Architecture Specification and Prototype

Last edited by **Mariana Marques Maio Pereira** 5 months ago

EAP: Architecture Specification and Prototype

askFEUP aims to create a collaborative online platform where FEUP students can ask questions, share experiences, and connect on academic and social topics. Our goal is to build a user-friendly, mobile-responsive space where students can seek advice, offer help, and engage with peers in a supportive community.

The platform will feature categorized questions, personalized content via a "For You" page, and interactive tools like upvoting and leaderboards to encourage participation. With clear roles for administrators, moderators, and users, askFEUP will foster a respectful, well-moderated environment, becoming a central hub for student engagement and collaboration.

A7: Web Resources Specification

This document outlines the web API to be developed, highlighting the required resources, their properties, and the corresponding JSON responses. The API will support operations for creating, reading, updating, and deleting (where applicable) for each resource.

1. Overview

This section presents an overview of the web application, where the modules are identified and briefly described.

Module	Description
M01: Authentication	Web resources associated with user authentication and registration, providing features such as login, logout, and registration.
M02: Users	Web resources related to user interaction, including features such as viewing user information (home page, profile page, editing profile details) and managing notifications.
M03: Posts	Web resources centered around posts, including creating, editing, deleting, and viewing posts (questions, answers, and comments).
M04: Administration	Web resources for administrative actions, such as enforcing terms of service, blocking/unblocking and banning users, deleting posts/comments, and updating static pages.

2. Permissions

This section defines the permissions used by the modules to establish the conditions of access to data and features. Only the permissions relevant to the Vertical Prototype component are outlined.

Identifier	Name	Description
USR	User	Anyone that can interact with public information of the system
GST	Guest	User who can login or register to the system
AUT	Authenticated User	User that has logged in the system
QA	Question Author	Authenticated user that posted a question
AA	Answer Author	Authenticated user that posted an answer to a question
ADM	Admin	Authenticated user who has administrator privileges

3. OpenAPI Specification

This section presents the OpenAPI specification in YAML format for the askFEUP system. It describes the Vertical Prototype's web resources.

