Recipients

This work is intended for all students enrolled in the curricular unit of Programming in a Web Environment under Evaluation during the academic period who intend to pass the curricular unit.

Objectives

With the completion of practical work, it is intended that students put into practice all the knowledge acquired in the Curricular Unit, demonstrating their skills in:

- Know and apply the Client-Server model and the main associated technologies/protocols for the development of Web applications;
- Understand and implement the MVC pattern;
- Implement Web applications through the manipulation of languages and frameworks for the development of client and server components in Web applications;
- Develop a Web application using the languages, libraries and frameworks taught;
- Understand application development as a fullstack developer.

Statement

The social valorisation company "Recicla Têxtil", in partnership with the environmental fund, intends to reinforce the promotion of the recovery and recycling of textile pieces, post-consumption. In this context, the development of a WEB application is requested to speed up the collection of textile pieces that users no longer use for reuse and sending them to social institutions, benefiting institutions. This platform will be free to use by users, who receive points convertible into discount vouchers in stores that sponsor this idea.
In practical terms, the application should include a *Backoffice*, which should be restricted to Administrators/employees and should include all the management features of the donation platform and management of delivery points by the employees of the company "Recicla Têxtil". In a future phase, there will be a *Frontoffice* where users can register postal shipments of textile items and check their stitch status. The *frontoffice* can also be used by social institutions to register on the platform and verify the donations received.
The collection of textiles can be carried out in person at a collection point or by post, with validation after receipt by post at the company "Recicla Têxtil" by company employees, validating the donation (e.g. number of pieces and weight) issuing points to the donor user and adding photographic proof of the donation received.
Thus, the platform must respond to 4 types of users, providing differentiated features according to the profile:

- Platform Administrators/Employees
    - Management of all users (Administrators/Employees, Beneficiaries and Donors);
    - Manage the rules for awarding points for donations. For each donation, points must be awarded to Donors, which can be for example for the number of Kgs donated and points to be made available to fundraisers, which can be according to the number of fundraisers received and delivered;
    - Management of donor collection requests, with the respective workflow from request to delivery at the collection point;

    - Creation of a dashboard, contemplating several metrics such as: average kgs per donation, places with more/less donations, etc.
- Beneficiary entities
    - They must be registered on the platform, and their request must be accepted or entered by an administrator/employee. Admins should receive a notification indicating that there are new entities to validate (this can be by email, or an indicator on the dashboard);
    - Whenever there is a request for donation by a Donor, the donation and its status (e.g. received, delivered, lost) must be registered with the beneficiary company;
- Donors
    - They must be registered on the platform and will be accepted automatically;
    - The Donor can make a request for a textile donation, selecting the entity to which he wishes to make the donation, filling out a form where he must indicate the pieces to be donated and their condition. According to each piece and its condition, you immediately have an estimate of the number of points you will receive;
    - Each donation request will entitle you to x points (according to the rules defined on the platform) and the user must have access to a dashboard with at least one KPI with the total number of points, another KPI with the total number of donations and a graph with the evolution of the number of points over time.

The web application should allow the management of all the information necessary for the platform, allowing visitors to search for the entities benefiting from the donations.

The points obtained by donors must be able to be converted into discount vouchers automatically issued on the configurable platform in the *Backoffice* for users (Donors)

To bonus the developed platform, you should consider adding requirements and features that complete this problem, making it richer in features:

- Enable direct donations in euros using an integration with an external payment API (e.g. Paypal),
- Allowing users to also play the role of fundraisers who collect donations from other users and deliver them to a collection point, sharing the points with the original users. (Provision must be made for the scheduling of collection and the sharing of points in the system);
- Integrate with a system for sending emails for notifications (e.g. Mailgun - https://www.mailgun.com/);

You should discuss the implementation of additional features or clarify doubts about the statement with the respective teachers of the practical classes.

Milestone #1

For the development of the first *milestone* of the work, the specification and elaboration of the *BackOffice* platform for this solution must be considered, at least.

- Registration of users (Administrators, Beneficiaries and Donors);
- Management of Benefiting Entities (each entity must have a profile, including name, description, images, among other relevant data);
- Management of donation requests (In this first milestone it can be assumed that the user physically delivers his donation at one of the company's collection points and an employee registers the donation, the user who made the donation and the points accumulated by a user);

The *Backoffice* is only to be used by the Administrators/Employees of the platform, the other users must use the platform through the *frontoffice* that will be developed in Milestone 2.

They must at least **implement the functionalities** for registration and management of beneficiary entities and the management of donation requests **to obtain a minimum grade** (7.5) in this *milestone*.

The development must be done using the *ExpressJS* **framework** for node.js using the *EJS* **template engine**. Database persistence should be ensured by using a MongoDB database for this purpose.

In addition, the presentation of additional work that is not considered mandatory at this stage will be valued, such as, for example, *dashboard* with indicators and graphs, definition of a rule for the attribution of points per kg and per collection, user blocking, user authentication.

