HYP 2024-25

Project Implementation – General Instructions – exam session June July 2025

See also document **HYP 2023-2024-application domain specifications - June-July 2024**

Front-end

The scope of the web technology project is the development of a website for a Yoga center.
The project **MUST** be implemented using Vue3 and Nuxt3. **No other framework will be accepted.**

Since Vue/Nuxt are compatible with JavaScript and TypeScript, you can decide which one to use.

The code should have comments.
The rendering mode used for the project MUST be explicitly indicated in the nuxt.config.ts.

Pages

The website must contain at least these PAGES:

1. **HOMEPAGE**
2. **"ABOUT the CENTER" PAGE**
3. **"CONTACTS" PAGE**
4. **MULTIPLE TOPICS PAGES (Content from DB)**
    - MT1: Activity (Type) – AT LEAST **8** INSTANCES
    - MT2: Teacher – AT LEAST **5** INSTANCES
5. **GROUP PAGES (Content from DB)**
    - G1: "All Activities"
    - G2: "All Teachers"
    - G3: "HIGHLIGHTS" (3-4 activities that the center wants to promote)

Relations

Links for all relevant relationships **MUST BE** implemented:
A Teacher teaches in, or is responsible for, zero, one or more Activity
An Activity is under the responsibility of one Teacher/is toucht by one or more teachers

External Libraries

If a library ONLY uses CSS (and small functionalities to make the library work) you are free to import it into your project.
Examples:

- Bootstrap (CSS)
- Tailwind

You are allowed to use Components libraries.
Examples:

- Bootstrap-Vue
- Vuetify
- Buefy

The conditions for using these libraries are:

- You are still required to create your custom components.
- Don't use too many already available components. Prefer the most complex one (e.g., carousel) instead of the simple one that you can create easily by yourself.
- Tutors won't help you with problems related to these libraries. Check the documentation of the library.
- If you are in doubt about whether you can use a components library (outside of those in the examples), ask the tutors first.

Any other library or script outside what was agreed on is **NOT** allowed. Any external library you use that was not approved by us will lower the final score of your implementation project.

In general, if you want to add a library/module/script (outside of those presented during lectures) ask the tutor first.

Server

Even though it's not the focus of this course, a server **MUST** be implemented. For this reason, we suggest you extend the functionalities of the example project we provide you on the course's Repository on GitLab.

The project **MUST** be reachable online when you deliver the project for the exam.
There are multiple ways to achieve this, and you can choose any solution as long as the website works. The platforms presented during the course are:

- GitHub pages
- Vercel + Supabase

Remember, though, that while we will not evaluate the server implementation, we will evaluate its functionalities and correctness based on the evaluation criteria.

We **won't** evaluate your implementation project if it only works locally.

Database

The Database must be implemented following what you designed for the DB Design (see Design Project Specifications).

(Even if you work on a static website, we want you to generate pages by taking the data from a DB)

GitHub

Every team should host its code on a **private** GitHub repository (unless you are using GitHub Pages. In that case you can decide to have the projects on two repositories or use only a public one). All the instructors must be invited as viewers of the repository so that we can have access to your code for the evaluation.

All the team members are required to commit and push their work to the common team repository under their individual names so that instructors will be able to discern the actual contribution ratio for each of them.

What to deliver

All we ask is to develop the website and deliver a pdf file with:

1. Title page/Introduction pages containing:
    - Group name
    - Group composition
        - Name
        - Surname
        - Person code
    - link to your website
    - link to your GitHub repository
2. Description of how you decided to split the work in the team
3. Short Documentation

The documentation section **MUST** contain at least the following topics:

- Brief description of the project explaining the chosen theme for the summer school
- Hosting service used and why.
    - Rendering mode used (CSR, SSR, SSG) and why.
- Structure of the project
    - Structure of links/pages folder with short description
    - Available server endpoint with short description
- List of Components implemented with description, props and emit (if used)
- List of extra modules imported in the project (for example: Supabase, Pinia, Vuetify etc…) and how they were used.

Additionally, you can add descriptions (not mandatory) of:

- Extra functionalities implemented (e.g., store, filters, etc…)
- different approaches from what was discussed during lectures.
- Approaches used to comply with accessibility and SEO guidelines.

Evaluation

The grade will be assigned based on:

- **Documentation**
- **Vue/Nuxt**
  - **Usage of Vue/Nuxt Functionalities:** we will evaluate the different functionalities of Vue/Nuxt you used and how they are integrated in the project.
  - **Technical correctness of page implementation.**
  - **Components Structure and Reusability**: an application in Vue/Nuxt can be seen as a set of components. We will assess the number and organization of components used.
- **Website features**
  - **SEO optimization:** We will evaluate the choices made and how they contribute to the SEO.
  - **Responsiveness**: when changing to a mobile device, the interface adapts to the new size.
  - **Accessibility**: compliance with the accessibility guidelines (you can use Wave to check your own website as shown during the lecture).
- **CONSISTENCY WITH CONTENT REQUIREMENTS AND DESIGN SPECIFICATIONS.**