

# PA: Product and Presentation

---

Bidding the future and Selling the past.

## A9: Product

Our website is designed to be a comprehensive information system that supports buying and selling a variety of items through a user-friendly web interface. Registered users can easily place items up for auction or bid on existing items that are available for auction. The system automatically manages the bidding process, including enforcing deadlines and determining the winning bid in a fair and transparent manner.

In addition to facilitating the auction process, our website also offers a range of tools and features to enhance the user experience. System administrators have the ability to stop auctions, block user accounts, or delete content as necessary to maintain the integrity and safety of the platform. We are dedicated to providing a reliable and enjoyable auction experience for all of our users.

Whether you are looking to sell items or find great deals on a wide range of products, InfinityAuctions is the perfect place to do it. We hope you will consider using our website for all of your buying and selling needs.

### 1. Installation

Link to the release with the final version of the source code in the group's Git repository.

```
docker run -it -p 8000:80 --name=lbaw2271 -e DB_DATABASE="lbaw2271" -e  
DB_SCHEMA="lbaw2271" -e DB_USERNAME="lbaw2271" -e DB_PASSWORD="00lRpmGt"  
git.fe.up.pt:5050/lbaw/lbaw2223/lbaw2271
```

### 2. Usage

URL to the product: <http://lbaw2271.lbaw.fe.up.pt>

#### 2.1. Administration Credentials

Administration URL: <http://lbaw2271.lbaw.fe.up.pt/users/1002>

Email	Password
testadmin@fe.up.pt	123456

#### 2.2. User Credentials

Type	Username	Password
Banned Account	banned@fe.up.pt	123456
Regular user	regular@fe.up.pt	123456

### 3. Application Help

In Infinity Auctions, help has been implemented through the use of a FAQ and a contact page. The FAQ provides answers to commonly asked questions, while the contact page allows users to reach out to the company through a form or by accessing their social media profiles. Additionally, the website includes error messages that provide information on what has caused the error and how to resolve it. Overall, our website was designed to be intuitive and address any issues that users may encounter. For example, when inputting invalid information on creating an auction, the user is alerted of this.

#### 4. Input Validation

In every form submission, the provided data is validated on the server-side using the built-in validation rules of Laravel. Moreover, we included two custom validation rules to properly validate the content of user's addresses (IsValidAddress) and to match the provided password with the stored one (MatchPassword) (this is required when the user is changing the password in the User page). Furthermore, we validate the input in every form's input field using the `pattern` attribute alongside regular expressions. An example of front end input validation (edit profile page): `<input type="text" name="name" pattern="^[a-zA-Z\s]{1,30}$" title="Only letters and white spaces are allowed. Maximum 30 characters." class="form-control is-invalid" placeholder="Name"`

# Edit Profile

Name

Conta teste<script>alert('ola')</script>|



Please match the requested format.

Only letters and white spaces are allowed. Maximum 30 characters.

Cellphone

918499953

Address

623 Vicksburg st

Birth Date

03/01/1994



Profile image:

Choose File

No file chosen

No file chosen

Save

If we disable this verification, we obtain the error messages from the server:

# Edit Profile

Name

<script>alert('ola')</script>

**Only letters and white spaces are allowed.  
Maximum 30 characters.**

Email address

teste@fe.up.pt

Cellphone

918499953

Address

623 Vicksburg st

Birth Date

03 / 01 / 1994



Profile image:

Browse...

No file selected.

Save

This form's input was validated in the server side using the following code:

```
$validated = $request->validate([
    'name' => 'required|string|max:30|regex:/^[a-zA-Z\s]{1,30}$/',
    'cellphone' => ['required', 'numeric', 'digits:9',
Rule::unique('users')->ignore($user->id)],
    'email' => ['required', 'email', 'max:255',
Rule::unique('users')->ignore($user->id)],
    'birth_date' => 'required|date|before:-18 years',
    'address' => ['required', Rule::unique('users')->ignore($user->id), new IsValidAddress],
    'profile_image' => 'mimes:jpeg,jpg,png,gif',
    ['birth_date.before' => "You need to be, at least, 18
years old to sign up in our website.",
    'name.regex' => "Only letters and white spaces are
allowed. Maximum 30 characters."
]);
```

## 5. Check Accessibility and Usability

Accessibility: [Report](#)

Usability: [Report](#)

## 6. HTML & CSS Validation

### HTML Validation:

- [About us](#)
- [Contacts](#)
- [FAQ](#)
- [Services](#)
- [Create auction](#)
- [Edit auction](#)
- [Other user page](#)
- [Self user page](#)
- [Balance](#)
- [Auction Page](#)
- [Search Page](#)
- [Admin Panel](#)

### CSS Validation:

- [Admin Panel](#)
- [Auction](#)
- [Auction Card](#)
- [Auction Edit](#)
- [Balance](#)
- [Checkout](#)
- [Confirmation Modal](#)

- [Contacts](#)
- [FAQ](#)
- [Footer](#)
- [Header](#)
- [Login](#)
- [Main](#)
- [Miligram.min](#)
- [Register](#)
- [Search page](#)
- [All users](#)
- [Create Auction](#)
- [User](#)
- [Users](#)

## 7. Revisions to the Project

Changes made to A2:

- Added user stories US42 (Notification), US23(Recover password), US26(Order search results) and US37(Logout)

Regarding the A7 artifact, we made heavy changes (mostly additions to the original artifact) that are now all present in the A9 artifact

## 8. Web Resources Specification

[a9\\_openapi.yaml](#)

```
openapi: 3.0.0

info:
  version: '1.0'
  title: 'LBAW Infinity Auction Web API'
  description: 'Web Resources Specification (A9) for Infinity Auction'

servers:
  - url: https://lbaw2271.lbaw.fe.up.pt/
    description: Production server

externalDocs:
  description: Find more info here.
  url:
https://git.fe.up.pt/lbaw/lbaw2223/lbaw2271/-/blob/main/a9_openapi.yaml

tags:
  - name: 'M01: Registration and authentication'
  - name: 'M02: Home page and Static pages'
  - name: 'M03: Platform administration'
  - name: 'M04: Users'
```

- name: 'M05: Auctions'
- name: 'M06: Reports'
- name: 'M07: Notifications'

## paths:

# M01: Registration and authentication

### /login:

#### get:

operationId: R101

summary: 'R101: Login Form.'

description: 'Provide login form UI. Access: PUB.'