Link to the `a7_openapi.yaml` file

```
openapi: 3.0.0

info:
  version: '1.0'
  title: 'askFEUP API'
  description: 'Web Resources Specification (A7) for askFEUP'

servers:
```

```
- url: http://lbaw.fe.up.pt
  description: Production server

tags:
 - name: 'M01: Authentication'
 - name: 'M02: Users'
 - name: 'M03: Posts'
 - name: 'M04: Administration'

paths:

### M01 Authentication ###
  /login:
    get:
      operationId: R101
      summary: 'R101: Login Form'
      description: 'Provide login form. Access: GST'
      tags:
        - 'M01: Authentication'
      responses:
        '200':
          description: 'Ok. Show Log-in UI'

    post:
      operationId: R102
      summary: 'R102: Login Action'
      description: 'Processes the login form submission. Access: GST'
      tags:
        - 'M01: Authentication'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                email:
                  type: string
                  format: email
                password:
                  type: string
                  format: password
              required:
                  - email
                  - password
      responses:
        '302':
          description: 'Redirect after processing the login credentials.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful authentication. Redirect to home page.'
                  value: '/home'
                302Error:
                  description: 'Unsuccessful authentication. Redirect to login form.'
                  value: '/login'

  /logout:
    get:
      operationId: R103
      summary: 'R103: Logout Action'
      description: 'Logout the current authenticated user. Access: AUT'
      tags:
        - 'M01: Authentication'
    responses:
      '302':
        description: 'Redirect after processing logout.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Successful logout. Redirect to login form.'
                value: '/login'

/register:
  get:
    operationId: R104
    summary: 'R104: Register Form'
    description: 'Provide new user registration form. Access: GST'
    tags:
      - 'M01: Authentication'
    responses:
      '200':
        description: 'Ok. Show Sign-Up UI'
  post:
    operationId: R105
    summary: 'R105: Register Action'
    description: 'Processes the new user registration form submission. Access: GST'
    tags:
     - 'M01: Authentication'
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              name:
                type: string
              tagname:
                type: string
              email:
                type: string
                format: email
              password:
                type: string
                format: password
              age:
                type: integer
              country:
                type: string
              degree:
                type: string
            required:
              - name
              - tagname
              - email
              - password
              - age
    responses:
      '302':
        description: 'Redirect after processing the new user information.'
        headers:
           Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Successful authentication. Redirect to home page.'
```

```
                  value: '/home'
                302Failure:
                  description: 'Failed authentication. Redirect to register form.'
                  value: '/register'

### M02 Users ###

  /profile/{id}:
    get:
      operationId: R201
      summary: 'R201: View User Profile'
      description: 'Displays the profile of a specific user. Access: AUT'
      tags:
        - 'M02: Users'
      parameters:
        - in: path
          name: id
          required: false
          schema:
            type: integer
          description: "The ID of the user whose profile to retrieve. If omitted, returns the profile of the authentica
      responses:
        '200':
          description: 'Ok! Show User Profile UI.'
        '404':
        description: "Profile not found for the given ID."

  /user/edit-profile:
    get:
      operationId: R202
      summary: 'R202: Edit User Profile'
      description: 'Shows the edit profile page of the user. Access: AUT'
      tags:
        - 'M02: Users'
      responses:
        '200':
          description: 'Success! You can now edit. Show edit profile UI.'
        '401':
          description: "Unauthorized. The user must be logged in to access the edit profile page."

  /edit-profile/name/{id}:
    post:
      operationId: R203
      summary: 'R203: Update User Name'
      description: 'Updates the name of a user in their profile. Access: AUT'
      tags:
        - 'M02: Users'
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: integer
          description: "The ID of the user whose name is being updated."
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
            type: object
            properties:
              name:
                type: string
                description: "The updated name of the user."
            required:
              - name
      responses:
        '302':
          description: 'Redirect after processing update name information.'
          headers:
            Location:
              schema:
                type: string
              example:
                302Success:
                  description: 'Successful update. Redirect to edit profile page.'
                  value: '/edit-profile/{id}'
                302Failure:
                  description: 'Failed update. Redirect to edit profile page.'
                  value: '/edit-profile/{id}'
        '400':
          description: 'Validation error.'
        '401':
          description: 'Unauthorized. The user must be logged in to update their name.'
        '404':
          description: 'User not found for the provided ID.'

  /edit-profile/tagname/{id}:
    post:
      operationId: R204
      summary: 'R204: Update User Tagname'
      description: 'Updates the tagname of a user in their profile. Access: AUT'
      tags:
        - 'M02: Users'
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: integer
