PA: Product and Presentation

A9 - Product

In this artifact, we will describe the final implementation of our project.

We developed a website, SmokeyGlass, that allows its users to buy sunglasses. Here, they can filter through categories and brands or even look for a specific product. After finding the desired sunglases, the user may proceed to checkout and create an order or they can continue to browse the site. They can calso manage their profile, their wishlist and shopping cart, and review their purchased products.

1. Installation

URL to the source code: https://git.fe.up.pt/lbaw/lbaw2223/lbaw2213

Command to test the Docker image:

```
docker run -it -p 8000:80 --name=lbaw2213 -e DB_DATABASE="lbaw2213" -e
DB_SCHEMA="lbaw2213" -e DB_USERNAME="lbaw2213" -e DB_PASSWORD="CyZjqIxK"
git.fe.up.pt:5050/lbaw/lbaw2223/lbaw2213
```

2. Usage

URL to the product: http://lbaw2213.lbaw.fe.up.pt

2.1 Administration Credentials

Email	Password
luis@mail.com	123456
marta@mail.com	123456
john@mail.com	123456
cath@mail.com	123456

2.2 User Credentials

Type	Email	Password
Customer	joao@mail.com	123456
Customer	maria@mail.com	123456
Customer	marco@mail.com	123456
Customer	rui@mail.com	123456

3. Apllication Help

In order to help the user browse our site, we created a frequently asked questions page. The user can access this page by clicking the FAQ area on the footer.

4. Input Validation

To validate user input we used the validations mechanisms available in laravel. Below there's an example of the validation function used for the edit product action.

```php
protected function validator_edit(array $data)
{
  return Validator::make($data, [
    'id_product' => 'integer',
    'name' => 'string|max:255|nullable',
    'price' => 'nullable',
    'stock' => 'integer|nullable',
    'description' => 'string|nullable',
    'category' => 'integer|nullable',
    'brand' => 'integer|nullable',
    'photo' => 'nullable'
  ]);
}
```

Html also supports validation inside the input forms. There, we can specify the type of data we want to receive and filter wrong inputs. We used this restrictions, for example, to validate the file format of the user's profile picture, when creating an account.

```html
<div class="form-group">
    <label for="photo" class="form-label mt-4">Profile Pricture</label>
    <input class="form-control" id="photo" type="file" name="photo"
accept=".jpg,.png,.jpeg" />
    @if ($errors->has('photo'))
        <span class="error">
            {{ $errors->first('photo') }}
        </span>
    @endif
</div>
```

Input validation has been used in several cases, all of which were forms, the method used was a combination of html validation, such as required fields, and php specifications in the controllers.

5. Check Acessibility and Usability

In order to make our site more acessible, we used some html element. For example, in order to make forms more understandable for all users, all of the forms that are displayed on our site are wrapped in fielset tags which make Also, all the images displayed contain a alternative text. This attribute allows the user to recieve information about the image in question in cases where they can't see it.

6. HTML & CSS Validation

- https://git.fe.up.pt/lbaw/lbaw2223/lbaw2213/-/tree/main/validation

7. Revisions to the Project

- Added two actors, Buyer and Reviewer.
- Changes in the priority of some user stories.
- Changed multiplicity between address and order and address and customer.
- Created customer_address table
- Report table update.
- Added user stories US26 e US27
- Addes to module M01 routes R109, R110, R111 and R112.