#### tags:

- 'M01: Registration and authentication'

#### responses:

'200':

description: 'Ok. Show Log-in UI [UI02].'

#### post:

operationId: R102

summary: 'R102: Login Action.'

description: 'Processes the login form submission. Access: PUB.'

#### tags:

- 'M01: Registration and authentication'

#### requestBody:

required: true

#### content:

application/x-www-form-urlencoded:

#### schema:

type: object

#### properties:

email: # <!-- form field name

type: string

password: # <!-- form field name

type: string

#### 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: '/'

#### 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: Registration and authentication'
    responses:
      '302':
        description: 'Redirect after processing logout.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Successful logout. Redirect to Home page.'
                value: '/'
              302Error:
                description: 'Failed to logout. Redirect to Home page.'
                value: '/'

/register:
  get:
    operationId: R104
    summary: 'R104: Register Form.'
    description: 'Provide new user registration form UI. Access: PUB.'
    tags:
      - 'M01: Registration and authentication'
    responses:
      '200':
        description: 'Ok. Show Sign-Up UI [UI04].'

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

    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              name:
                type: string
              gender:
```



```

        type: string
        enum: ['M', 'F', 'NB', 'O']
    phone:
        type: string
    email:
        type: string
        format: email
    birthdate:
        type: string
        format: date
    profile_picture:
        type: string
        format: binary
        description: 'Can be jpeg, jpg, png and gif'
    address:
        type: string
    password:
        type: string
        format: password
    confirm_password:
        type: string
        format: password
    required:
        - name
        - gender
        - phone
        - email
        - birthdate
        - profile_picture
        - address
        - password
        - confirm_password
    responses:
        '302':
            description: 'Redirect after processing new user information.'
            headers:
                Location:
                    schema:
                        type: string
            examples:
                302Success:
                    description: 'Successful Registration. Redirect to user
profile.'
                    value: '/'
                302Failure:
                    description: 'Failed Registration. Redirect to Register
Form.'
                    value: '/register'

/forgot-password:
    get:
        operationId: R106
        summary: 'R106: Forgot Password page UI.'
        description: 'Provide forgot password form UI. Access: PUB.'
```

```
tags:
  - 'M01: Registration and authentication'

responses:
  '200':
    description: 'Ok. Show Forgot-Password UI [UI03].'
```

post:

```
  operationId: R107
  summary: 'R107: Forgot Password Action.'
  description: 'Sends an email to reconfigure password. Access: PUB.'
  tags:
    - 'M01: Registration and authentication'
```

requestBody:

```
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          email:
            type: string
            format: email
          required:
            - email
```

responses:

```
  '302':
    description: 'Redirect after processing user request.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Successfully sent recover password email.
Redirect to Forgot-Password page UI.'
            value: '/forgot-password'
          302Failure:
            description: 'Failed to sent recover password email.
Redirect to Forgot-Password Page UI.'
            value: '/forgot-password'
```

/reset-password/{token}:

```
get:
  operationId: R108
  summary: 'R108: Reset-Password page UI.'
  description: 'Provide Reset-Password form UI. Access: USR.'
  tags:
    - 'M01: Registration and authentication'
```

parameters:

```
- in: path
  name: token
  schema:
    type: string
  required: true

responses:
  '200':
    description: 'Ok. Show Reset-Password UI.'

post:
  operationId: R109
  summary: 'R109: Reset Password Action.'
  description: 'Resets user password. Access: USR.'
  tags:
    - 'M01: Registration and authentication'

parameters:
  - in: path
    name: token
    schema:
      type: string
    required: true

requestBody:
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          email:
            type: string
            format: email
          password:
            type: string
            format: password
          confirm_password:
            type: string
            format: password
          token:
            type: string
        required:
          - email
          - password
          - confirm_password
          - token

responses:
  '302':
    description: 'Redirect after processing user request.'
    headers:
      Location:
        schema:
```

```
        type: string
      examples:
        302Success:
          description: 'Successfully recovered password. Redirect to
login page UI [UI02].'
          value: '/login'
        302Failure:
          description: 'Failed to recover password email. Redirect
to Reset-Password Page UI.'
          value: '/reset-password/{token}'

# M02: Home page and Static pages
/:
  get:
    operationId: R201
    summary: 'R201: Home page.'
    description: 'The Home page is the center of navigation of the web
app. Access: PUB.'
    tags:
      - 'M02: Home page and Static pages'
    responses:
      '200':
        description: 'Ok. Show Home Page UI [UI01].'

/about:
  get:
    operationId: R202
    summary: 'R202: About page.'
    description: 'About Page containing info about our webpage. Access:
PUB.'
    tags:
      - 'M02: Home page and Static pages'
    responses:
      '200':
        description: 'Ok. Show about us UI [UI05].'

/services:
  get:
    operationId: R203
    summary: 'R203: Services page.'
    description: 'Services Page containing info about the services we
provide. Access: PUB.'
    tags:
      - 'M02: Home page and Static pages'
    responses:
      '200':
        description: 'Ok. Show Services Page UI [UI07].'