          description: "The ID of the user whose tagname is being updated."
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
            type: object
            properties:
              tagname:
                type: string
                description: "The updated tagname of the user."
            required:
              - tagname
      responses:
        '302':
          description: 'Redirect after processing update tagname information.'
          headers:
            Location:
              schema:
                type: string
              example:
                302Success:
                  description: 'Successful update. Redirect to edit profile page.'
                  value: 'user/edit-profile'
                302Failure:
                  description: 'Failed update. Redirect to edit profile page.'
                  value: '/profile/edit'
        '400':
          description: 'Validation error.'
        '401':
          description: 'Unauthorized. The user must be logged in to update their tagname.'
        '404':
          description: 'User not found for the provided ID.'
```

```
/edit-profile/email/{id}:
  post:
    operationId: R205
    summary: 'R205: Update User Email'
    description: 'Updates the email of a user in their profile. Access: AUT'
    tags:
      - 'M02: Users'
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
        description: "The ID of the user whose email is being updated."
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
          type: object
          properties:
            old-email:
              type: string
              format: email
              description: "The user's current email address for verification."
            new-email:
              type: string
              format: email
              description: "The new email address to update."
            confirm-email:
              type: string
              format: email
              description: "Confirmation of the new email address, must match `new-email`."
          required:
            - old-email
            - new-email
            - confirm-email
    responses:
      '302':
        description: 'Redirect after processing update email information.'
        headers:
          Location:
            schema:
              type: string
            example:
              302Success:
                description: 'Successful update. Redirect to edit profile page.'
                value: 'user/edit-profile'
              302Failure:
                description: 'Failed update. Redirect to edit profile page.'
                value: '/profile/edit'
      '400':
        description: 'Validation error.'
      '401':
        description: 'Unauthorized request.'
      '404':
        description: 'User not found for the provided ID.'

/edit-profile/password/{id}:
  post:
    operationId: R206
    summary: 'R206: Update user password.'
    description: 'Update the user password. Access: AUT'
    tags:
      - 'M02: Users'
    parameters:
      - name: id
```

```
        in: path
        required: true
        description: ID of the user whose password is updated.
        schema:
          type: integer
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              old-password:
                type: string
                description: User's current password.
              new-password:
                type: string
                description: New password value.
              new-password_confirmation:
                type: string
                description: Confirmation of the new password.
            required:
              - old-password
              - new-password
              - new-password_confirmation
    responses:
      '302':
        description: 'Redirect after processing update password information.'
        headers:
          Location:
            schema:
              type: string
            example:
              302Success:
                description: 'Successful update. Redirect to edit profile page.'
                value: 'user/edit-profile'
              302Failure:
                description: 'Failed update. Redirect to edit profile page.'
                value: '/profile/edit'
      '400':
        description: 'Validation error or incorrect password.'
      '401':
        description: 'Unauthorized request.'
      '404':
        description: 'User not found for the provided ID.'

/edit-profile/age/{id}:
  post:
    operationId: R207
    summary: 'R207: Update user age.'
    description: 'Update the user age. Access: AUT'
    tags:
      - 'M02: Users'
    parameters:
      - name: id
        in: path
        required: true
        description: ID of the user whose age is updated.
        schema:
          type: integer
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              age:
                type: integer
                description: The new age for the user.
            required:
              - age
    responses:
      '302':
        description: 'Redirect after processing update age information.'
        headers:
          Location:
            schema:
              type: string
            example:
              302Success:
                description: 'Successful update. Redirect to edit profile page.'
                value: 'user/edit-profile'
              302Failure:
                description: 'Failed update. Redirect to edit profile page.'
                value: '/profile/edit'
      '400':
        description: 'Validation error.'
      '401':
        description: 'Unauthorized request.'
      '404':
        description: 'User not found for the provided ID.'

/edit-profile/country/{id}:
  post:
    operationId: R208
    summary: 'R208: Update user country.'
    description: 'Update the user country. Access: AUT'
    tags:
      - 'M02: Users'
    parameters:
      - name: id
        in: path
        required: true
        description: ID of the user whose country is updated.
        schema:
          type: integer
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              country:
                type: string
