A. Els professors han de poder marcar els aprenentatges per a tenir un seguiment de què s’ha treballat.

1. Design System and Theme

- **Task:** Implement the color palette and base styles.
    - Define global styles for white, yellow, and black.
    - Set up a basic layout for the page (header for inputs, body for tables, and sections).
    - Ensure responsive design using Flexbox or Grid.

2. Create Input Section

- **Task:** Develop the input form for the project details.
    - Create an Angular form (Reactive or Template-driven) with the following fields:
        - Title of the SDA or project.
        - Short description.
        - URL input for the Drive directory/file.
        - Date inputs for start and end dates.
    - Add form validation (e.g., required fields, URL format, date range validation).
    - Implement a service to handle form data submission.

3. Create Vectors Table

- **Task:** Build the vectors table with checkboxes and progress tracking.
    - Use an Angular table component (<table>).
    - Add checkboxes for each vector, using ngFor to loop through vector options.
    - Bind each checkbox to a form control and use two-way data binding ([(ngModel)]).
    - Display progress as x/y dynamically, updating whenever a checkbox is selected/deselected.
    - Create a function to calculate and display the progress.

4. Create Rubric Section with Competency Table

- **Task:** Build the rubric table for competencies and specific criteria.
    - Similar to the Vectors table, create a table with checkboxes for each competency.
    - Use nested loops with ngFor to iterate through both the competencies and their specific criteria.
    - Implement logic to ensure selecting a criterion automatically checks the related competency.
    - Display progress for each competency as a percentage (ex:, use a getProgress function to calculate this).
    - Display the percentage at the top of each competency section.
    -

7. Checkbox Selection Logic

- **Task:** Implement logic to auto-select related specific competencies when criteria are selected.
  - Set up event binding to listen for checkbox changes.
  - Create logic in the component to select or deselect related checkboxes based on criteria selections.
  - Ensure the progress value updates in real-time.

8. Progress Tracking and Display

- **Task:** Develop a dynamic progress tracking system for both the vectors and competencies.
  - Create helper functions in the component to track how many checkboxes are selected.
  - Display the total progress dynamically as a fraction for vectors (ex:, 4/7) and as a percentage for competencies.

9. Define Template

- **Task:** Define the build as a template that new projects will pull.
  - Create a reusable Angular component for the form, vectors table, and rubric section.
  - Abstract the form fields, checkboxes, and progress tracking into reusable components to improve maintainability and reusability.
  - Ensure that the template includes dynamic inputs, so the fields and tables can adapt based on the project type.
  - Implement input decorators (@Input) to pass in project-specific data and settings when the template is used.
  - Make the template modular by defining Angular modules or using shared services for form data, table content, and rubric logic.

10. Pull Template and Data

- **Task:** Upon the creation of a new project, pull the template alongside the respective data from the database.
  - Implement a service to retrieve project data from the database (use HttpClient for API requests or connect to Firebase/other database).
  - Use the service to fetch specific data (e.g., project title, description, vectors, competencies, etc.) when a new project is created.
  - Inject the retrieved data into the template components by passing them through @Input properties or services.
  - Ensure that the form and tables are populated dynamically with the data received from the database.
  - Handle loading states and errors in case the data fetching fails, providing feedback to the user.
  - Implement form submission logic to save any new or modified data back to the database after editing a project.