/faq:
  get:
    operationId: R204
    summary: 'R204: FAQ page.'
    description: 'FAQ page containing the most frequently asked
questions. Access: PUB.'
    tags:
```

```

    - 'M02: Home page and Static pages'
  responses:
    '200':
      description: 'Ok. Show FAQ UI [UI06].'

```

```

        schema:
          type: string
        examples:
          302Success:
            description: 'Successfully issued penalty. Redirect to
admin center.'
            value: '/users/{id}'
          302Failure:
            description: 'Failed to issue penalty. Redirect to admin
center.'
            value: '/users/{id}'
      '403':
        description: 'Error, user does not have access to the target
report.'
      '500':
        description: 'The server has encountered a situation it does not
know how to handle. Possibly invalid database parameters.'

    /api/report/delete/{report_id}:
      delete:
        operationId: R602
        summary: 'R602: Dismiss report.'
        description: 'Processes the request to delete a report from the
database. Access: ADM.'
        tags:
          - 'M03: Platform administration'

      parameters:
        - in: path
          name: report_id
          schema:
            type: integer
            minimum: 1
            required: true

      responses:
        '302':
          description: 'Redirect after issuing penalty.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successfully deleted report. Redirect to
admin center.'
                  value: '/users/{id}'
                302Failure:
                  description: 'Failed to delete report. Redirect to admin
center.'
                  value: '/users/{id}'
        '403':
          description: 'Error, user does not have access to the target
report.'

```

```
# M04: User
/user/{id}:
  get:
    operationId: R401
    summary: 'R401: My profile UI.'
    description: 'Show signed user profile, including personal
information, form to edit personal information, bidding activity, owned
auctions and followed auctions. Access: USR.'
    tags:
      - 'M04: Users'

    parameters:
      - in: path
        name: id
        required: true
        schema:
          type: integer
          minimum: 1

    responses:
      '200':
        description: 'Ok. Show My Profile UI [UI14]. Edit profile section
UI [UI18]. Bidding History section UI [UI20]. Followed auctions section UI
[UI17]. Owned auctions section UI [UI09]'
      '403':
        description: 'Error user does not have permissions.'

  post:
    operationId: R402
    summary: 'R402: Edit profile.'
    description: 'Process new profile information. Access: USR, ADM.'
    tags:
      - 'M04: Users'

    parameters:
      - in: path
        name: id
        required: true
        schema:
          type: integer
          minimum: 1

    requestBody:
      required: true
      content:
        multipart/form-data:
          schema:
            type: object
            properties:
              name:
                type: string
              phone:
                type: string
```

```
    email:
      type: string
      format: email
    birthdate:
      type: string
      format: date
    address:
      type: string
    profile_image:
      type: string
      format: binary
      nullable: true
      description: 'Can be jpeg, jpg, png and gif'
    password:
      type: string
      format: password
    confirm_password:
      type: string
      format: password
    required:
      - name
      - phone
      - email
      - birthdate
      - address
      - profile_image

  responses:
    '302':
      description: 'Redirect after processing new user information.'
      headers:
        Location:
          schema:
            type: string
      examples:
        302Success:
          description: 'Successful edit. Redirect to user profile.'
          value: '/user/{id}'
        302Failure:
          description: 'Failed to edit. Redirect to user Form.'
          value: '/user/{id}'
    '403':
      description: 'Error user does not have permissions.'
    '500':
      description: 'Invalid database parameters.'

/users:
  get:
    operationId: R403
    summary: 'R403: List Users.'
    description: 'Show list of users that are registered in the platform.'
  Access: PUB.
  tags:
    - 'M04: Users'
```



```
responses:
  '200':
    description: 'Ok. List users UI [UI14].'
```

**/users/{id}:**

```
get:
  operationId: R404
  summary: 'R404: View user profile.'
  description: 'This resource fetchs the user profile page including
private and public data. Depending on the role of the user, the resulting
page hides sensitive information. The authenticated user that owns the
account can see all the information. The admin can see all the information
of every user account. Every other role can only see the public
information. In the case of the admin account, it shows the admin center,
containing the admin private info, reported auctions and reported users.
This last information can only be consulted by the ADM role. Access: PUB,
USR, ADM'
  tags:
    - 'M04: Users'
```

**parameters:**

```
- in: path
  name: id
  schema:
    type: integer
    minimum: 1
    required: true
```

**responses:**

```
'200':
  description: 'Ok. Show Profile UI [UI14]. Edit profile section UI
[UI18] available for the USR and ADM. Bidding History section UI [UI20]
available for the USR and ADM. Bidding auctions section UI available for
the USR and ADM. Followed auctions section UI [UI17] available for the USR
and ADM roles. Owned auctions section UI [UI09] available for the USR, ADM
and PUB roles. Wishlist section UI [UI21] available for the USR and ADM.
Won auctions section UI available for the USR and ADM. Admin center section
UI[UI22] available for the ADM. Reported users section UI[UI25] available
for the ADM. Reported auctions section UI[UI23] available for the ADM'
```

**/api/users/delete/{id}:**

```
delete:
  operationId: R405
  summary: 'R405: Delete User.'
  description: 'Delete User profile, by deleting personal information
stored in the database. Access: USR, ADM.'
  tags:
    - 'M04: Users'
```

**parameters:**

```
- in: path
  name: id
  schema:
```

```
    type: integer
    minimum: 1
    required: true

responses:
  '204':
    description: 'User profile successfully deleted.'
  '403':
    description: 'Can be the result of 3 operations: unauthorized
access; user has running auctions; user has the highest bid on a running
auction.'
```

**/api/users/addReview:**

```
post:
  operationId: R406
  summary: 'R406: Add Review.'
  description: 'Add a five star rating review to another user. Access:
USR'
  tags:
    - 'M04: Users'

  requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            rate:
              type: integer
              minimum: 1
              maximum: 5
            user_id:
              type: integer
          required:
            - rate
            - user_id

  responses:
    '200':
      description: 'Review added successfully!'
    '403':
      description: 'You do not have permissions to add this review!'
    '500':
      description: 'You can only review each seller once!'
```

**/api/user/follow\_auction:**

