HYP 2020-2021

Project Implementation – General Instructions

Your starting point

- C-IDM schema in the large
- Content TABLES (content design in the small)
- Mappings Content-Tables->Pages
- P-IDM schema
- LOW FIDELITY WIREFRAMES (PAGE STRUCTURES)
- DB DESIGN: ER Schema and Logic design (RelationalTables))
- (High fidelity wireframes, if already created; you can start the implementation work while you are still designing these wireframes)

What do you have to do?

Client side

For the Client side implementation we required you to use **NuxtJS**.
You can find some examples here: https://mainstreaming.tv/, https://www.outsystems.com/
THE WHOLE CODE SHOULD BE COMMENTED IN ORDER TO FACILITATE THE COMPREHENSION AND EVALUATION FROM THE TEACHERS.

Pages
The website has to contain **at least**:

1. **HOMEPAGE**
2. **“ABOUT US” PAGE**
3. **“CONTACT US” PAGE**
4. **MULTIPLE TOPICS PAGES** (Multimedia content and links must be retrieved from DB)
   - **MT1: Areas – AT LEAST 2 INSTANCES**
   - **MT2: People – AT LEAST 3 INSTANCES**
   - **MT3: Services/Products – AT LEAST 2 INSTANCES**
5. **GROUPS PAGES** (Multimedia content and links must be retrieved from DB)
   - **G1: “All Areas”**
   - **G2: “All People”**
   - **G3: “All Services”**

Links
Links for **all relevant relationships** must be implemented:

- **R1-a Person→ Service**
- **R1-b Service → Person**
- **R2-a Area → Service**
- **R2-b Service → Area (**it can be 1-1 or 1-N)
- **R3-a Person → Area**
- **R3-b Area → Person**

Server side

For the Server side implementation you are required to develop a **simple web server**. This is mandatory in order to have **your application accessible online**. The implementation of this part **WILL NOT BE EVALUATED**

You can use any server side technology. We suggest you to use **NodeJS with Express**. For hosting your application we suggest you to use Heroku (how to configure Heroku and how to upload your application is explained during the course).

Database

Your web site must be data-based driven. For the Database implementation any relational database can be used. We suggest you to use PostgreSQL as shown during past lectures. The DB implementation **WILL NOT BE EVALUATED.**

Shared repository (Github)

Every project must host its code on a **private GitHub repository** and provide a running application on cloud hosting platform. **Course teachers and tutors must be invited as viewers of the repository.**

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

Taking acquaintance with Git

We urge you to take acquaintance with git by reading thoroughly the following links:

- Learn Git with Bitbucket Cloud
- Beginner / What is version control
- Beginner / What is git
- Collaborating / Syncing. Learn the basics of git remote, fetch, pull and push.
- Collaborating / Centralized workflow. Among the available ones, we suggest this workflow for your team as it is very simple. The suggested centralized workflow dictates every developer to have his own local copy of the entire project. To work locally on the project, you could use either the command line git (every OS has its own way of installing it), or a graphical client (GitHub).

What should I commit over to git?

When working with a Node.js based project, a common mistake done by everyone is to include in the repository the node_modules or dist directory. **These directories should not be stored in git!** This post should clarify how to avoid this mistake.

What to deliver

By the chosen exam date, you must make all your implementation work available by delivering a **.txt file** containing:

1. the **link to your website**
2. the **link to your GitHub repository**

PLEASE INCLUDE THESE LINKS ALSO THE COVER PAGE OF YOUR DESIGN PROJECT DOCUMENTATION.

The GitHub repository should contain both your Client and Server code. (**same folder** / **two separated folders**)
The GitHub repository should contain also a **README file** providing at least this information:

- Group Name
- For each Group member: name, surname, id number and email.
- For each group member: a short description of his/her contribution. (to let us know who did what)
- A technical documentation describing:
  - Server and DB technologies used (just few lines)
  - **Components developed**, their functionalities and structure
  - Plugins (if used) e.g., routing, store, …
- Comments about how your usage of the framework was compliant to the best practices for the specific application domain of your website (discussed during the course).

The .TXT file must be posted on a delivery folder in Beep. You will have time until 23:59 of the exam day to upload the file.

Evaluation

The evaluation of the implementation part of the exam for each student is scored in **30 points**; this score accounts for **60%** of the overall score of the project course.
Grades will be assigned considering the following aspects:

- **Technical correctness and completeness of pages implementation**
- **Components Structure and reusability**: an application in Vue/Nuxt can be seen as a set of components. We will consider the structure and organization of the components used.
- **SEO (Search Engine Optimization):** the client side frameworks presented during the course allow you to do both Client Side Rendering and Server Side Rendering. During the course we have discussed the pros and cons of each approach. We will evaluate the implementation choices made by the teams and how they contribute to SEO optimization.
- **Adherence to Vue/Nuxt best practices:** we will acknowledge those who will use in the most appropriate way the features of the framework used.
- **Responsiveness**: the degree at which the interface adapts to different end user devices
- **Accessibility**: the degree at which your web site satisfies the standard accessibility requirements.
- **CONSISTENCY WITH DESIGN SPECIFICATIONS (IDM schemas and concrete page design).**