The first *milestone* will not have a presentation, however the content delivered by the students will be evaluated and feedback will be provided during the practical classes with the result of the specific evaluation of each group.

Observations:

The first *milestone* should assume the realization of a *backoffice* project, that is, it should only be used by the company's employees. The pages created in this *milestone* should only be visible by employees of the company "Recicla Têxtil" who record all the information. Automating the process and *front-office* so that users can interact with the platform remotely will be done in *milestone* #2.

Milestone #2

The second *milestone* considers the delivery of a set of web resources that **allow a complete response** to the statement. In this case, all the work will be evaluated, including the modifications and recommendations proposed by the teachers after *milestone* #1. The quality of the app and the features presented will be evaluated.

The second *milestone* should also be based on the implementation of the *frontoffice* using the Angular platform, using web services (REST API) for the management of the customer's page, surveys and management of donations. The user should be allowed to schedule donation shipments (e.g. by post) and check the status of donations and points awarded.

On the other hand, it should be possible to register beneficiary entities, with their own login and the verification of all donations received through this platform.

On the server side, it is mandatory to use REST *endpoints* using *nodeJS* and the *ExpressJS framework* to create *webservices* to support the application. All REST services developed must be properly documented using the Swagger platform.

Observations:

The second *milestone* is oriented to the elaboration of the website for the end customer, in this case, the *frontoffice* project for customers who visit the platform (Benefit Entities, Fundraisers and Donors). For this purpose, you should share the same backend project (nodeJS + ExpressJS) developed in the first *milestone*. The *Angular frontend* should be developed from scratch and doesn't need to have the same theme/formatting as the first *milestone*.

Only the necessary functionalities in the backend application *to support the new* frontoffice *application should be added and/or modified*, which should include a REST API, swagger and other methods that are relevant to the development of the application. It is reminded that the customer portal should at least allow the customer to:

- Consult all beneficiary entities, with the possibility of filtering according to various parameters (location, typology, ...)
- Check the history of donations, and the respective current account of points earned
- Make donation requests, accept donation requests, as well as your entire workflow according to the user profile, including sending notifications

- Request the conversion of points into discount vouchers

The content of the first milestone should not be redone and/or updated to the angular *framework*. Only the *frontoffice application* of the second *milestone* suffers the penalty for not using the Angular *framework*. The *backoffice application* of the first *milestone* should continue in the same format (ExpressJS + EJS).

Tools

In the development of practical work, working groups should use:

- NodeJS and the ExpressJS framework
- Angular
- Development IDE (eg VSCode)
- Git and Gitlab (gitlab.estg.ipp.pt)

Final Report

The work must be accompanied by a project report, having the following topics:

- Identification and characterization of the project, justifying the approach followed according to the company's business process. A development diagram must also be presented, identifying the main tasks, the member of the group responsible for this task and the expected time;
- General specification of the software to be developed, mapping the business requirements with the software components to be developed. You can use UML mockups and/or diagrams to justify the decisions made at the implementation level for the most relevant decisions.
- Analysis of the main points of the work. Technical discussion on development options and their assessment (a critical assessment of the initially established plan can be carried out).

Evaluation

Weighting of components:

- 40% first milestone (minimum grade 7.5 points)
- 60% second milestone (minimum grade 7.5 values)

In each of the components, a total of up to 3 values will be reserved for bonus features or additional requirements. A critical spirit and the implementation of additional features and/or bonuses are encouraged, which must be coordinated with the curricular unit's professors.

Delivery Format

The submitted works should avoid (if possible) the use of absolute paths or specific addresses, so that they can be easily used on any machine. In addition, and in order to facilitate the reception of the various works received, they must comply with the following rules:

- All elements of the group must submit the work in the respective link;
- When submitting projects, the node_modules folder must be omitted;
- The work developed must be delivered through moodle, by submitting a ZIP file with the name PAW_<number_of_group>_<number_of_student>_<number_of_student2>_<number_of_student3>.zip.

Dates and Considerations

Students must notify their workgroup on the Moodle platform in due time, by April 3rd. Groups must consist of a maximum of 3 members.

The work must be delivered by:

- at 23:55 **(Moodle time) on the 3rd of May** in the first milestone
- at 23:55 **(Moodle time) on the 7th of June** in the second milestone

Submissions must be made through the curricular unit's page at http://moodle.estg.ipp.pt.

The defense of the work will be at the respective time for each class and only for the delivery of the second milestone. The students involved in the work must prepare the presentation in order to demonstrate the work developed in about 15 minutes. The presentation can be performed on the student's personal computer or on the computer available in the exam room. In both cases, the student must prepare all the necessary content so that he can demonstrate the developed functionalities.

A **satisfactory defense** is considered when the student proves that he completed the work submitted and that he **masters all the concepts** applied in the resolution of the work. Fraud attempts will result in the evaluation of the work as: Academic Fraud.