```
post:
  operationId: R407
  summary: 'R407: Follow auction.'
  description: 'Proccesses request to follow auction. Access: USR'
  tags:
    - 'M04: Users'
```

```
requestBody:
  required: true
  content:
    multipart/form-data:
      schema:
        type: object
        properties:
          auction_id:
            type: integer
            minimum: 1
          user_id:
            type: integer
            minimum: 1
        required:
          - auction_id
          - user_id

responses:
  '200':
    description: 'Successfully followed target auction.'
```

/api/user/unfollow\_auction:

```
post:
  operationId: R408
  summary: 'R408: Unfollow auction.'
  description: 'Processes request to unfollow auction. Access: USR'
  tags:
    - 'M04: Users'
```

```
requestBody:
  required: true
  content:
    multipart/form-data:
      schema:
        type: object
        properties:
          auction_id:
            type: integer
            minimum: 1
          user_id:
            type: integer
            minimum: 1
        required:
          - auction_id
          - user_id

responses:
  '200':
    description: 'Successfully unfollowed target auction.'
```

/api/user/follow\_term:

```
post:
  operationId: R409
  summary: 'R409: Add term to wishlist.'
```

```
description: 'Processes request to follow term. Access: USR'
tags:
  - 'M04: Users'

requestBody:
  required: true
  content:
    multipart/form-data:
      schema:
        type: object
        properties:
          search_term:
            type: string
        required:
          - search_term

responses:
  '200':
    description: 'Successfully added term to wishlist.'
```

/api/user/unfollow\_term:

```
post:
  operationId: R410
  summary: 'R410: Remove term from wishlist.'
  description: 'Processes request to unfollow term. Access: USR'
  tags:
    - 'M04: Users'

  requestBody:
    required: true
    content:
      multipart/form-data:
        schema:
          type: object
          properties:
            search_term:
              type: string
          required:
            - search_term

  responses:
    '200':
      description: 'Successfully removed term from wishlist.'
```

/balance:

```
get:
  operationId: R411
  summary: 'R411: Balance page UI.'
  description: 'View funds of the account. Access: USR'
  tags:
    - 'M04: Users'

  responses:
    '200':
```

```
    description: 'Ok. Show Balance page UI.'
```

```
/deposit:
  get:
    operationId: R412
    summary: 'R412: Paypal redirect.'
    description: 'Fetchs paypal API for deposit. Access: USR'
    tags:
      - 'M04: Users'

    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              deposit_value:
                type: integer
                minimum: 5
                maximum: 999999999
            required:
              - deposit_value

    responses:
      '200':
        description: 'Ok. Show Paypal page.'
```

```
/deposit/cancel:
  get:
    operationId: R413
    summary: 'R413: Show Cancelled Deposit page.'
    description: 'If paypal request was cancelled, this page displays
that information. Access: USR'
    tags:
      - 'M04: Users'

    responses:
      '200':
        description: 'Ok. Show Cancelled Deposit page.'
```

```
/deposit/success:
  get:
    operationId: R414
    summary: 'R414: Paypal request validator'
    description: 'Check if the deposit operation was successfull, and
display Successfull Deposit page. Access: USR'
    tags:
      - 'M04: Users'

    responses:
      '200':
        description: 'Ok. Show Successfull Deposit page UI[UI19].'
```

```
/withdraw:
  get:
    operationId: R415
    summary: 'R415: Paypal redirect.'
    description: 'Fetchs paypal API for Withdraw. Access: USR'
    tags:
      - 'M04: Users'

    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              withdraw_value:
                type: integer
                minimum: 5
                maximum: 999999999
            required:
              - withdraw_value

    responses:
      '200':
        description: 'Ok. Show Paypal page.'
```

```
/withdraw/cancel:
  get:
    operationId: R416
    summary: 'R416: Show Cancelled Withdraw page.'
    description: 'If paypal request was cancelled, this page displays
that information. Access: USR'
    tags:
      - 'M04: Users'

    responses:
      '200':
        description: 'Ok. Show Cancelled Withdraw page.'
```

```
/withdraw/success:
  get:
    operationId: R417
    summary: 'R417: Paypal request validator'
    description: 'Check if the withdraw operation was successfull, and
display Successfull Withdraw page. Access: USR'
    tags:
      - 'M04: Users'

    responses:
      '200':
        description: 'Ok. Show Successfull Withdraw page UI UI[UI16].'
```

```
/auctions/{auction_id}:
  get:
    operationId: R501
    summary: 'R501: View Auction.'
    description: 'Show auction page. Access: PUB.'
    tags:
      - 'M05: Auctions'

    parameters:
      - in: path
        name: auction_id
        schema:
          type: integer
          minimum: 1
          required: true

    responses:
      '200':
        description: 'Ok. Show auction page UI [UI10]'
```

```
/sell:
  get:
    operationId: R502
    summary: 'R502: Create Auction page UI.'
    description: 'Fetch Create Auction Form. Access: USR.'
    tags:
      - 'M05: Auctions'

    responses:
      '200':
        description: 'Ok. Show Create Auction Form UI [UI11].'
```

```
  post:
    operationId: R503
    summary: 'R503: Create Auction action.'
    description: 'Processes the creation form submission. Access: USR.'
    tags:
      - 'M05: Auctions'

    requestBody:
      required: true
      content:
        multipart/form-data:
          schema:
            type: object
            properties:
              title:
                type: string
              description:
                type: string
              images:
                type: array
                items:
                  type: string
```

```

        format: binary
    base_price:
        type: integer
    start_date:
        type: string
        format: date-time
    end_date:
        type: string
        format: date-time
    buy_now:
        type: integer
        nullable: true
    categories:
        type: array
        items:
            type: boolean
    required:
        - title
        - description
        - images
        - base_price
        - start_date
        - end_date
        - buy_now
        - categories

responses:
    '302':
        description: 'Redirect after creating auction.'
        headers:
            Location:
                schema:
                    type: string
        examples:
            302Success:
                description: 'Successfully created auction. Redirect to
auction page.'
                value: '/auction/{id}'
            302Failure:
                description: 'Failed to create auction. Redirect create
auction page.'
                value: '/sell'
    '403':
        description: 'Error admins do not create auctions.'
    '500':
        description: 'Invalid database parameters.'

