Rails and Angular fullstack development basics exercise

This exercise consists of two parts, where you are tasked to implement a very simple proof of concept application using Ruby on Rails powered JSON API backend and Angular frontend web application framework. The goal of these exercises to familiarize yourself on these frameworks, learn how to get your development environment up and running, and how to implement basic mechanics that typically exist in every single application you would encounter in a real project.

No technical help or step-by-step instructions are provided; searching the required information to complete the tasks is an essential part of this exercise.

Part 1: Ruby on Rails API backend service

For the sake of this exercise, let's assume that Oivan wants to develop an internal application to keep track of what projects each developer has been working on. The following user story describes the required functionality:

- As a user, I can create a new project in the database
    - Each project has a name, description, start date, end date (optional)
    - Each project has one or more developers (see below)
    - Each project has one or more technologies listed, like Angular or Rails (see below)
    - I **cannot** change start date after the project has been created
    - I can update the name and description and end date
    - I can delete the project (hint: consider soft delete)
- I can create a new developer in the database
    - Each developer has firstname and lastname
    - I can update the firstname and lastname of a developer
    - I can delete the developer from the database
    - Each developer belongs to at least one project, so there should never exist a developer in the database that is not part of at least one project
    - There should not be duplicate developers, that share the same firstname and lastname (hint: it is perfectly fine to use an additional helper field to ensure this)
- I can create new technology in the database
    - Each technology has a name
    - Each technology can belong to one or more projects, or no project at all (so it is possible to create a technology in the database without a project)
    - I can update the name of the technology
    - I can delete the technology from the database, **only if it doesn't belong to any project**

API requirements

The functionality described above should be accessible over a JSON api using a client app like Postman https://www.postman.com/product/api-client/ . The API should thus allow the following kind of actions:

- Create, update, read and delete projects
    - It is up to you, how do you implement adding the developers and technologies to the projects. You can utilize several request, where these are update to the model after it has been created, or just one, where the developers and technologies are included in the same request that creates the project

- Create, update, read and delete developers
  - Note that developers should always be assigned to at least one project entity, so logically you only need to implement a create endpoint for developers, if you want to create the developers after the project has been created, otherwise no separate "create developer" endpoint is needed
- Create, update, read and delete technologies

Other requirements and notes

- Every field in the models is required to have a valid value (look into validation in Rails)
  - The only optional field is the project end date, that can have an empty/null value
- Not every conceivable error scenario needs to be handled, as this is just a demo app. But one example of error handling should be included, to show that you are familiar on how to implement it.
- The backend service should be implemented with Ruby on Rails https://rubyonrails.org/ framework and PostgreSQL https://www.postgresql.org/ database. Setup your Rails project in "api only" mode. Linux OS is highly recommended to avoid common issues.

Bonus conditions

Utilize Test Driven Development model and include unit tests in your models as you implement them.

Part 2: Angular frontend application

In this part, you will add a simple frontend user interface that connects with your backend service. When you are ready, it should be possible to:

- List projects and see who developers worked in them, and what technologies were used in those projects
  - Update the properties of the project
  - Add and remove developers and technologies in the project
- Delete project
- Create new project with developers and technologies
- Create, List, Update and Delete developers (note that creating a developer without a project is not allowed)
- Create, List, Update and Delete technologies

Notes:

- Use Angular https://angular.io/ framework to implement your frontend application
- Your application should make request to your Ruby on Rails API backend as HTTP requests, with the data in JSON format - Angular provides excellent tooling for this
- Hint: look into Angular forms
- Hint 2: you will need an API Proxy configuration in your Angular project
- Feel free to use any UI component library of your choosing, like Angular Material https://material.angular.io/ if you like

Bonus conditions

- Add validation with guiding messages into your forms
- Display success and error messages to the user