8. Web Resources Specification

Module M01: Authentication and Individual Profile

Web Resource Reference	URL
R101: Login Form	GET /login
R102: Login Action	POST /login
R103: Logout Action	POST /logout
R104: Register Form	GET /register
R105: Register Action	POST /register
R106: View Profile	GET /user/{id}
R107: Register Form	GET /user/{id}/edit
R108: Register Action	PUT /edit_user
R109: View Own Addresses	GET /user/{id}/addresses
R110: View Order History	GET /user/{id}/past_orders
R111: View Orders	GET /user/{id}/orders
R112: View Individual Order	GET /user/{id}/order/{id_order}
R113: Edit Profile Form	GET /user/{id}/edit
R114: Edit Profile Action	PUT /edit_user
R115: Delete Account	PUT /user/{id}/delete_user

Module M02: Products

Web Resource Reference	URL
R201: List Products	GET /
R202: Search products	GET /search

Module M03: Product and Reviews

Web Resource Reference	URL
R301: View Product	GET /product/{id}
R302: Create Review	POST /product/{id}
R303: Add Product Form	GET /product/add
R304: Add Product Action	POST /product/add_product
R305: Edit Product Form	GET /product/{id}/edit
R306: Edit Product Action	PUT /product/{id}/edit_product
R307: Delete Product Action	GET /product/{id}/delete
R308: Review Report Form	GET /report/review/{id}/{motive}
R309: Review Report Action	POST /report/review/{id}/{motive}
R310: Delete Review	DELETE /api/review/{id}

Module M04: Wishlist and Shopping cart

Web Resource Reference	URL
R401: View Shopping Cart	GET /shopping_cart
R402: Add product to shopping cart	POST /api/product/{id}
R403: Remove product from shopping cart	DELETE /api/product/{id}
R404: Checkout	POST /checkout
R405: View Wish list	GET /wishlist
R406: Add product to wish list	POST /api/wishlist/product/{id}
R407: Remove product from wish list	DELETE /api/wishlist/product/{id}

Module M05: Management

Web Resource Reference	URL
R501: Search users API	GET /api/users
R502: View another user profile	GET /management/users/{id}
R503: Edit Profile Form	GET /management/users/{id}/edit
R504: Edit Profile Action	PUT /management/users/{id}/edit
R505: Create User Form	GET /management/users/create
R506: Create User Action	POST /management/users/create
R507: Change Order Status	PATCH /api/order/{id}

Web Resource Reference	URL
R508: View Reports Page	GET /reports
R509: Ban User	POST /api/ban/{id}/{motive}/{id_review}
R510: Dismiss Report	PUT /api/report/dismiss/{id_user}/{id_review}/{motive}

Module M06: Static Pages

Web Resource Reference	URL
R601: View About Page	GET /about
R602: View Contact Us Page	GET /contact