/auctions/edit/{id}:
    get:
        operationId: R504
        summary: 'R504: Edit Auction page.'
        description: 'Fetch form to edit the auction. Access: OWN, ADM'
        tags:
            - 'M05: Auctions'

```



```
parameters:
  - in: path
    name: id
    schema:
      type: integer
      minimum: 1
      required: true

responses:
  '200':
    description: 'Show edit form UI [UI12].'
```

post:

```
  operationId: R505
  summary: 'R505 Edit Auction Action.'
  description: 'Processes form in order to edit the auction. Access:
OWN, ADM.'
```

tags:

```
  - 'M05: Auctions'
```

parameters:

```
  - in: path
    name: id
    schema:
      type: integer
      minimum: 1
      required: true
```

requestBody:

```
  required: true
  content:
    multipart/form-data:
      schema:
        type: object
        properties:
          title:
            type: string
          description:
            type: string
          base_price:
            type: integer
          start_date:
            type: string
            format: date-time
          end_date:
            type: string
            format: date-time
          buy_now:
            type: integer
            nullable: true
          images:
            type: array
```

```

        items:
          type: string
          format: binary
          description: 'Can be jpeg, jpg, png and gif'
      required:
        - title
        - description
        - base_price
        - start_date
        - end_date
        - buy_now

  responses:
    '302':
      description: 'Redirect after editing auction.'
      headers:
        Location:
          schema:
            type: string
      examples:
        302Success:
          description: 'Successfully edited auction. Redirect to
auction page.'
          value: '/auction/{id}'
        302Failure:
          description: 'Failed to edit auction. Redirect edit
Auction page.'
          value: '/auction/edit/{id}'
    '403':
      description: 'Error user does not have permissions.'

/auctions/cancel/{id}:
  post:
    operationId: R506
    summary: 'R506: Cancel Auction.'
    description: 'Enables Owner to set auction state to cancelled.
Access: OWN, ADM.'
    tags:
      - 'M05: Auctions'

  parameters:
    - in: path
      name: id
      schema:
        type: integer
        minimum: 1
      required: true

  responses:
    '302':
      description: 'Redirect after cancelling the auction.'
      headers:
        Location:
          schema:

```

```
    type: string
    examples:
      302Success:
        description: 'Successfully canceled. Redirect to auction
page.'
        value: '/'
      302Failure:
        description: 'Failed to cancel. Redirect to auction
page.'
        value: '/auction/{id}'
    '403':
      description: 'Error user does not have permissions.'
```

/api/auctions/getAllBids/{auction\_id}:

```
get:
  operationId: R507
  summary: 'R507: Auctions bids'
  description: 'List the bids of a certain auction. Access: PUB'
  tags:
    - 'M05: Auctions'

  parameters:
    - in: path
      name: auction_id
      schema:
        type: integer
        minimum: 1
      required: true

  responses:
    '200':
      description: 'Successfully fetched data.'
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                bid_id:
                  type: string
                  minimum: 1
                date:
                  type: string
                  format: date-time
                amount:
                  type: integer
                  minimum: 1
                user_id:
                  type: integer
                  minimum: 1
                auction_id:
                  type: integer
                  minimum: 1
```

```
    username:
      type: string
    required:
      - bid_id
      - date
      - amount
      - user_id
      - auction_id
      - username
    example:
      - bid_id: 100
        date: '2021-07-11 00:00:00+00'
        amount: 300
        user_id: 4
        auction_id: 9
        username: Manel
      - bid_id: 32
        date: '2021-07-09 00:00:00+00'
        amount: 200
        user_id: 2
        auction_id: 1
        username: Maria
```

#### /api/auctions:

##### post:

```
  operationId: R508
  summary: 'R508: Bid on auction.'
  description: 'User bids on the auction. Access: BDR.'
  tags:
    - 'M05: Auctions'
```

##### requestBody:

```
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          user_id:
            type: integer
            minimum: 1
          amount:
            type: integer
            minimum: 1
          auction_id:
            type: integer
            minimum: 1
        required:
          - user_id
          - value
          - auction_id
```

##### responses:

```

    '302':
      description: 'Redirect after bidding on the auction.'
      headers:
        Location:
          schema:
            type: string
      examples:
        302Success:
          description: 'Successfull bid. Redirect to auction page.'
          value: '/auction/{id}'
        302Failure:
          description: 'Failed to make a bid. Redirect to auction
page.'
          value: '/auction/{id}'
    '403':
      description: 'Error user does not have permissions.'

/api/search:
  get:
    operationId: R509
    summary: 'R509: Search auctions api.'
    description: 'This resource implements a search engine used to find
specific auctions. User can search by text and filter the results based on
the auction category. If no parameters get passed, returns all auctions.
Access: PUB.'

    tags:
      - 'M05: Auctions'

    parameters:
      - in: query
        name: search
        schema:
          type: string
        required: false
        examples:
          searchQuery:
            description: 'Full text search of an item that matches
computer dell.'
            value: 'computer dell' # ?search=computer+dell
      - in: query
        name: filter[category]
        schema:
          type: array
          items:
            type: integer
        required: false
        examples:
          oneFilter:
            description: 'Getting the results of one category.'
            value: [1] # ?filter%5Bcategory%5D%5B0%5D=1
          multipleFilters:
            description: 'Getting results matching many categories id.'
            value: [1, 5, 7] # ?

```