                description: The new country for the user.
            required:
              - country
    responses:
      '302':
        description: 'Redirect after processing update country information.'
        headers:
          Location:
            schema:
              type: string
            example:
              302Success:
                description: 'Successful update. Redirect to edit profile page.'
                value: 'user/edit-profile'
              302Failure:
                description: 'Failed update. Redirect to edit profile page.'
                value: '/profile/edit'
      '400':
          description: 'Validation error.'
        '401':
          description: 'Unauthorized request.'
        '404':
          description: 'User not found for the provided ID.'

  /edit-profile/degree/{id}:
    post:
      operationId: R209
      summary: 'R209: Update user degree.'
      description: 'Update the user degree. Access: AUT'
      tags:
        - 'M02: Users'
      parameters:
        - name: id
          in: path
          required: true
          description: ID of the user whose degree is updated.
          schema:
            type: integer
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                degree:
                  type: string
                  description: The new degree for the user.
              required:
                - degree
      responses:
        '302':
          description: 'Redirect after processing update degree information.'
          headers:
            Location:
              schema:
                type: string
              example:
                302Success:
                  description: 'Successful update. Redirect to edit profile page.'
                  value: 'user/edit-profile'
                302Failure:
                  description: 'Failed update. Redirect to edit profile page.'
                  value: '/profile/edit'
        '400':
          description: 'Validation error.'
        '401':
          description: 'Unauthorized request.'
        '404':
          description: 'User not found for the provided ID.'

### M03 Posts ###

  /:
    get:
      operationId: R301
      summary: 'Redirect to home page'
      description: 'Redirects the url `/` to `/home`. Access: USR'
      tags:
        - 'M03: Posts'
      responses:
        '302':
          description: 'Redirect to the `/home` page.'
          headers:
            Location:
```

```
            description: 'The URL of the redirection target.'
            schema:
              type: string

/home:
  get:
    operationId: R302
    summary: 'R301: View user home page'
    description: 'Show user home page. Access: USR'
    tags:
      - 'M03: Posts'
    responses:
      '200':
        description: 'Success! Show the home page'

/questions/{id}:
  get:
    operationId: R303
    summary: 'R303: View a specific question'
    description: 'Show a question, Access: USR'
    tags:
      - 'M03: Posts'
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
        description: The ID of the question to retrieve.
    responses:
      '200':
        description: 'Success! Show the question.'
      '404':
        description: 'Question not found.'
      '403':
        description: "You don't have access to the question."

/new-question:
  get:
    operationId: R304
    summary: 'R304: Create a new question'
    description: 'Provides the UI to create a new question. Access: AUT'
    tags:
      - 'M03: Posts'
    responses:
      '200':
        description: 'Success! Show create question UI.'
      '302':
        description: 'Access to create question UI denied. Redirect to login form.'
  post:
    operationId: R305
    summary: 'R305: Store a new question'
    description: 'Stores a new question in the system. Access: AUT'
    tags:
      - 'M03: Posts'
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              title:
                type: string
                description: 'The title of the question.'
              text:
                type: string
```

```
                  description: 'The content of the question.'
                tag:
                  type: string
                  description: 'The tag associated with the question.'
              required:
                - title
                - text
      responses:
        '302':
          description: 'Redirect after processing the new question information.'
          headers:
            Location:
              schema:
                type: string
              example:
                302Success:
                  description: 'Successful question create. Redirect to specific question view.'
                  value: '/questions/{id}'
                302Failure:
                  description: 'Failed creation. Redirect back to the form.'
                  value: '/new-question'
        '400':
          description: 'Validation error. Input data is invalid.'

  /api/questions/{id}/delete:
    post:
      operationId: R306
      summary: 'R306 : Delete Question'
      description: 'Deletes a question from the system. Access: QA, ADM'
      tags:
        - 'M03: Posts'
      parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
        description: "The ID of the question post to delete."
      responses:
        '200':
        description: "Question post deleted successfully."
        '302':
          description: "Redirect after question deletion. Redirects to the home page if the user was on the question pag
          headers:
            Location:
              schema:
                type: string
                examples:
                  home:
                    description: "Redirect to the home page."
                    value: "/home"
                  back:
                    description: "Redirect back to the previous page."
                    value: "/previous-page-url"
          '401':
            description: 'You are not authorized to perform this action.'
          '404':
            description: "The question post with the specified ID was not found."

  /api/questions/{id}/edit:
    post:
      operationId: R307
