

A10: Product

Presto is an online platform where people can ask questions and contribute with answers, developing a community of unique insights on a wide variety of topics.

Members can post questions associated with one or more topics, which other people may answer. Any member can reflect their opinion on the application's content via commentaries (where it is possible to mention fellow members), ratings or both. Members are notified of activity that can be of their interest, such as questions from people they follow, upvotes on their content and mentions by other people. There is an advanced search feature where users can search for almost anything, with powerful filters and sorting options to better suit what one seeks. Each member has a score associated, built on his contributions to the community and others' opinion on it.

In this artifact, all the features developed will be described, as well as some implementation details about the final product.

1. Installation

Final product source code is available in the A10 release at <https://github.com/antonioalmeida/lbaw1725/releases/tag/v1.0>

Docker command to run image:

```
docker run -it -p 8000:80 -e DB_DATABASE=lbaw1725 -e DB_USERNAME=lbaw1725 -e DB_PASSWORD=nk77sp08 diogotorres97/lbaw1725
```

To create the image:

1. composer install
2. npm install
3. npm run prod
4. php artisan clear-compiled
5. php artisan optimize
6. docker build -t diogotorres97/lbaw1725

2. Usage

The product is available at <http://lbaw1725.lbaw-prod.fe.up.pt>

2.1. Administration Credentials

Administration login: <http://lbaw1725.lbaw-prod.fe.up.pt/admin/login>

Email	Password
-------	----------

admin@presto.com	randompassword@99
------------------	-------------------

2.2. User Credentials

Type	Email	Password
Regular member	antonioalmeida@gmail.com	randompassword@99
Moderator	diogo.rey@gmail.com	piroxenas@99

The other accounts (whose email can be checked in the SQL script) have the password `qwerty@99`

3. Application Help

In order to provide a better user experience, some help tips are present throughout the application. In any page, it is possible to access the About section (in the right corner of the footer), where users can learn about the project's goals, its main features and the development team. Another relevant examples is the sign up form, where users have a tooltip on each form field explaining what is expected in each field.

4. Input Validation

Input data validation was applied both on the client end and the server end. In the client side, validation was done using the **type**, **required** and **pattern** attributes for input elements, to, respectively, indicate what type of input is expected, its mandatory status in the form and possible regular expressions that define the set of accepted values (e.g. in the case of a member's username, the input type is text, but it was defined that usernames must be 2-20 characters long with alphanumeric characters, hyphens and underscores). Two examples of this type of validation are in the register form and the edit profile form.

As for server-side validation, Laravel's inbuilt validation facilities were used (i.e. request validation). Through this, request fields can easily be tested against a set of constraints. In this context, input was validated mostly through parameters like **required**, **type** (in reality, the type was specifically indicated, e.g. **string**, **url**). However, sometimes, due to some limitations in the specificity of some rules, custom ones were defined. This type of validation is done on almost all requests, however some examples where all of what was mentioned is applied can be seen on the Profile (*update* method) and Topic (*update* method) controller files. Other relevant example is the *validateLogin* method in the Login controller file, where a custom rule is applied to the credentials to check if the member is banned and, if so, disallow their authentication.

5. Check Accessibility and Usability

The PDFs for the accessibility and usability checklists are provided with the artifact.

Small note about an **accessibility** checklist element:

- Rule 17, which the group doesn't consider to have completed, states "The website is readable and navigable with CSS deactivated", does not make much sense in today's era, in the group's opinion. As web pages have become more and more complex, it is increasingly difficult to produce good readable, navigable pages without the help of proper styling, and the necessity for readable, navigable complex pages with no CSS at all is minimal.

6. HTML & CSS Validation

The pages validated in the **HTML validator** were the main feed (index) and the profile page. The validation results are sent in the respective PDF files. Notes about the validation:

- In some browsers, it might be possible that the view-source: directive does not show the full page's HTML (e.g. in Firefox it apparently does not). As such, to extract the source HTML for validation, one must go to the inspector and proceed to copy the outer HTML of the <html> element
- The infinite-scroll-disabled attribute is used for the infinite scroll mechanism. Not being a HTML standard attribute is what probably made it be recognised as an error
- All the warnings are of the same type: "type attribute not necessary for style elements" - As the page was extracted directly from the browser inspector, some elements were inserted automatically, like these. Therefore, the warning is unavoidable.

As for the **CSS Validator**, the generated public app.css file was submitted to the tool (PDF also sent with the artifact). Notes about the validation:

- Warning "Unknown vendor extension..." - Extensions are automatically used by Bootstrap to ensure backwards compatibility, it is something that the group believes they could not control

7. Revisions to the Project

7.1 Database

Main changes to the database's tables since its specification:

- Table **country** was removed, as the information was not being used for anything and was deemed unimportant.
- Table **notification** was removed and replaced by a notifications table created via migration, so as to better integrate our notifications with the notification class already provided by Laravel.