```

filter%5Bcategory%5D%5B0%5D=1&filter%5Bcategory%5D%5B1%5D=5&filter%5Bcategory%5D%5B2%5D=7
  - in: query
    name: filter[state]
    schema:
      type: array
      items:
        type: string
    required: false
    examples:
      oneFilter:
        description: 'Getting the results of one state, in this case
Cancelled auction state.'
        value: ['Cancelled'] # ?filter%5Bstate%5D%5B0%5D=Cancelled
      multipleFilters:
        description: 'Getting results matching many states.'
        value: ['Cancelled', 'Running'] # ?
filter%5Bstate%5D%5B0%5D=Cancelled&filter%5Bstate%5D%5B1%5D=Running
  - in: query
    name: filter[maxPrice]
    schema:
      type: integer
    required: false
    examples:
      maxPrice:
        description: 'Getting the results whose current price is
lower or equal to 200.'
        value: 200 # ?filter%5BmaxPrice%5D=200
  - in: query
    name: filter[buyNow]
    schema:
      type: string
    required: false
    examples:
      buyNow:
        description: 'Getting auctions who have buy now
functionality.'
        value: 'on' # ? filter%5BbuyNow%5D=on
  - in: query
    name: order
    schema:
      type: integer
    required: false
    examples:
      HighestPrice:
        description: 'Getting auctions ordered by current price,
from highest to lowest.'
        value: 2
      LowestPrice:
        description: 'Getting auctions ordered by current price,
from lowest to highest.'
        value: 3
      MostTrusted:
        description: 'Getting auctions ordered by owner rating, from

```

```

highest to lowest.'
    value: 4

responses:
  '200':
    description: 'Successfully fetched data.'
    content:
      application/json:
        schema:
          type: array
          items:
            type: object
            properties:
              id:
                type: integer
                minimum: 1
              name:
                type: string
              description:
                type: string
              base_price:
                type: integer
              start_date:
                type: string
                format: date-time
              end_date:
                type: string
                format: date-time
              buy_now:
                type: integer
                nullable: true
              state:
                type: string
                enum: ['Cancelled', 'Running', 'To be started',
'Ended']

              auction_owner_id:
                type: integer
            required:
              - id
              - name
              - description
              - base_price
              - start_date
              - end_date
              - state
              - buy_now
              - auction_owner_id
          example:
            - id: 22
              name: Adidas shirt
              description: Adidas shirt. Good state.
              base_price: 20
              start_date: '2021-07-11 00:00:00+00'
              end_date: '2021-10-01 00:00:00+00'

```

```

        state: 'Ended'
        buy_now: 300
        auction_owner_id: 4
      - id: 45
        name: World cup Football
        description: Adidas Football. Good state.
        base_price: 100
        start_date: '2021-07-11 00:00:00+00'
        end_date: '2021-10-01 00:00:00+00'
        buy_now: null
        state: 'Running'
        auction_owner_id: 9

/search:
  get:
    operationId: R510
    summary: 'R510: Search auctions UI.'
    description: 'This resource passes the filtering parameters to the
search API, receiving and listing the results and the respective filters on
the web page. Access: PUB.'

    tags:
      - 'M05: Auctions'

    parameters:
      - in: query
        name: search
        schema:
          type: string
        required: false
        examples:
          searchQuery:
            description: 'Searching for items with computer dell in their
description.'
            value: 'computer dell' # ?search=computer+dell
      - in: query
        name: filter[category]
        schema:
          type: array
          items:
            type: integer
        required: false
        examples:
          oneFilter:
            description: 'Searching for the results of one category.'
            value: [1] # ?filter%5Bcategory%5D%5B0%5D=1
          multipleFilters:
            description: 'Searching for the results matching many
categories id.'
            value: [1, 5, 7] # ?
filter%5Bcategory%5D%5B0%5D=1&filter%5Bcategory%5D%5B1%5D=5&filter%5Bcatego
ry%5D%5B2%5D=7
      - in: query
        name: filter[state]

```



```

    schema:
      type: array
      items:
        type: string
      required: false
    examples:
      oneFilter:
        description: Searching for the results of one state, in this
case Cancelled auction state.'
        value: ['Cancelled'] # ?filter%5Bstate%5D%5B0%5D=Cancelled
      multipleFilters:
        description: 'Searching for the results matching many
states.'
        value: ['Cancelled', 'Running'] # ?
filter%5Bstate%5D%5B0%5D=Cancelled&filter%5Bstate%5D%5B1%5D=Running
- in: query
  name: filter[maxPrice]
  schema:
    type: integer
    required: false
  examples:
    maxPrice:
      description: 'Searching for the results whose current price
is lower or equal to 200.'
      value: 200 # ?filter%5BmaxPrice%5D=200
- in: query
  name: filter[buyNow]
  schema:
    type: string
    required: false
  examples:
    buyNow:
      description: 'Searching for the results who have buy now
functionality.'
      value: 'on' # ?filter%5BbuyNow%5D=on
- in: query
  name: order
  schema:
    type: integer
    required: false
  examples:
    HighestPrice:
      description: 'Searching for the auctions ordered by current
price, from highest to lowest.'
      value: 2 # ?order=2
    LowestPrice:
      description: 'Searching for the auctions ordered by current
price, from lowest to highest.'
      value: 3 # ?order=3
    MostTrusted:
      description: 'Searching for the auctions ordered by owner
rating, from highest to lowest.'
      value: 4 # ?order=4

```

```
    responses:
      '200':
        description: 'List search results UI [UI13].'
```

**/api/image/{id}:**

```
  delete:
    operationId: R511
    summary: 'R511: Delete auction image.'
    description: 'This api deletes an image that is associated with a
specific auction. Access: OWN, ADM'
    tags:
      - 'M05: Auctions'

    parameters:
      - in: path
        name: auction_id
        schema:
          type: integer
          minimum: 1
          required: true

    responses:
      '200':
        description: 'Ok. Image was successfully deleted.'
      '403':
        description: 'Error, user does not have the required
permissions.'
```

**/api/auctions/update:**

