WEB LANGUAGES AND TECHNOLOGIES

LTW :: 2020/21 PROJECT

LTW PROJECT

Project description for the 2020 edition of the Web Languages and Technologies course.

OBJECTIVE

Create a website where **users** can list rescue **pets** for **adoption** and/or offer them a *forever home* . To create this application, students should:

- Create a **sqlite** database where information about *users* and *pets* is stored.
- Create documents using **HTML** and **CSS** that represent the web pages of the application.
- Use **PHP** to generate those web pages after retrieving/changing data from the database.
- Use **Javascript** to enhance the user experience (for example using Ajax).

WORK GROUPS

- This project will be completed by students in groups of **four** (*except in classes where the number of students is not a multiple of four*).
- Groups should be composed by **two** students having an **even** number, and **two** with an **odd** number (*if not possible, then at least one student should have an even number and one an odd number*).

- Students should contact their practical class teachers, using *Slack*, to establish these work groups.
- The list of groups will be available here.

REQUIREMENTS

The **minimum** expected requirements are the following:

- **All users** should be able to (users can simultaneously be looking for a pet, or have a pet for a adoption):
  - **Register** a new account.
  - **Login** and **logout**.
  - **Edit** their **profile** (username and password at least).
- **Users** that found a pet and are looking for someone to adopt it should be able to:
  - **Add** information about the pet. Including name (if any), species (*e.g.,* dog, cat), size, color, photos, location, ...
  - **Manage** previous postings.
  - **List** any questions, inquiries, and adoption proposals.
  - **Accept** or **refuse** adoption proposals.
- **Users** looking for a pet should be able to:
  - **Search** for a pet using several search criteria.
  - **Add** pets to a favourites list.
  - **Ask** questions about a pet listed for adoption.
  - **Propose** to adopt a pet and list previous proposals.

Students should also **make sure** that:

- The following technologies are all used: **HTML**, **CSS**, PHP, **Javascript**, **Ajax/JSON**, **PDO/SQL** (using **sqlite**).

- The web site should be as **secure** as possible.
- Code should be **organized** and **consistent**.
- Frameworks are **not** allowed.
- Small helper libraries (e.g. displaying a gallery of pictures) might be allowed (talk with your practical class teacher).

Some suggested **extra** requirements. Extra requirements are a way of making sure each project is unique. You do not have to implement all of these:

- Animal **shelters** should also be able to register as users.
- **Shelters** have a dedicated page with all dogs available for adoption.
- **Users** can be collaborators of a certain shelter and have permission to edit information about the shelter and any pets for adoption.
- **Create** a more robust workflow for pet adoption.
- **Pets** can be in several states (being prepared for adoption, prepared, proposal accepted, delivered, ...).
- **Think** about how pets can move from one state to another (maybe using a state diagram).
- **Users** that adopt a pet should be able to still **post photos** of that animal **after** the **adoption**.
- Users should receive a **notification** anytime something important happens (*e.g.*, a new pet matching a saved query, a new adoption proposal, ...).
- Develop a **REST API** that allows bots or other apps to use the website.
- **Anything else** you can think of...

WORK PLAN

This is a proposed plan to guide your work. **No deliverables are expected** or will be evaluated at these dates.

- **Week 7:** - **Mockups** and **navigation diagrams** for the main pages, first draft of the **database design**.
- **Week 8:** - Finalize **database script** and **create database**. Partial implementation of **some main pages** and first **CSS** version.
- **Week 9:** - All **main pages** implemented. Start working on **secondary features**.
- **Week 10:** - Continue working on **secondary features**. Start working on **Javascript** and **Ajax**.
- **Week 11:** - Finish **secondary features**. Main focus on **security** aspects.
- **Week 12:** - REST **API** or other **secondary features**. **Testing** and **code cleanup**.
- **Week 13:** - **Delivery** and **presentation**.

We recommend that students adopt an agile methodology. Don't start by planning every little detail right from the start as you run the risk of ending up with a great plan but a poor implementation; but be aware of code organization and quality from the beginning.

EVALUATION

Evaluation will be done on the following topics:

- **Complexity** (implemented features)
- **Security** (XSS, CSRF, injection, password storage, ...)
- **Technology** (correct usage of HTML, CSS, Javascript, Ajax, No frameworks, ...)
- **Quality** (code quality, file organization, consistency, ...)
- **User Interface** (usability, consistency, ...)

DELIVERY

- **Presentation**: On the last practical class (18th of December).

- Instructions