- A **password_reset** table was added, via migrations, to implement the password recovery functionality.
- Some fields were added to the **member** table so as to provide a better integration with the Laravel framework (remember_token, provider, provider_id).
- A new field was added to the **answer** table to know if that answer is the answer chosen as correct for its question.
- All three rating tables (**question_rating**, **answer_rating**, **comment_rating**) have a new field, deleted_at, so as to be able to implement soft deletes in the rating functionality, providing a better UX.

Furthermore, triggers and stored procedures were updated accordingly. Main changes include the removal of notification triggers (the insertion is done in a different way) and addition of chosen_answer integrity triggers (guarantee chosen answer is unique, disassociate chosen answer when question is reopened).

7.2 Web Resources

The JSON answers specified in the A7 artifact were replaced by the use of Laravel Resources. Some routes had their name changed and others were added due to the exchange of blades for Vue components.

7.2.1. M01: Authentication and Static Pages

To complete the newly added user story Password Recovery, the following resources were added so that the user can reset their password after receiving the respective recovery email.

VIEW Send Email Reset password Page password/reset
POST Send Email Reset password Request password/email
VIEW Reset password Page password/reset/{token}
POST Reset password Request password/reset

To complete the user story Log in/Sign up with external API, the following resources were added so that the user can login/sign up with their social media account.

GET Redirect to Provider API External auth/{provider}
GET Redirect to Provider Callback API External auth/{provider}/callback

Finally, a route was added to check if there is currently a logged in user in the current session in Presto

GET Get isLoggedIn /isLoggedIn

7.2.2. M02: Members

For simplification, **R203** and **R209** was deleted and replaced by:

GET Get Member logged in profile/
R212 and R213 deleted because it's not used.

In **R215** and **R204** the parameter member is useless so the URLs were simplified:

GET Get Notifications API /api/notifications

GET Edit Profile Action /api/profile

R216 was deleted and replaced by a middleware in laravel that handles the marking a notification as read mechanism

New resources:

PATCH Edit Profile Pic Action api/member/edit-profile-pic

GET Get Member Answers api/profile/{member}/answers

GET Get Member Questions api/profile/{member}/questions

GET Get Member Followers api/profile/{member}/followers

GET Get Member Following api/profile/{member}/following

GET Get Member Notifications Stats api/notificationsStats

GET Get Member Unread Notifications api/UnreadNotifications

7.2.3. M03: Content (Questions, Answers and Comments)

R306: Close/open Question API was replaced by the following routes:

POST Solve question /api/questions/{question}/solve

POST Unsolve question /api/questions/{question}/unsolve

R303 changed to an API Resource so the prefix was added :

R303: POST Add Question api/questions

R317: Add Comment API was replaced by the following routes:

POST Store question comment /api/comments/question/{question}/

POST Store answer comment /api/comments/answer/{answer}/

New resources:

GET Get Question Answers api/questions/{question}/answers

7.2.4. M04: Topics

In all resources, the prefix topics/ was changed to topic/.

Resources **R403: Add Topic API**, **R405: Get Edit Topic Form API** and **R407: Delete Topic Action** were removed

Resource **406: Edit Topic Action** was replaced by

POST Edit Topic api/topic/{topic}

New resources:

PATCH Edit topic picture api/topic/{topic}/edit-pic

GET Get all topics api/topic

7.2.5. M05: Index and Search

R501 and **R502** URLs were replaced by only '/'

R503: Get Feed API was replaced by the following resources to retrieve the content displayed on the page:

GET Get Most Recent Content /api/feed/getNewContent

GET Get Top Content /api/feed/getTopContent

GET Get Recommended Content /api/feed/getRecommendedContent

GET Get trending topics /api/feed/getTrendingTopics

GET Get top members /api/feed/getTopMembers

R504: Search Page and **R505: Search API** were updated to the resource search/{query} (**VIEW** and **GET**, respectively)

7.2.6. M06: Administration

The following resources were added to handle admin login and logout actions:

POST Admin login /admin/login

VIEW Admin login Form /admin/login

GET Admin logout /admin/logout

The the following resources were added to retrieve the users displayed on the admin page:

GET Get users /api/admin/get-users (replaced **R602: Search for Members API**)

GET Get banned users /api/admin/get-banned

GET Get flagged users /api/admin/get-flagged

GET Get moderators /api/admin/get-moderators

R604: Promote Member API and **R605: Dismiss Moderator API** were combined and replaced by

POST Toggle moderator status api/members/{username}/toggle-moderator

The resource

DELETE Dismiss user flag api/flags/{member_id}/{moderator_id}/dismiss

was added to dismiss (i.e. to delete) a flag.