```
  post:
    operationId: R512
    summary: 'R512: Update Auctions state.'
    description: 'Handles new auction state condition. Sends
notifications regarding new states. Access: PUB'
    tags:
      - 'M05: Auctions'

    responses:
      '200':
        description: 'Successfully updated auctions.'
```

**/auctions/checkout/{id}:**

```
  get:
    operationId: R513
    summary: 'R513: Auction checkout UI.'
    description: 'Provide auction checkout form UI Access: BDR.'
    tags:
      - 'M05: Auctions'

    parameters:
      - in: path
        name: auction_id
        schema:
          type: integer
```

```
        minimum: 1
        required: true

    responses:
        '200':
            description: 'Ok. Show checkout UI.'
```

**/auctions/checkout/{auction\_id}/success:**

```
    get:
        operationId: R514
        summary: 'R514: Auction checkout success page.'
        description: 'After successfull checkout this page gets fetched,
confirming and displaying the checkout information for the user. Access:
BDR.'
        tags:
            - 'M05: Auctions'

        parameters:
            - in: path
              name: auction_id
              schema:
                  type: integer
                  minimum: 1
                  required: true

        responses:
            '200':
                description: 'Ok. Show successfull checkout page.'
```

**# M06: Reports**

**/users/report/{id}:**

```
    get:
        operationId: R601
        summary: 'R601: Get report form.'
        description: 'Provide user report form UI. Access: USR.'
        tags:
            - 'M06: Reports'

        parameters:
            - in: path
              name: reported_user_id
              schema:
                  type: integer
                  minimum: 1
                  required: true

        responses:
            '200':
                description: 'Ok. Show Report form UI.'
```

**/users/report:**

```
    post:
        operationId: R602
        summary: 'R602: Report form submission'
```

```
description: 'Processes the user report form submission. Access: USR'
tags:
  - 'M06: Reports'

requestBody:
  required: true
  content:
    multipart/form-data:
      schema:
        type: object
        properties:
          reported_id:
            type: integer
            minimum: 1
          reported_name:
            type: string
        required:
          - reported_id

responses:
  '302':
    description: 'Redirect after creating report.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Successfully created report. Redirect to
main page.'
            value: '/'
          302Failure:
            description: 'Failed to create report. Redirect to report
page.'
            value: 'users/report/{id}'
  '500':
    description: 'The server has encountered a situation it does not
know how to handle. Possibly invalid database parameters.'

/auctions/report/{id}:
  get:
    operationId: R603
    summary: 'R603: Get report form.'
    description: 'Provide auction report form UI. Access: USR.'
    tags:
      - 'M06: Reports'

parameters:
  - in: path
    name: reported_user_id
    schema:
      type: integer
      minimum: 1
    required: true
```

```

    responses:
      '200':
        description: 'Ok. Show Report form UI.'

/auctions/report:
  post:
    operationId: R604
    summary: 'R604: Report form submission'
    description: 'Processes the auction report form submission. Access:
USR'
    tags:
      - 'M06: Reports'

    requestBody:
      required: true
      content:
        multipart/form-data:
          schema:
            type: object
            properties:
              reported_id:
                type: integer
                minimum: 1
              reported_name:
                type: string
            required:
              - reported_id

    responses:
      '302':
        description: 'Redirect after creating report.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Successfully created report. Redirect to
main page.'
                value: '/'
              302Failure:
                description: 'Failed to create report. Redirect to report
page.'
                value: 'auctions/report/{id}'
      '500':
        description: 'The server has encountered a situation it does not
know how to handle. Possibly invalid database parameters.'

# 'M07: Notifications'
/api/notifications/delete/{id}:
  delete:
    operationId: R701

```

```
summary: 'R701: Delete notification'
description: 'Issues request to delete notification from database.
Access: USR'
tags:
  - 'M07: Notifications'

parameters:
  - in: path
    name: notification_id
    schema:
      type: integer
      minimum: 1
      required: true

responses:
  '200':
    description: 'Notification successfully deleted'
  '403':
    description: 'Error, user does not have the required
permissions.'

/api/notifications/get:
  post:
    operationId: R702
    summary: 'R702: Get notifications'
    description: 'Get user notifications from the database. Access: USR'
    tags:
      - 'M07: Notifications'