Link to the Swagger generated documentation: https://app.swaggerhub.com/apis/UP202003574_1/lbaw-smokey_glass_web_api/1.0.0

```yaml
openapi: 3.0.0

info:
 version: 1.0.0
 title: 'LBAW SmokeyGlass Web API'
 description: 'Web Resources Specification (A7) for SmokeyGlass'

servers:
# Added by API Auto Mocking Plugin
- description: SwaggerHub API Auto Mocking
  url: https://virtserver.swaggerhub.com/UP202003574_1/lbaw/1.0.0
- url: http://lbaw.fe.up.pt
  description: Production server

externalDocs:
 description: Find more info here.
 url: https://git.fe.up.pt/lbaw/lbaw2223/lbaw2213/-/wikis/eap

tags:
 - name: 'M01: Authentication and Individual Profile'
 - name: 'M02: Products'
 - name: 'M03: Product and Reviews'
 - name: 'M04: Wishlist and Shopping cart'
 - name: 'M05: Management'
 - name: 'M06: Static Pages'
paths:
  /login:
    get:
      operationId: R101
      summary: 'R101: Login Form'
      description: 'Provide login form. Access: PUB'
      tags:
        - 'M01: Authentication and Individual Profile'
      responses:
```

```yaml
          '200':
            description: 'Ok. Show Log-in UI'
      post:
        operationId: R102
        summary: 'R102: Login Action'
        description: 'Processes the login form submission. Access: PUB'
        tags:
          - 'M01: Authentication and Individual Profile'
        requestBody:
          required: true
          content:
            application/x-www-form-urlencoded:
              schema:
                type: object
                properties:
                  username:
                    type: string
                  password:
                    type: string
                  required:
                        - username
                        - password

        responses:
          '302':
            description: 'Redirect after processing the login credentials.'
            headers:
              Location:
                schema:
                  type: string
                examples:
                  302Success:
                    description: 'Successful authentication. Redirect to  main
  page.'
                    value: '/mainpage'
                  302Error:
                    description: 'Failed authentication. Redirect to login form.'
                    value: '/login'
    /logout:
     post:
        operationId: R103
        summary: 'R103: Logout Action'
        description: 'Logout the current authenticated user. Access: USR, ADM'
        tags:
          - 'M01: Authentication and Individual Profile'
        responses:
          '302':
            description: 'Redirect after processing logout.'
            headers:
              Location:
                schema:
                  type: string
                examples:
                  302Success:
```

```yaml
                        description: 'Successful logout. Redirect to mainpage.'
                        value: '/mainpage'
    /register:
     get:
       operationId: R104
       summary: 'R104: Register Form'
       description: 'Provide new user registration form. Access: PUB'
       tags:
         - 'M01: Authentication and Individual Profile'
       responses:
         '200':
           description: 'Ok. Show Sign-Up UI'

     post:
       operationId: R105
       summary: 'R105: Register Action'
       description: 'Processes the new user registration form submission. Access:
 PUB'
       tags:
         - 'M01: Authentication and Individual Profile'

       requestBody:
         required: true
         content:
           application/x-www-form-urlencoded:
             schema:
               type: object
               properties:
                 username:
                   type: string
                 email:
                   type: string
                 password:
                   type: string
                 gender:
                   type: string
                 date_of_birth:
                   type: string
                 picture:
                   type: string
                   format: binary
               required:
                       - username
                       - email
                       - password
                       - date_of_birth
                       - picture
       responses:
         '302':
           description: 'Redirect after processing the new user information.'
           headers:
             Location:
               schema:
                 type: string
```

```
                examples:
                  302Success:
                    description: 'Successful authentication. Redirect to main page.'
                    value: '/mainpage'
                  302Failure:
                    description: 'Failed authentication. Redirect to register form.'
                    value: '/register'
  /users/{id}:
   get:
     operationId: R106
     summary: 'R106: View user profile'
     description: 'Show the individual user profile. Access: USR'
     tags:
       - 'M01: Authentication and Individual Profile'

     parameters:
       - in: path
         name: id
         schema:
           type: integer
         required: true

     responses:
       '200':
         description: 'Ok. Show User Profile UI'
  /users/{id}/edit:
     parameters:
         - in: path
           name: id
           schema:
             type: integer
           required: true

    get:
      operationId: R107
      summary: 'R107: Edit Profile Form'
      description: 'Provide edit profile form. Access: USR'
      tags:
        - 'M01: Authentication and Individual Profile'
      responses:
        '200':
          description: 'Ok. Show Edit Profile Form UI'

    put:
      operationId: R108
      summary: 'R108: Edit Profile Action'
      description: 'Processes the edit profile form submission. Access: USR'
      tags:
        - 'M01: Authentication and Individual Profile'

      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
```

```yaml
              schema:
                type: object
                properties:
                 username:
                    type: string
                 email:
                    type: string
                 password:
                    type: string
                 gender:
                    type: string
                 date_of_birth:
                    type: string
                 picture:
                    type: string
                    format: binary

     responses:
       '302':
        description: 'Redirect after processing the new user information.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Successful edit. Redirect to user profile.'
                value: '/users/{id}'
              302Failure:
                description: 'Failed edit. Redirect to user profile.'
                value: '/users/{id}/edit'

 /users/{id}/addresses:
   parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true
   get:
     operationId: R109
     summary: 'R109: View Own Addresses'
     description: 'Provide view of the users addresses Access: USR, ADM'
     tags:
       - 'M01: Authentication and Individual Profile'
     responses:
       '200':
        description: 'Ok. Show user addresses UI'

 /user/{id}/past_orders:
   parameters:
      - in: path
        name: id
        schema:
```

```
              type: integer
            required: true
    get:
      operationId: R110
      summary: 'R110: View Order History'
      description: 'Provide view of the users previous orders. Access: USR, ADM'
      tags:
        - 'M01: Authentication and Individual Profile'
      responses:
        '200':
          description: 'Ok. Show user past orders UI'

  /user/{id}/orders:
    parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true
    get:
      operationId: R111
      summary: 'R111: View Orders'
      description: 'Provide view of the users orders. Access: USR, ADM'
      tags:
        - 'M01: Authentication and Individual Profile'
      responses:
        '200':
          description: 'Ok. Show user orders UI'

  /user/{id}/order/{id_order}:
    parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true
      - in: path
        name: id_order
        schema:
          type: integer
        required: true
    get:
      operationId: R112
      summary: 'R112: View Individual Order'
      description: 'Provide view of the specific. Access: USR, ADM'
      tags:
        - 'M01: Authentication and Individual Profile'
      responses:
        '200':
          description: 'Ok. Show user orders UI'
        '404':
          description: 'Order not found! '

  /user/{id}/delete_user:
```

```
        parameters:
          - in: path
            name: id
            schema:
              type: integer
            required: true
      put:
        operationId: R115
        summary: 'R115: Delete Account'
        description: 'Processes the delete account. Access: USR,ADM'
        tags:
          - 'M01: Authentication and Individual Profile'
        responses:
          '302':
            description: 'Redirect to main page.'
  /product/{id}:
    parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true

    get:
      operationId: R301
      summary: 'R301: View product page'
      description: 'Show the individual product page. Access: PUB'

      tags:
        - 'M03: Product and Reviews'

      responses:
        '200':
          description: 'Ok. Show Product Page UI'
        '404':
          description: 'Product not found!'

    post:
      operationId: R401
      summary: 'R401: Add product to shopping cart'
      description: "Shows the admin's options. Access: ADM"
      tags:
        - 'M04: Wishlist and Shopping cart'

      requestBody:
       required: true
       content:
         application/x-www-form-urlencoded:
           schema:
             type: object
             properties:
               id:
                 type: number
             required:
```

```yaml
                          - id
      responses:
        '302':
          description: 'Redirect after processing the new user information.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful added items. Redirect to product page.'
                  value: '/product/{id}'
                302Failure:
                  description: 'Failed to add items. Redirect to product page.'
                  value: '/product/{id}'
  /wishlist/product/{id}:
    parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true
    post:
      operationId: R406
      summary: 'R406: Add product to wish list'
      description: "Add product to wishlist. Access: USR"
      tags:
        - 'M04: Wishlist and Shopping cart'

      responses:
        '302':
          description: 'Successful added item. Redirect to wishlist page.'
    delete:
      operationId: R407
      summary: 'R407: Remove product from wish list'
      description: "Remove product from wishlist. Access: USR"
      tags:
        - 'M04: Wishlist and Shopping cart'
      responses:
        '302':
          description: 'Ok. Show Wishlist UI'
  /wishlist:
    get:
      operationId: R405
      summary: 'R405: View Wish list'
      description: 'Show the user wishlist. Access: USR'

      tags:
        - 'M04: Wishlist and Shopping cart'

      responses:
        '200':
          description: 'Ok. Show Wishlist UI'
  /api/order/{id}:
```

```yaml
      parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true
      get:
        operationId: R507
        summary: 'R507: Change Order Status'
        description: 'Change order status. Access: ADM'

        tags:
          - 'M05: Management'

        responses:
          '302':
            description: 'Successful change order status. Redirect to order page.'
  /product/add:
    get:
      operationId: R303
      summary: 'R303: Add Product Form'
      description: 'Show the Create product form. Access: ADM'

      tags:
        - 'M03: Product and Reviews'

      responses:
        '200':
          description: 'Ok. Show Create Product UI'
        '404':