      summary: 'R307: Edit Question Action'
      description: 'Edit post. Access: QA'
      tags:
        - 'M03: Posts'
      parameters:
```

```
      - name: id
        in: path
        required: true
        schema:
          type: integer
        description: "The ID of the question to edit."
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              text:
                type: string
                description: "The updated content of the question."
              title:
                type: string
                maxLength: 255
                description: "The updated title of the question. Must be unique."
              tag_id:
                type: integer
                nullable: true
                description: "The ID of the associated tag. Optional."
            required:
              - text
              - title
    responses:
      '200':
        description: 'Question updated successfully.'
      '400':
        description: "Validation error. Input data is invalid."
      '401':
        description: 'You are not authorized to perform this action.'
      '404':
        description: 'Question not found for the provided ID.'

/questions/{id}/answers:
  post:
    operationId: R308
    summary: "R308: Create an answer"
    description: "Post a new answer to a specific question. Access: AUT"
    tags:
      - "M03: Posts"
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
        description: "The ID of the question for which the answer is being created."
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              newAnswer:
                type: string
                description: "The content of the new answer."
            required:
              - newAnswer
    responses:
      '201':
        description: "Answer created successfully."
        content:
          application/json:
            schema:
              type: object
              properties:
                success:
                  type: boolean
                  example: true
                message:
                  type: string
                  example: "Answer created successfully."
                redirect:
                  type: string
                  example: "/questions/{id}"
      '400':
        description: "Validation error."
      '401':
        description: "Unauthorized. User does not have permission to create an answer for the question."
      '404':
        description: "Question not found."

/answers/{id}:
post:
  operationId: R309
  summary: "R309: Delete an answer"
  description: "Delete a specific answer. Access: AA, ADM"
  tags:
    - "M03: Posts"
  parameters:
    - name: id
      in: path
      required: true
      schema:
        type: integer
      description: "The ID of the answer to delete."
  responses:
    '200':
      description: "Answer deleted successfully."
    '401':
      description: "Unauthorized. User does not have permission to delete the answer."
    '404':
      description: "The answer was not found."

/api/answers/{id}/edit:
  post:
    operationId: R310
    summary: "R310: Edit an answer"
    description: "Edit a specific answer. Access: AA"
    tags:
      - "M03: Posts"
    parameters:
    - name: id
      in: path
      required: true
      schema:
        type: integer
      description: "The ID of the answer to edit."
    requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            text:
              type: string
              maxLength: 5000
              description: "The updated content of the answer."
          required:
            - text
    responses:
      '200':
        description: "Answer edited successfully."
      '400':
        description: "Validation error. Input data is invalid."
      '401':
        description: "Unauthorized. User does not have permission to update the answer."
      '404':
        description: "Answer not found."

/question/{questionId}/answer/delete:
  post:
    operationId: R311
    summary: "R311: Delete an answer"
    description: "Delete a specific answer for a question. Access: USR, ADM"
    tags:
      - "M03: Posts"
    responses:
      '200':
        description: "Answer deleted successfully"
      '403':
        description: "Forbidden action"

/api/home/{section}/edit:
  get:
    operationId: "R312"
    summary: "R312: Render a specific section of questions. Access: AUT"
    description: "Fetches and returns JSON with html for a specific section of questions based on the section type (n
    tags:
      - "M03: Posts"
    parameters:
      - name: section
        in: path
        required: true
        schema:
          type: string
          enum: ["new", "trending", "foryou"]
        description: "The section to fetch questions from. Valid values are 'new', 'trending', or 'foryou'."
      - name: page
        in: query
        required: false
        schema:
          type: integer
          default: 1
        description: "The page number for paginated results. Has a default value of 1."
    responses:
      '200':
        description: "Success! HTML returned for the requested section."
        content:
          application/json:
            schema:
              type: object
              properties:
                html:
                  type: string
                  description: "HTML content for the requested section."
      '400':
        description: "Invalid section provided."
        content:
          application/json:
            schema:
              type: object
              properties:
                error:
                  type: string
                  description: "Error message."

/api/questions/{id}/answers/{page}:
  get:
    operationId: "R313"
    summary: "R313: Get paginated answers for a question. Access: USR"
    description: "Gets JSON with the html to render a list of answers for a given question with pagination."
    tags:
      - "M03: Posts"
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
        description: "ID of the question for which answers are being retrieved."
      - name: page
        in: path
        required: false
        schema:
