as C1 starting at step 3 (plan/merge/persist).

Postconditions: Draft marked "activated"; dedup key stored.

IDs: POST /assignments/activate

C3 — Bulk Create (Multiple Assignments)

Goal: Add N assignments at once and globally re-balance.

Trigger: Student pastes syllabus or confirms LMS batch.

Main Success: Validate all → plan each → global conflict resolution → persist batch with one ScheduleVersion.

IDs: POST /assignments/bulk

UPDATE

Assignment

1. Update assignment metadata
 - a. request-body: assignment-id
 - b. title, description, course, priority
 - i. AI controller calculates estimatedhours and complexity
2. Update assignment priority, complexity, time-left, due-date
3. Adaptive Update
 - a. extend the schedule
 - b. shift others?

UPDATE API

UPDATE

U1 — Update Assignment Metadata

Goal: Edit title, notes, course, priority (no time changes).

Trigger: Student saves edits.

Main Success: Persist fields; no replan; bump updated_at.

IDs: PATCH /assignments/{id}

U2 — Update Estimate / Scope

Goal: Change est_effort_min or complexity.

Main Success: Recompute **remaining** effort = new_est – actual_logged; regenerate **future** tasks only; merge; persist ScheduleVersion.

IDs: PATCH /assignments/{id} (body includes estEffortMin)

U3 — Update Due Date

Goal: Move due date earlier/later.

Main Success: Re-plan remaining tasks to meet new due_at; shift competing tasks within constraints; warn if infeasible.

IDs: PATCH /assignments/{id} (body includes dueAt)

U4 — Adaptive Update (Overrun During Work) - App driven

Goal: Heal plan when actual > planned.

Trigger: Session end or mid-task threshold.

Main Success: Offer strategy (extend now / add follow-up / shift others); apply; update tasks; persist ScheduleVersion.

IDs:

- POST /tasks/{taskId}/sessions (telemetry)
- POST /schedule/adapt (strategy)

U5 — Manual Task Adjust/Reorder (Optional) - V1

Goal: Student drags a planned block.

Main Success: Validate constraints; move block; ripple minimal shifts; record reason.

IDs: POST /tasks/reorder

DELETE

1. Particular Assignment
2. Batch Assignments

DELETE

D1 — Delete Assignment (Incorrect Addition)

Goal: Remove an assignment and free time.

Trigger: Student taps delete.

Main Success: If sessions exist → soft-delete (status canceled), keep history; otherwise hard-delete. Remove **future** tasks; choose strategy: leave buffer or pull forward others. Persist ScheduleVersion.

IDs: DELETE /assignments/{id}?strategy=pull_forward|leave_buffer

D2 — Delete Task (One Block) V1

Goal: Remove a single planned block without killing the assignment.

Main Success: Delete task; optionally re-plan remaining effort; log reason.

IDs: DELETE /tasks/{id}

D3 — Undo Delete (Grace Window) V1

Goal: Quick restore after accidental delete.

Main Success: Restore assignment/tasks from last ScheduleVersion.

IDs: POST /schedule/versions/{versionId}/restore