          description: 'Product not found!'
    post:
      operationId: R304
      summary: 'R304: Add Product Action'
      description: 'Create product action. Access: ADM'

      tags:
        - 'M03: Product and Reviews'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                name:
                  type: string
                price:
                  type: integer
                id_category:
                  type: string
                picture:
                  type: string
                description:
```

```yaml
                               type: string
                    stock:
                      type: integer
                    id_brand:
                      type: integer
                  required:
                    - name
                    - price
                    - id_category
                    - id_brand
                    - picture
                    - description
                    - stock

      responses:
        '302':
          description: 'Redirect after processing and validating the information.'
/product/{id}/edit:
  parameters:
  - in: path
    name: id
    schema:
      type: integer
    required: true
  get:
    operationId: R305
    summary: 'R305: Edit Product Form'
    description: 'Show the edit product form. Access: ADM'

    tags:
      - 'M03: Product and Reviews'

    responses:
      '200':
        description: 'Ok. Show Edit Product UI'
      '404':
        description: 'Product not found!'
  put:
    operationId: R306
    summary: 'R306: Edit Product Action'
    description: 'Edit product action. Access: ADM'

    tags:
      - 'M03: Product and Reviews'
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              name:
                type: string
              price:
```

```yaml
                type: integer
              id_category:
                type: string
              picture:
                type: string
              description:
                    type: string
              stock:
                type: integer
              id_brand:
                type: integer
      responses:
        '302':
          description: 'Redirect after processing and validating the information.'
/product/{id}/delete:
    parameters:
    - in: path
      name: id
      schema:
        type: integer
      required: true
    put:
      operationId: R307
      summary: 'R307: Delete Product Action'
      description: 'Delete product action. Access: ADM'

      tags:
        - 'M03: Product and Reviews'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                id_product:
                    type: integer
      responses:
        '302':
          description: 'Redirect after processing the information.'

/report/review/{id}/{motive}:
    parameters:
    - in: path
      name: id
      schema:
        type: integer
      required: true
    - in: path
      name: motive
      schema:
        type: integer
      required: true
    get:
```

```yaml
      operationId: R308
      summary: 'R308: Review Report Form'
      description: 'Show the Report form. Access: USR,ADM'

      tags:
        - 'M03: Product and Reviews'

      responses:
        '200':
          description: 'Ok. Show Report Form UI'
        '404':
          description: 'Review not found!'
  put:
    operationId: R309
    summary: 'R309: Review Report Action'
    description: 'Report review action. Access: USR,ADM'

    tags:
      - 'M03: Product and Reviews'
    parameters:
      - in: path
        name: id
        description: Product identifier
        schema:
          type: integer
        required: true
    responses:
        '302':
          description: 'Redirect after processing and validating the
information.'
  /api/review/{id}:
    delete:
      operationId: R310
      summary: 'R310: Delete Review'
      description: 'Processes the delete product action. Access: ADM'
      tags:
        - 'M03: Products and Reviews'

      parameters:
        - in: path
          name: id
          description: Product identifier
          schema:
            type: integer
          required: true

      responses:
        '302':
          description: 'Redirect after processing the product deletion.'
          headers:
            Location:
              schema:
                type: string
              examples:
```

```yaml
                        302Success:
                          description: 'Successful deletion. Redirect to main page.'
                          value: '/mainpage'
                        302Error:
                          description: 'Failed deletion. Redirect to main page.'
                          value: '/mainpage'
  /api/products:
    get:
      operationId: R201
      summary: 'R201: Search products API'
      description: 'Searches for products and returns the results as JSON. Access:
PUB.'

      tags:
        - 'M02: Products'

      parameters:
        - in: query
          name: query
          description: 'String to use for full-text search'
          schema:
            type: string
          required: false
      responses:
        '200':
          description: Success
          content:
            application/json:
              schema:
                type: array
                items:
                  type: object
                  properties:
                    id:
                      type: integer
                    name:
                      type: string
                    description:
                      type: string
                    price:
                      type: number
                    stock:
                      type: integer
                    id_category:
                      type: integer
                    id_brand:
                      type: integer
                    details:
                      type: array
                      items:
                        type: array
                        items:
                          type: string
  /shoppingCart:
```

