

**Page/API Mapping**

Course:

Advanced Programming

Prepared By:

Eliane Jreij (202212259)

Tony Katrib (202112491)

Youssef Farah (202110800)

Presented To:

Dr. Adib Haddad

Mejdlaya 2025

**Signup Page:**

* /signup: create a user using full\_name, email and password (includes a user\_ID and created\_at) returns to login after successful signup.

**Login Page:**

* /login: user login using email and password, include token JWT, for better authentication. Navigates to user dashboard.

**Dashboard:**

* /logout: user logout and remove the token.
* /get-workspaces: fetch workspaces of the logged-in user (workspaces => admin, member or viewer).  
  Workspaces will be shown filtered by role firstly admin then member then viewer.
* /create-workspace: user create a workspace (name, description (optional) and created\_by\_user\_ID) besides created\_at and workspace\_ID.
* /count-workspaces-by-role: get the count of workspaces of logged-in user by each role (ex: workspaces as admin = 2, workspaces as member = 1, workspaces as viewer = 0).
* /get-notifications: show all notifications of the logged-in user, notifications are when assigning to user, project deadline approach and not status “completed” or “archived”, duedate of task approach and still not marked “Done” or “Removed”, on comment, on mention…
* /count-not-read-notifications: count of false “isRead” in notifications to display the number.

**Workspace:**

* /add-user: workspace admin adds users as members or viewers, through workspace settings and users tab, the that user will be added should have an account already, he will be assigned a role (member or viewer) and filled automatically “joined\_at”.
* /create-project: user (only admins and members of a workspace) can create a project. Getting the workspace\_ID as FK, and have as attributes name, description (optional), start\_date, deadline, status as enum(Unstarted, Active, Completed, Archived) and created\_at.
* /get-users: fetch users of the workspace.
* /get-projects: fetch projects of the workspace.
* /create-tag: user (admin or member only) can create tags, tags are used to identify and class a task (if the user want) like bug, frontend… Creating a tag involve name, color\_code (hex), created\_by\_user\_ID (FK), and workspace\_ID (FK) as they are created on workspace tier.
* /delete-workspace: only admin of the workspace can delete it (change status to ‘removed’)
* /edit-workspace: only admin can edit the workspace details.

**Project and Task:**

* /create-task: user (admin or member) creates tasks for the project. A task has task\_ID, project\_ID (FK), created\_by\_user\_ID (FK), assigned\_to\_user\_ID (FK) (optional), parent\_task\_ID (FK) (optional this is only for subtasks), priority enum(Low, Medium, High, Critical), status enum(ToDo, InProgress, Done, Removed), duedate (optional), created\_at.
* /create-subtask: user (admin or member) creates a subtask parent\_task\_ID will be filled along the others.
* /assign-tag-to-task: user (admin or member only) assigns **tags** to the task from the defined tags in workspace.
* /add-comment-on-task: user (all roles => admin, member and viewers) adds comments to the task.
* /mention-user-in-comment: user that creating a comment can mention users of (inside the) workspace.
* /edit-project: user (admin or member) can edit project details and statuses.
* /delete-project: user (admin or member) can delete a project (change status to ‘removed’).
* /edit-task: user (admin or member) can edit task details and tags.
* /delete-task: user (admin or member) can delete a task (change status to ‘removed’).