Project Management user story

User Story: Project and Task Management with User Tracking

Description

As a user of the project management platform, I want to be able to create projects and assign tasks, as well as assign users to tasks, so that I can efficiently organize my work and collaborate with other users.

System Requirements

1. Users

- Users can register, log in, and be assigned to tasks.
- There are no differentiated roles.
- Users can only see their own projects on the dashboard.

User fields:

- `first_name` (string, required)
- `last_name` (string, required)
- `email` (string, unique, required)
- `password` (string, required)
- `milestone` (integer, updated every 100 users)

2. Projects

- A user can create multiple projects.
- A project can have multiple tasks.
- Only the user who created a project can delete it.
- A project can only be deleted if all its tasks are in the **pending** state.

Project fields:

- `name` (string, required)
- `description` (string, optional)
- `user_id` (integer, required, references users)

3. Tasks

- A user can create tasks within a project.
- A task can be assigned to a specific user or remain unassigned initially.
- Tasks have the following statuses:
    - `pending` (Default when created)
    - `in_progress` (When in development)
    - `completed` (When finished)

- Only the user who created the project can delete tasks from it. Otherwise, the system should return an "Unauthorized" message.
- Tasks are created in the **pending** state by default.
- Tasks can only be deleted if they are in the **pending** state.
- Tasks can be created without an assigned user, and they can be assigned later when edited.

Task fields:

- `title` (string, optional, max. 255 characters)
- `description` (string, optional)
- `due_date` (date, optional)
- `status` (string, allowed values: `'pending'`, `'in progress'`, `'completed'`)
- `project_id` (integer, required, references projects)
- `user_id` (integer, optional, references users)

Interactive Mockups

Create mockups with a prototype of the views that allow navigation between screens, simulating the real experience, but without functional logic.

Include them in the user_story/interactive_mockups folder.

Acceptance Criteria (Using BDD - Given/When/Then)

Scenario 1: Successfully creating a project

Given I am an authenticated user, **When** I send a request to create a new project with a valid `name` and `description`, **Then** the system should store the project in the database and return a `201 Created` response with the created project.

Scenario 2: Cannot delete a project with tasks in progress or completed

Given I am the creator of a project, **And** the project has tasks in the `in_progress` or `completed` state, **When** I attempt to delete the project, **Then** the system should respond with a `400 Bad Request` error stating: "Only projects without tasks or with pending tasks can be deleted."

Scenario 3: Assigning a user to an existing task

Given I am an authenticated user, **And** there is a task without an assigned user, **When** I edit the task to assign myself as the responsible user, **Then** the system should update the task and return a `200 OK` response with the updated task.

Scenario 4: A user can only see their own projects on the dashboard

Given I am an authenticated user, **And** I have created some projects, **And** there are other projects created by different users, **When** I access my dashboard or request the `/projects` endpoint, **Then** I should only see the projects I have created, **And** projects from other users should not appear in my list.

Scenario 5: A new task is created with a *`pending`* status by default

Given I am an authenticated user, **And** I want to create a new task within a project, **When** I send a request to create a task without specifying the `status` field, **Then** the system should store the task in the database with the `pending` status automatically.

Scenario 6: A task can be created without an assigned user

Given I am an authenticated user, **And** I want to create a new task in a project, **When** I send a request to create a task without specifying the `user_id` field, **Then** the system should store the task in the database with `user_id` set to `null`, **And** the task can be assigned later through an update.

Scenario 7: A user can assign themselves to a task after its creation

Given I am an authenticated user, **And** there is a task without an assigned user (`user_id` is `null`), **When** I send a request to update the task with my `user_id`, **Then** the system should assign me as the responsible user, **And** it should return a `200 OK` response with the updated task.

API Endpoints

Users

`POST /api/register` → Register a user.
`POST /api/login` → Log in.
`GET /api/users` → List users.
`GET /api/users/{id}` → Get a specific user.

Projects

`POST /api/projects` → Create a new project.
`GET /api/projects` → List all projects.
`GET /api/projects/{id}` → Get a specific project.
`PUT /api/projects/{id}` → Update a project.
`DELETE /api/projects/{id}` → Delete a project (only if all tasks are pending).

Tasks

`POST /api/tasks` → Create a new task.
`GET /api/tasks` → List all tasks.

`GET /api/tasks/{id}` → Get a specific task.
`PUT /api/tasks/{id}` → Update a task (including assigning a user).
`DELETE /api/tasks/{id}` → Delete a task (only if it is in pending state).