1/4/2023

```
    get:
      operationId: R402
      summary: 'R402: View user shopping cart'
      description: 'Show the individual user shopping cart. Access: USR'
      tags:
        - 'M04: Wishlist and Shopping cart'
      responses:
        '200':
          description: Success
          content:
            application/json:
              schema:
                type: array
                items:
                  type: object
                  properties:
                    name:
                      type: string
                    description:
                      type: string
                    price:
                      type: number
                    quantity:
                      type: number
    post:
      operationId: R403
      summary: 'R403: Checkout'
      description: 'Create a order with the shopping cart. Access: USR'
      tags:
        - 'M04: Wishlist and Shopping cart'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                id:
                  type: number
              required:
                    - id
      responses:
        '302':
          description: 'Redirect after processing the new information.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful added items. Redirect to product page.'
                  value: '/mainpage'
                302Failure:
                  description: 'Failed to add items. Redirect to product page.'
```

```yaml
                         value: '/api/shoppingCart'
  /api/users:
    get:
      operationId: R501
      summary: 'R501: Search users API'
      description: "Searches for users and returns the result as JSON. Access:
ADM"
      tags:
        - 'M05: Management'
      parameters:
        - in: query
          name: query
          description: 'String to use for full-text search'
          schema:
            type: string
          required: false
      responses:
       '200':
         description: Success
         content:
           application/json:
             schema:
               type: array
               items:
                 type: object
                 properties:
                     id:
                       type: integer
                     username:
                       type: string
                     picture:
                       type: string
                       format: binary
  /management/users/{id}:
    get:
      operationId: R502
      summary: 'R502: View another user profile'
      description: 'Show the individual user profile. Access: ADM'
      tags:
        - 'M05: Management'

      parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true

      responses:
       '200':
         description: 'Ok. Show User Profile UI'
  /management/users/{id}/edit:
    parameters:
        - in: path
```

```yaml
        name: id
        schema:
          type: integer
        required: true

    get:
      operationId: R503
      summary: 'R503: Edit Profile Form'
      description: 'Provide edit profile form. Access: ADM'
      tags:
        - 'M05: Management'
      responses:
        '200':
          description: 'Ok. Show Edit Profile Form UI'

    put:
      operationId: R504
      summary: 'R504: Edit Profile Action'
      description: 'Processes the edit profile form submission. Access: ADM'
      tags:
        - 'M05: Management'

      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                email:
                  type: string
                password:
                  type: string
                gender:
                  type: string
                date_of_birth:
                  type: string
                picture:
                  type: string
                  format: binary

      responses:
        '302':
          description: 'Redirect after processing the new user information.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful edit. Redirect to user profile.'
                  value: '/management/users/{id}'
```

```
                  302Failure:
                    description: 'Failed edit. Redirect to profile edit form.'
                    value: '/management/users/{id}/edit'
  /management/users/create:
    get:
      operationId: R505
      summary: 'R505: Create User Form'
      description: 'Create User Form. Access: ADM'
      tags:
       - 'M05: Management'
      responses:
       '200':
          description: 'Ok. Show Create User Form UI'

    post:
      operationId: R506