    responses:
      '200':
        description: 'Notification were successfully fetched.'
```

## 9. Implementation Details

### 9.1. Libraries Used

This project was implemented using just one library besides the ones already specified in the [Computational Resources](#) artifact: Font Awesome. This library was used to add some icons to the checkout and checkout\_success pages.

### 9.2 User Stories

US Identifier	Name	Module	Priority	Team Members	State
US36	View Users profile	M04: Users	High	Pedro Moreira	100%

US Identifier	Name	Module	Priority	Team Members	State
US32	Bid an auction	M05: Auctions	High	<b>André Sousa</b>	100%
US51	Update bid	M05: Auctions	High	<b>André Sousa</b>	100%
US02	Sign-up	M01: Registration and authentication	High	<b>André Sousa, Pedro Fonseca</b>	100%
US01	Sign-in	M01: Registration and authentication	High	<b>André Sousa, Pedro Fonseca</b>	100%
US37	Logout	M01: Registration and authentication	High	<b>André Sousa, Pedro Fonseca</b>	100%
US34	Edit account	M04: Users	High	<b>Pedro Moreira</b>	100%
US63	Cancel auction	M05: Auctions	High	<b>André Sousa</b>	100%
US71	Manage auctions	M03: Platform administration	High	<b>Pedro Moreira</b>	100%
US16	Search	M05: Auctions	High	<b>Vitor Cavaleiro</b>	100%
US72	Cancel auctions	M03: Platform administration	High	<b>Pedro Moreira</b>	100%
US31	Create auctions	M05: Auctions	High	<b>Pedro Fonseca</b>	100%
US21	Most active auctions	M05: Auctions	Medium	<b>André Sousa</b>	100%
US24	Social networks	M05: Auctions	Low	<b>André Sousa</b>	100%
US25	More from this seller	M05: Auctions	Low	<b>André Sousa</b>	100%
US21	Most active auctions	M05: Auctions	Medium	<b>André Sousa</b>	100%
US22	New auctions	M05: Auctions	Medium	<b>André Sousa</b>	100%
US35	Bidding history	M05: Auctions	High	<b>André Sousa</b>	100%
US14	Consult FAQ	M02: Home page and Static pages	High	<b>Pedro Fonseca</b>	100%
US11	Home Page	M02: Home page and Static pages	High	<b>Pedro Fonseca, André Sousa</b>	100%
US62	Manage auction status	M05: Auctions	High	<b>André Sousa</b>	100%
US61	Edit auctions	M05: Auctions	High	<b>André Sousa, Pedro Fonseca</b>	100%

US Identifier	Name	Module	Priority	Team Members	State
US15	Consult Contacts	M02: Home page and Static pages	High	<b>Pedro Fonseca</b>	100%
US13	Consult Services	M02: Home page and Static pages	High	<b>Pedro Fonseca</b>	100%
US33	View activity	M04: Users	High	<b>Pedro Moreira</b>	100%
US12	See About page	M02: Home page and Static pages	High	<b>Pedro Fonseca</b>	100%
US39	Delete account	M04: Users	Medium	<b>André Sousa, Pedro Moreira</b>	100%
US19	Report users	M06: Reports	Medium	<b>Pedro Moreira</b>	100%
US69	Buy it now	M05: Auctions	Medium	<b>Pedro Fonseca</b>	100%
US58	Buy now	M05: Auctions	Medium	<b>Pedro Fonseca</b>	100%
US23	Recover Password	M04: Users	Medium	<b>André Sousa</b>	100%
US17	Filters	M05: Auctions	Medium	<b>Vitor Cavaleiro</b>	100%
US57	Rate a seller	M04: Users	Medium	<b>André Sousa</b>	100%
US40	Follow auctions	M05: Auctions	Medium	<b>Pedro Moreira</b>	100%
US18	Report auctions	M06: Reports	Medium	<b>André Sousa</b>	100%
US52	Notified on new bid on participating auction	M07: Notifications	Medium	<b>André Sousa</b>	100%
US53	Notified on participating auction ended	M07: Notifications	Medium	<b>André Sousa</b>	100%
US54	Notified on participating auction winner	M07: Notifications	Medium	<b>André Sousa</b>	100%
US55	Notified on followed auction canceled	M07: Notifications	Medium	<b>André Sousa</b>	100%
US56	Notified on participating auction ending	M07: Notifications	Medium	<b>André Sousa</b>	100%



US Identifier	Name	Module	Priority	Team Members	State
US20	Featured items	M05: Auctions	Medium	<b>Pedro Moreira</b>	100%
US38	Add funds to account	M04: Users	Medium	<b>Pedro Fonseca</b>	100%
US43	Profile Picture	M04: Users	Medium	<b>André Sousa</b>	100%
US26	Order search results	M04: Users	Low	<b>Vitor Cavaleiro</b>	100%
US59	Max bid	M05: Auctions	Low	<b>Pedro Fonseca</b>	100%
US45	Withdraw funds	M04: Users	Low	<b>Pedro Fonseca</b>	100%
US42	Notification	M07: Notifications	Medium	<b>André Sousa</b>	100%
US41	Wishlist	M07: Users	Medium	<b>Pedro Fonseca</b>	100%
US74	Delete user accounts	M03: Platform administration	Medium	<b>Vitor Cavaleiro</b>	100%
US64	Notified on new bid on owned auction	M07: Notifications	Medium	<b>Pedro Moreira</b>	100%
US65	Notified on owned auction ended	M07: Notifications	Medium	<b>Pedro Moreira</b>	100%
US66	Notified on owned auction winner	M07: Notifications	Medium	<b>Pedro Moreira</b>	100%
US67	Notified on owned auction canceled	M07: Notifications	Medium	<b>Pedro Moreira</b>	100%
US68	Notified on owned auction ending	M07: Notifications	Medium	<b>Pedro Moreira</b>	100%
US75	Manage auction reports	M03: Platform administration	Low	<b>Vitor Cavaleiro</b>	100%
US73	Punish users	M03: Platform administration	Medium	<b>Vitor Cavaleiro</b>	100%

## A10: Presentation

This artifact corresponds to the presentation of the product.

### 1. Product presentation

InfinityAuctions is a cutting-edge online auction platform designed to bring users together and streamline the auction process. Whether you're an art collector, phone seller, or vintage enthusiast, our platform has something for everyone. With a user-friendly design and advanced search features, finding and bidding on the items you want has never been easier. Plus, our platform is responsive to all devices, so you can access it from your smartphone, tablet, or desktop.

In addition to making it easy to find and bid on items, InfinityAuctions also offers a range of features to help users manage their auctions and account. After creating an account, authenticated users can create new auctions, follow auctions, view their bidding history, add credit to their account, and report inappropriate users/auctions. Auction owners can edit the details of their auctions, manage their status, and cancel them if necessary. And with our team of administrators working to keep the platform safe and user-friendly through careful evaluation of user reports, you can feel confident using InfinityAuctions for all your auction needs.

URL to the product: <http://lbaw2271.lbaw.fe.up.pt>

Slides used during the presentation should be added, as a PDF file, to the group's repository and linked to here.

## 2. Video presentation

[Link of the video](#)

---

## Revision history

Changes made to the first submission:

1. No changes were made

---

GROUP2271, 03/01/2023

- André Sousa, up202005277@fe.up.pt (editor)
- Pedro Moreira, up201905429@fe.up.pt
- Pedro Fonseca, up202008307@fe.up.pt
- Vítor Cavaleiro, up202004724@edu.fe.up.pt