          type: integer
          default: 1
        description: "The page number for paginated results."
    responses:
      '200':
        description: "Success! Answers retrieved."
        content:
          application/json:
            schema:
              type: array
              items:
                type: string
                description: "HTML content for each answer."
      '404':
        description: "The question was not found."
        content:
          application/json:
            schema:
              type: object
              properties:
                error:
                  type: string
                  description: "There is no question with provided ID."

/api/tags:
  get:
    operationId: "R314"
    summary: "R314: Gets all tags"
    description: "Gets a JSON of all tags. Access: USR"
    tags:
      - "M03: Posts"
    responses:
      '200':
        description: "Success!"
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                properties:
                  id:
                    type: integer
                  name:
                    type: string
              example:
                - id: 1
```

```
                    name: "Programming"
                  - id: 2
                    name: "Networking"
        '403':
          description: "Unauthorized access to view tags."
          content:
            application/json:
              schema:
                type: object
                properties:
                  error:
                    type: string
        '302':
          description: "Redirect to home when unauthorized."
          headers:
            Location:
              schema:
                type: string
                example: "/home"

### M04 Administration ###

  /admin-center:
    get:
      operationId: R401
      summary: 'R401: View Admin Page'
      description: 'Show admin page UI, Access: ADM'
      tags:
        - 'M04: Administration'

      responses:
        '200':
          description: 'OK. Show the admin page UI'
        '403':
          description: 'This action is unauthorized.'

  /api/admin-center/ban/{id}:
    post:
      operationId: R402
      summary: 'R402: Ban user from logging in action'
      description: 'Ban user. Access: ADM'
      tags:
        - 'M04: Administration'
      parameters:
        - in: path
          name: id
          required: true
          schema:
            type: integer
          description: The ID of the user to be banned.
      responses:
        '200':
          description: 'Success! User banned.'
        '401':
          description: 'Unauthorized! You cannot ban this user!'

  /api/admin-center/revoke-ban/{id}:
    post:
      operationId: R403
      summary: 'R403: Revoking ban on user logging in action'
      description: 'Revoke user ban. Access: ADM'
      tags:
        - 'M04: Administration'
      parameters:
        - in: path
          name: id
          required: true
          schema:
            type: integer
          description: The ID of the user to revoke the ban.
      responses:
        '200':
          description: 'User banning has been revoked successfully.'
        '401':
          description: 'Unauthorized! You cannot revoke this user ban.'

  /api/admin-center/moderator/{id}:
    post:
      operationId: R404
      summary: 'R404: Promoting user to moderator role'
      description: 'Promote user to moderator. Access: ADM'
      tags:
        - 'M04: Administration'
      parameters:
        - in: path
          name: id
          required: true
          schema:
            type: integer
          description: The ID of the user to be promoted to moderator.
      responses:
        '200':
          description: 'User has been promoted to moderator successfully.'
        '401':
          description: 'Unauthorized! You cannot revoke this user ban.'


  /api/admin-center/remove-moderator/{id}:
    post:
      operationId: R405
      summary: 'R405: Remove user from moderator role'
      description: 'Remove user from moderator. Access: ADM'
      tags:
        - 'M04: Administration'
      parameters:
        - in: path
          name: id
          required: true
          schema:
            type: integer
          description: The ID of the user to be removed from moderator.
      responses:
        '200':
          description: 'User permissions as moderator have been removed successfully.'
        '401':
          description: 'Unauthorized! You cannot revoke this user ban.'

  /api/admin-center/actions/{id}/edit:
    get:
      operationId: R406
      summary: 'R406: Dropdown of admin actions for user'
      description: 'Gets a json response of html code with the dropdown of admin actions available for a user. Access:
      tags:
        - 'M04: Administration'
      parameters:
        - in: path
          name: id
          required: true
          schema:
            type: integer
          description: The ID of the user to get the dropdown actions.
      responses:
        '200':
          description: Success
          content:
            application/json:
              schema:
                type: array
                items:
                  type: object
                  properties:
                    html:
                      type: string

  /api/users:
    get:
      operationId: R209
      summary: 'R209: Retrieve a list of users.'
      description: 'Retrieve JSON with the html to render all users in the system. Access: ADM'
      tags:
        - 'M04: Administration'
      parameters:
        - in: query
          name: page
          required: false
          description: The page number for pagination.
          schema:
            type: integer
            default: 1
        - in: query
          name: search
          required: false
          description: The seatch value to filter users by tagname.
          schema:
            type: string
      responses:
        200:
          description: A list of users rendered as HTML.
          content:
            application/json:
              schema:
                type: string
```

