NumericALL

Final project and presentation

Aim

The end-of-training project is an opportunity to put into practice the maximum number of topics seen during the training. It is also a chance to experiment with collaborating within a team of developers.

Team size and composition

Teams are composed of 3 to 5 people. The teams are chosen by the learners, unless the trainer and training organisers deems it useful to compose them himself.

Types of projects and trainer validation

An overall theme and /or a project brief will be given by the training organisation. Each team must think about the project they wish to develop in accordance with the theme/brief, while keeping in mind the following recommendations:

- The project must use notions of back end and front end development
- It must include use of a database
- The front end part must be integrated using techniques of Responsive Web Design
- The PHP code must be organised according to MVC architecture using the Laravel framework
- The PHP code must be object oriented
- The JavaScript code must use at least one external API
- AJAX must be used in both the JavaScript code the PHP code
- The HTML / CSS code must be valid and accessible
- The project must put into practice some notions related to security, such as password hashing or measures to avoid "SQL injection"-type attacks.

Submission and validation of a project proposal

Apart from these recommendations, the projects can relate to any topic of interest to the team (see "Examples of projects"). Once the project idea has been found, the learners pass it on to the trainer with the following details:

- The name of the project (if it has already been chosen)
- The purpose of the website or web application they wish to create
- The main features
- Recommendations that the team commits to put into practice
- Team members and outline of the division of roles

On the basis of this information, the trainer will validate the project, propose modifications or strongly discourage it if it deviates too much from the recommendations and the general spirit of the projects as described in this document. In the event that the team can not find a project, for lack of ideas or lack of consensus, the trainer will make available to the latter the specifications of a predefined project.

Functional scope of the project

The complete list of features that the team wants to integrate into the project represents the functional scope. This must be achievable in the time available.

It is very important not to overload the project with features that the team will not have time to develop. Here is a method that can help you determine a list of achievable features:

1. General brainstorming so that all members of the team can offer features
2. For each feature, appoint a developer who will be responsible for that feature
3. Each developer makes an estimate of the time it will take to develop each of the functionalities for which he is responsible (see "Estimated time required for each feature")
4. Thanks to the estimated time available, the team is able to retain only those features that can realistically be developed.

Estimated time available
The time available corresponds to the number of hours that each developer can devote to the development of the project. It is determined by adding:

- class time in the presence of the trainer
- the hours that everyone is ready to put into the project in the evening (may be variable from
- one day to another depending on each person's availability)
- hours on holidays or weekends (can be variable from one day to another depending on the each person's availability)

Important: all available time should not be spent developing features. In general, it takes ⅘ of the time available for the development of features and 1/5 for:

- unexpected details
- debugging
- small unplanned but essential features
- preparation of the project presentation

Estimated time needed for each feature : to estimate the time required to complete a feature, it may be useful to split the feature into subparts. For example, for a "Registration form with user account creation" feature:

- Reflection on the list of fields to be included on the form
- HTML / CSS integration of the form
- Validation of fields in JavaScript and error handling
- Validation of fields in PHP and error handling
- Password hash
- Storing information in the database
- Sending an email to validate the email address
- Sending a confirmation email
- Tests and evaluation of the accessibility of the form
- Verification of the validity of the HTML / CSS code
- Function tests by a member of the team who did not contribute to the development of the form

Each subpart is estimated individually and the sum is used to determine the time of the feature. **Very important:** when we think we have determined the time required for the development of a feature, we multiply it by 2! It is far better to overestimate and have additional time to develop new features rather than underestimate and have to rush the project and skip features that had originally been planned.

Examples of projects
Here are some projects done by learners:

- Job offers website allowing the connection between candidates and companies
- Website linking tourists and local artisans
- Platform inspired by Stackoverflow that allows bootcamp learners to value the help they could bring to other learners
- Website of an online library, with catalogue, delivery in relay point and management of late payments
- Clone of Twitter
- Commercialization of projects and intellectual property
- Travel agency website
- Website of a jewellery shop
- A learning management site
- A tourist photograph catalogue site
- Natural disaster prevention app
- Logistics management app

Versioning

All projects must use Git for their versioning. Each team is free to choose the platform they prefer (GitHub, Bitbucket, etc.). Within each team, each developer is free to use the Git client that they prefer, be it on the command line or with a GUI (GitHub, GitKracken, SourceTree, etc.). Each team must give access to the repository (at least in reading) to the trainer.

NumericALL

Defence: Final presentation

On the last day of the training, the teams will present their project to a jury. The jury will consist of at least one member of the teaching team, a member of the NumericALL management and one or more members external to the school. The other learners in the class will also be present.
These presentations will last 45 minutes per team. The speaking time should be divided equally between each member of the team.

We suggest that you run your presentation in the following order:

1. Project description (with or without slides)
2. Presentation of the different members of the team and their respective roles
3. Presentation of the collective production stages, the tools and technologies used, justify the technical choices
4. Demonstration
5. Difficulties encountered during the project
6. Possible future prospects of the project after the training
7. Questions / Answers with the public

Here is the list of the main skills evaluated:

Overall quality of the project:

- Structuring the project: tree of files and pages
- Design and set up of a database
- Develop a user interface
- Set up and respect a teamwork process (versioning, distribution of roles, internal tests, etc.)
- Respect the recommendations in terms of web quality
- Implement good security practises
- Individual contribution and presentation:
- Presentation of individual production
- Quality of individual contribution
- Quality of the presentation of the work done

The assessment covers a common part (75% of the mark) and an individual part (25% of the mark).