      summary: 'R506: Create User Action'
      description: 'Processes the new user information in the form. Access: ADM'
      tags:
       - 'M05: Management'

      requestBody:
       required: true
       content:
         application/x-www-form-urlencoded:
           schema:
             type: object
             properties:
               username:
                 type: string
               email:
                 type: string
               password:
                 type: string
               gender:
                 type: string
               date_of_birth:
                 type: string
               picture:
                 type: string
                 format: binary
             required:
                   - username
                   - email
                   - password
                   - date_of_birth
                   - picture
      responses:
       '302':
         description: 'Redirect after processing the new user information.'
         headers:
           Location:
             schema:
               type: string
```

```
              examples:
                302Success:
                  description: 'Successful authentication. Redirect to new user
page.'
                  value: '/management/users/{id}'
                302Failure:
                  description: 'Failed authentication. Redirect to create user
form.'
                  value: '/management/users/create'
  /about_us:
    get:
      operationId: R601
      summary: 'R601: View About Us page'
      description: 'Shows some information about the company. Access: PUB.'
      tags:
        - 'M06: Static Pages'
      responses:
        '200':
          description: 'Ok. Show About Us Page UI'
  /contact_us:
    get:
      operationId: R602
      summary: 'R602: View Contact Us page'
      description: 'Shows the contacts of the company. Access: PUB.'
      tags:
        - 'M06: Static Pages'
      responses:
        '200':
          description: 'Ok. Show Contact Us Page UI'
  /reports:
    get:
      operationId: R508
      summary: 'R508: View Reports Page '
      description: 'Show the list of waiting reports. Access: ADM'

      tags:
        - 'M05: Management'

      responses:
        '200':
          description: 'Ok. Show Reports UI'

  /api/ban/{id}/{motive}/{id_review}:
    post:
      operationId: R509
      summary: 'R509: Ban User'
      description: 'Ban a users by a report. Access: ADM'
      tags:
        - 'M05: Management'

      parameters:
        - in: path
          name: id
          description: User id
```

```yaml
              schema:
                type: integer
              required: true

          - in: path
            name: motive
            description: Report Motive
            schema:
              type: integer
            required: true

          - in: path
            name: id_review
            description: ID of the review that caused the ban
            schema:
              type: integer
            required: true

      responses:
        '302':
          description: 'User banned with success.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful ban. Redirect to Report Management
Page.'
                  value: '/reports'
                302Error:
                  description: 'Failed ban. Redirect to Report Management Page.'
                  value: '/reports'
  /api/report/dismiss/{id_user}/{id_review}/{motive}:
    put:
      operationId: R510
      summary: 'R510: Dismiss Report'
      description: 'Dismisses a report made by a user. Access: ADM'
      tags:
        - 'M05: Management'

      parameters:
        - in: path
          name: id_user
          description: User id
          schema:
            type: integer
          required: true

        - in: path
          name: motive
          description: Report Motive
          schema:
            type: integer
```

```
            required: true

      - in: path
        name: id_review
        description: ID of the review that caused the ban
        schema:
          type: integer
        required: true


    responses:
      '302':
        description: 'Report Dismissed with success.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Successful dismissed Redirect to Report
Management Page.'

                value: '/reports'
              302Error:
                description: 'Failed dismiss. Redirect to Report Management
Page.'

                value: '/reports'
```