A8: Vertical prototype

The Vertical Prototype implements the features identified as necessary in the common and theme requirements documents. Based on this information, we prioritized and implemented all user stories classified as high priority, as detailed in the sections below. Additionally, some Medium priority user stories were also implemented.

The purpose of this artifact is to validate the proposed architecture and gain foundational knowledge of the technologies used in the project. Following recommendations, the implementation leverages the LBAW Framework and addresses all layers of the solution's architecture.

The prototype includes the following functionalities: page visualizations (such as home, profile, admin, and search pages), content management (insertion, editing, and deletion of posts), and the display of error and success messages.

1. Implemented Features

1.1. Implemented User Stories

User Story reference	Name	Priority	Description
FR.031	Exact Match Search	High	As a User, I want the ability to search for any question by exact match, so I can quickly find exactly what I'm looking for.
FR.032	Full Text Search	High	As a User, I want the ability to search for any question using full-text search, so I can easily find what I need.
FR.101	View Top Questions	High	As a User, I want to filter the home page by top questions (based on number of likes, and responses), so that I can easily view the most popular questions.

User Story reference	Name	Priority	Description
FR.103	Browse Questions	High	As a User, I want to browse through all questions posted on the system, so that I can discover topics of interest without having to search for specific questions.
FR.105	View Question Details	High	As a User, I want to view the details of a specific question (e.g., author, date, number of likes, and responses), so that I can understand the relevance of the question.
FR.011	Login	High	As a Guest, I want the ability to log in the system, so that I can access my personal profile.
FR.012	Registration	High	As a Guest, I want the option to register an account, so I can engage with and participate in the community.
FR.015	Logout	High	As an Authenticated User, I want the ability to log out of the system, so that I can securely end my session and protect my account.
FR.021	View Profile	High	As an Authenticated User, I want to view my profile, so I can understand what other users can see about me.
FR.022	Edit Profile	High	As an Authenticated User, I want to be able to edit my personal information, so I can keep my information up-to-date.
FR.201	View Personal Feed	High	As an Authenticated User, I want to view a personalized feed of questions (based on the tags and users I follow), so that I can easily interact with content relevant to my interests.
FR.208	View My Questions	High	As an Authenticated User, I want to view a list of all the questions I have posted in my profile, so that I can track the interactions with my questions.
FR.209	View My Answers	High	As an Authenticated User, I want to view a list of all the answers I have given in my profile, so that I can track the interactions with my questions.
FR.202	Post Question	High	As a Question Author, I want to post a question with specific tags to the system, so that I can receive feedback from the community.
FR.401	Edit Question	High	As a Question Author, I want to edit my questions, so that I can correct any mistakes or clarify its content.
FR.402	Delete Question	High	As a Question Author, I want to delete my questions, so that I can remove content that I no longer wish to share.
FR.203	Post Answer	High	As an Answer Author, I want to answer to other users' questions, so that I can share my knowledge and insights on topics of interest.
FR.403	Edit Answer	High	As an Answer Author, I want to edit my answers, so that I can update it with new information.
FR.404	Delete Answer	High	As an Answer Author, I want to delete my answers, so that I can remove information that I no longer wish to share.
FR.042	Administer User Accounts (search, view, edit, create)	High	As an administrator, I want access to multiple admin features, so I can effectively manage the website.
FR.043	Block and Unblock User Accounts	Medium	As an administrator, I want the ability to block and unblock accounts, so I can better manage the community in accordance with the rules.
FR.052	Contextual Error Messages	Medium	As a User, I want to receive contextual error messages, so I can grasp the situation more effectively.

1.2. Implemented Web Resources

This section identifies the web resources that were implemented.

Module M01: Authentication

Web Resource Reference	URL
R101: Login Form	GET /login
R102: Login Action	POST /login

Web Resource Reference	URL
R103: Logout Action	POST /logout
R104: Register Form	GET /register
R105: Register Action	POST /register

Module M02: Users

Web Resource Reference	URL
R201: View User Profile	GET /user/{id}
R202: View User Edit Profile	GET /user/edit-profile
R203: User Edit Name Profile Action	POST /user/edit-profile/name/{id}
R204: User Edit Tagname Profile Action	POST /user/edit-profile/tagname/{id}
R205: User Edit Email Profile Action	POST /user/edit-profile/email/{id}
R206: User Edit Password Profile Action	POST /user/edit-profile/password/{id}
R207: User Edit Age Profile Action	POST /user/edit-profile/age/{id}
R204: User Edit Country Profile Action	POST /user/edit-profile/country/{id}
R204: User Edit Degree Profile Action	POST /user/edit-profile/degree/{id}

Module M03: Posts

Web Resource Reference	URL
R301: View Home Page	GET /home
R302: View Specific Question	GET /questions/{id}
R303: View Create Question Page	GET /new-question
R304: Create Question Action	POST /new-question
R305: Edit Question Action	POST /api/questions/{id}/edit
R306: Like Question Action	POST /api/questions/like
R307: Delete Question Action	POST /api/questions/{id}/delete
R308: Create Answer Action	POST /api/questions/{questionId}/answer/create
R309: Edit Answer Action	POST /api/questions/{questionId}/answer/edit
R310: Delete Answer Action	POST /api/questions/{questionId}/answer/delete

Module M04: Administration

Web Resource Reference	URL
R401: View Admin Page	GET /admin-center
R402: Ban User	POST /api/admin-center/ban/{id}
R403: Revoke Ban User	POST /api/admin-center/revoke-ban/{id}
R404: Make User Moderator	POST /api/admin-center/moderator/{id}
R405: Remove User Moderator	POST /api/admin-center/remove-moderator/{id}
R406: Get Users	GET /api/users

2. Prototype

To start the Docker image, run the following command:

```
docker run -d --name lbaw24042 -p 8001:80 gitlab.up.pt:5050/lbaw/lbaw2425/lbaw24042
```

To test the implemented features, the following user credentials can be used:

admin user:

- email: ines.mendes@gmail.com
- password: securepass

regular user:

- email: joao.silva@gmail.com
- password: password123

The source code can be found in our Git repository.

Revision history

Changes made to the first submission:

No changes were made so far!

`GROUP24042, 26/11/2024`

- Eduardo Cruz, up202205138@up.pt
- Jorge Mesquita, up202108614@up.pt
- Mariana Marques, up201606434@up.pt
- Tatiana Lin, up202206371@up.pt

Editor of Submission: Eduardo Cruz

Comments