7.3 Other Requirements

- Facebook log in/sign up system wasn't implemented because since March 2018 the API only allows apps with HTTPS.
- Business rule BR03 (Edit time limit for non-moderator members) was not enforced, as it was deemed irrelevant

8. Implementation Details

8.1. Libraries Used

Library/framework	Reference	Usage description	Usage example
Laravel Socialite	https://laravel.com/docs/5.5/socialite	Authentication through OAuth provider	https://lbaw1725.lbaw-prod.fe.up.pt/signup - Option to sign up through Google API
Pusher	https://pusher.com/	Implementation of in real time notifications	https://lbaw1725.lbaw-prod.fe.up.pt (or any page, when logged in) - Notifications are sent and show in real time to the user (new/unread notifications available through the bell icon in the navbar)
Axios	https://github.com/axios/axios	Make AJAX requests	https://lbaw1725.lbaw-prod.fe.up.pt - Obtain information to display in main feed (actually, all pages that perform requests use it)
Laravel Vue Pagination	https://github.com/gilbitron/laravel-vue-pagination	Pagination of request results using Vue components	https://lbaw1725.lbaw-prod.fe.up.pt/admin - Listing platform members in administration dashboard
Tribute.js	https://github.com/zurb/tribute	Implementation of @mentions suggestions on comments	https://lbaw1725.lbaw-prod.fe.up.pt/questions/18 (while logged in; can be any question that is not solved) - Mentioning fellow members in comments (either to the question or any answer)
Vue.js	https://vuejs.org/	Build interactive web interfaces through reactive data binding and composable view components.	https://lbaw1725.lbaw-prod.fe.up.pt (or any other page in the whole product) - Each page is defined as a Vue component
Vue infinite scroll	https://github.com/ElementFE/vue-infinite-scroll	Implementation of infinite scroll on feeds	https://lbaw1725.lbaw-prod.fe.up.pt - When reaching the end of the page, more content is automatically loaded

TinyMCE	https://www.tinymce.com/	Answer questions in a more detailed and visually appealing way	https://lbaw1725.lbaw-prod.fe.up.pt/questions/18 (while logged in; can be any question that is not solved) - Answers are written in the WYSIWYG editor TinyMCE, providing a good interface to write more styled answers according to what the member seeks
---------	---	--	---

8.2 User Stories

User stories bolded and marked with an asterisk (*) were added after the requirements specification stage.

Identifier	Name	Priority	Team members	State
US11	About page	high	António Almeida, Diogo Torres	100%
US12	Search	high	António Almeida, João Damas, Bruno Piedade	100%
US13	Filter	high	António Almeida, João Damas	100%
US14	Sort	high	António Almeida, João Damas	100%
US15	Inline Help	high	António Almeida, João Damas	100%
US21	Log in	high	Diogo Torres	100%
US22	Sign up	high	Diogo Torres	100%
US23	Sign up using external API	low	Diogo Torres	100%
US24	Log in using external API	low	Diogo Torres	100%
US31	Question	high	António Almeida, Diogo Torres, João Damas	100%
US32	Answer	high	António Almeida, Diogo Torres, João Damas	100%
US33	Comment	high	António Almeida, João Damas	100%
US34	Rate	high	Diogo Torres, João Damas	100%

US35	Notification	high	António Almeida, Bruno Piedade, Diogo Torres	100%
US36	Main feed	high	António Almeida, Diogo Torres, João Damas, Bruno Piedade	100%
US37	Follow	high	António Almeida, Diogo Torres	100%
US38	Unfollow	high	António Almeida, Diogo Torres	100%
US39	Score	high	João Damas	100%
US310	Log out	high	Diogo Torres	100%
US311	Follow user	high	António Almeida, Diogo Torres	100%
US312	Unfollow user	high	António Almeida, Diogo Torres	100%
US313	Statistics	medium	Diogo Torres	100%
US314	Report	medium	António Almeida	100%
US315	Block user	medium	-	0%
US316	Mention user	medium	João Damas	100%
US317	Share content	low	-	0%
US318	Message user	low	-	0%
US319	Suggestions	low	-	0%
US320*	Recover password	high	Diogo Torres	100%
US41	Remove content	high	António Almeida	100%
US42	Edit content	high	António Almeida, João Damas, Diogo Torres	100%
US43	Profile	high	António Almeida, Bruno Piedade, Diogo Torres	100%
US44	Close question	high	António Almeida, João Damas	100%
US45	Reopen question	high	António Almeida, João Damas	100%

US46*	Choose correct answer	high	João Damas	100%
US51	Remove content	high	António Almeida	100%
US52	Edit content	high	António Almeida, João Damas	100%
US53	Request ban	high	João Damas	100%
US54	Add certified badge	medium	-	0%
US55	Remove certified badge	medium	-	0%
US56*	Edit topic	high	António Almeida, João Damas	100%
US61	Ban user	high	Bruno Piedade	100%
US62	Appoint Moderator	high	Bruno Piedade	100%
US63	Dismiss Moderator	high	Bruno Piedade	100%

GROUP1725, 30/05/2018

António Cunha Seco Fernandes de Almeida, up201505836@fe.up.pt
Bruno Manuel Nascimento Costa Galvinas Piedade, up201505668@fe.up.pt
Diogo Luis Rey Torres, up201506428@fe.up.pt
João Paulo Madureira Damas, up201504088@fe.up.pt