9. Implementation Details

9.1 Libraries Used

We didn't use any libraries.

9.2 User Stories

Identifier	Name	Module	Priority	Team Member	State
US01	View Products List	M02: Products	high	**Henrique Seabra**, Beatriz Fontão	100%
US02	View Product Details	M03: Product and Reviews	high	**Beatriz Fontão**	100%
US03	Add Product to Shopping Cart	M04: Wishlist and Shopping cart	high	**Henrique Seabra**	100%
US04	Manage Shopping Cart	M04: Wishlist and Shopping cart	high	**Henrique Seabra**	100%
US05	Search Products	M02: Products	high	**Henrique Seabra**	100%

Identifier	Name	Module	Priority	Team Member	State
US11	Log In Account	M01: Authentication and Individual Profile	high	**Beatriz Fontão**	100%
US12	Create An Account	M01: Authentication and Individual Profile	high	**Beatriz Fontão**	100%
US21	View Profile	M01: Authentication and Individual Profile	high	**Beatriz Fontão**	100%
US22	Edit Profile	M01: Authentication and Individual Profile	high	**Beatriz Fontão**	100%
US23	Logout From Account	M01: Authentication and Individual Profile	high	**Beatriz Fontão**	100%
US24	View Purchase History	M01: Authentication and Individual Profile	high	**Beatriz Fontão**	100%
US25	Checkout	M04: Wishlist and Shopping cart	high	**José Ramos**	100%
US51	Search for other User Accounts	M05: Management	high	**José Ramos**, Beatriz Fontão	100%
US52	View other User Accounts	M05: Management	high	**José Ramos**, Beatriz Fontão	100%
US53	Edit other User Accounts	M05: Management	high	**José Ramos**, Beatriz Fontão	100%
US54	Create user account	M05: Management	high	**José Ramos**, Beatriz Fontão	100%
US06	Browse Product Categories	M02: Products	high	**Henrique Seabra**	100%
US07	View Product Reviews	M03: Product and Reviews	medium	**Henrique Seabra**	100%
US08	Remove Product From Shopping Cart	M04: Wishlist and Shopping cart	medium	**Henrique Seabra**	100%
US26	View Wishlist	M04: Wishlist and Shopping cart	medium	**Henrique Seabra**, Beatriz Fontão	100%
US27	Add Product to Wishlist	M04: Wishlist and Shopping cart	medium	**Henrique Seabra**	100%
US28	Remove Product From Wishlist	M04: Wishlist and Shopping cart	medium	**Henrique Seabra**	100%

Identifier	Name	Module	Priority	Team Member	State
US210	Report Review	M03: Product and Reviews	low	**Henrique Seabra**	100%
US31	Review Purchased Product	M03: Product and Reviews	medium	**Henrique Seabra**	100%
US33	Track Order	M01: Authentication and Individual Profile	medium	**Beatriz Fontão**	100%
US37	Notification when Product on Cart Price Change	M01: Authentication and Individual Profile	low	**José Ramos**	100%
US41	Edit Review	M03: Product and Reviews	medium	**Henrique Seabra**	100%
US42	Delete Review	M03: Product and Reviews	medium	**Henrique Seabra**	100%
US55	Add/Delete Products	M03: Product and Reviews	medium	**Beatriz Fontão**	100%
US56	Manage Products Information	M03: Product and Reviews	medium	**Beatriz Fontão**	100%
US57	Manage Products Stock	M03: Product and Reviews	medium	**Beatriz Fontão**	100%
US59	Manage Product Categories	M03: Product and Reviews	medium	**Beatriz Fontão**	100%
US510	View Users' Purchase History	M05: Management	high	**Beatriz Fontão**	100%
US511	Manage Order Status	M05: Management	high	**Beatriz Fontão**	100%
US512	View reports	M05: Management	low	**Henrique Ferreira**	100%
US513	Ban user	M05: Management	low	**Henrique Ferreira**	50%
US513	Dismiss report	M05: Management	low	**Henrique Ferreira**	100%

A10 - Presentation

1. Product presentation

In our website, SmokeyGlasses, we sell a large variety of sunglasses in order to fit everyone's personal style.

Our users are divided in three categories: visitor, these are unauthenticated users that are just browsing through the site; then, we have authenticated users, that have their own profile and can purchase products;

and finally, we have the administrators.

Even though the visitors can't checkout, they can add and remove products to their shopping cart. This shopping cart can then be purchased once the guest logs in or creates and account.

When a customer gets an account, they recieve access to a wishlist where they can place product to buy later on. Once they have made some orders, they can access the page of each of those orders to check details, such as date of purchase or the order's status. They can also edit or even delete their account.

The administrators run the site and can manage all its elements. For example: they can add, delete and edit products, including changing brand and category. They can also manage user accounts and access the users' previous and current orders.

2. Video presentation

https://drive.google.com/file/d/1zhpdxea6Y2Apt2Ef5IF-jtg268N_bwQo/view?usp=sharing

Revision history

1. Added User Stories US508, US509 e US510

GROUP2213, 03/01/2023

- Ana Beatriz Fontão, up202003574
- Henrique Seabra Ferreira, up202007044
- José Pedro Ramos, up202005460