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

Firstly, we have to create a UI for the teachers to visualize said tasks/learnings, we decided to create a sleek design with a visually attractive palette of colors (white, yellow and black). At the top of the page there are some inputs that the teachers have to fill in, these are: The title of the SDA (or the project) followed by a short description of the project and the URL to the drive directory/file. They then should add a starting and an ending date to the adjacent inputs.

Below that section, we can find the Vectors table, which is basically a small table for the teachers to select the values that have been worked on already via the use of checkboxes. These also have a progress value (Ex: 4/7) to let the teachers easily know how many are left.

After that section, we can finally find the rubric that contains the competencies that have to be completed during the project in question. To do this, the teachers use the exact same checkbox method as the vectors table with the addition that, since every competency is divided in specific competencies and evaluation criteria, once one of the latter are selected the related specific competency will be selected as well. Both of these also contain a progress value that is "translated" into a percentage that is shown at the top row of the section of that competency.

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.