*AuctioX*

1. **SERVER:**

* Apache

1. **DATABASE**:

* MySQL (My Structured Query Language)

1. **TECHNOLOGIES STACK**:

* XAMPP ( stands for Cross-Platform(X), Apache(A), MariaDB(M), PHP(P) and Perl(P) )

1. **SCRIPTING LANGUAGE**:

* BackEnd - PHP (Hypertext Preprocessor), JavaScript
* FrontEnd - HTML(Hyper Text Markup Language), CSS(Cascading Style Sheets)

1. **ARCHITECTURE**: MVC (Model View Controller)

* The **model** is the central component of the pattern. It expresses the application's behavior in terms of the problem domain, independent of the user interface. It directly manages the data, logic and rules of the application.
* A **view** can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
* The **controller** accepts input and converts it to commands for the model or view.

1. **HASH FUNCTION**: MD5

* Description:
* The md5() function calculates the MD5 hash of a string. The md5() function uses the RSA Data Security, Inc. MD5 Message-Digest Algorithm. We will use it to hash passwords submitted by users.

1. **SQL INJECTION**

* Description:
* SQL injection is one of the most common web hacking techniques. SQL injection is the placement of malicious code in SQL statements, via web page input.
* Prevention:
* The problem with SQL injection is, that a user input is used as part of the SQL statement. By using prepared statements you can force the user input to be handled as the content of a parameter (and not as a part of the SQL command).

1. **TYPES OF USERS**:

* Guest:
* can view the website, without being able to interact with it
* can login / register
* Authenticated:
* can do everything a guest can do
* can see the feed of auctions
* can bid for items and, if desired, buy them
* can add a new item for sale through a form
* can change its user settings
* Administrator:
* can do everything a authenticated user can do
* can block / delete accounts
* can end / delete an auction if it violates the regulations
* can assign someone to be an administrator

1. **PAYMENT OPTION**:

* PayPal / credit card

1. **PDF Generation API**:

* FPDF is a PHP class which allows to generate PDF files with pure PHP, that is to say without using the PDFlib library. F from FPDF stands for Free: you may use it for any kind of usage and modify it to suit your needs.

1. **XML Generation API:**

* The DOM extension allows you to operate on XML documents through the DOM API with PHP.

1. **JSON Generation API:**

* string **json\_encode** ( [mixed](http://php.net/manual/en/language.pseudo-types.php#language.types.mixed) $value [, int $options = 0 [, int $depth = 512 ]] )
* Returns a string containing the JSON representation of the supplied value.

1. **TASKS**

* Haloca Dorin:
  + login page
  + query the users table
  + create session cookies
  + register page
    - insert the data in the users table
    - hash password
  + checkout page
    - query the products table
    - redirect to PayPal
    - form validation
  + create Model-View-Controller frame
  + export auctions in xml, json and pdf formats
* Tiron Adrian:
  + header of the website
  + logo
  + search bar
  + login / register buttons
  + navigation menu
  + currencies (EUR / USD)
  + user settings page
  + user profile (basic info)
  + security settings (change password / email)
  + payment options (add / delete credit cards)
  + adresses (modify / delete)
    - review items before checkout page
  + image / name / details / quantity / unit price
  + select which items to buy
  + checkout button
    - Sitemap page
    - footer of the website
* useful links and trademark acknowledgement
  + finish uploaded products page
  + contact, about and shipping pages stylized
  + functionalities for My Items
* Tab buttons
* Won items feed w/ checkout button
* Uploaded items feed w/ upload form
* Templates for product wrapper and product page
* Various database changes
* Functionality for ‘Bid’ button in products feed
* Mancas Mihai:
  + products page
* provide a page with a list of products selected after some criteria, the status of products, current price, small description, time remained until the offert is no longer availble
* product page
* detailed description of the product including a slider with photos uploaded by user, full description, details
  + uploaded products page
* lists all the products that the user(with account) has uploaded until prezent
* a form where the user can post his item
  + won auction products page
    - lists all the products that the user (with account) won in auction
    - real-time display of products detained at any moment (TODO)
  + add an item for bidding form (TODO)
  + user’s items put up for auctioning page (TODO)
  + form for generating a RSS feed page(TODO)
* Manolache Mihaita:
  + - database
    - advanced search page:
    - after searching a item you will be redirected to the advanced search page where you will have the products listed.
    - you will be able to use the check boxes so you can search a more specific item
    - some of the criterias are price, time until expiration, time since the object has been posted and so on
    - the advanced search is based on the tags (these are declared by the user at the creation of the object and/or modified by the admins before the form is accepted and placed in the data base)
    - after a field is checked the page will autocratically be reloaded and display the new items
    - drop-down links in nav menu
    - about / contact pages:
* contact information for the site

1. **USER PERSONAS**

* Introduction
* There are two types of personas: the buyer and the seller.
* The seller persona is the person that is willing to put products up for sale because of the money and the riddance of said objects.
* The buyer persona is the person who will use the products by buying them to solve their problems.
* Sometimes the buyer and the seller can be one and the same. For example, if you buy a cellular phone and put up your old one for sale, you are both the buyer and the seller.

(SEE PERSONAS AT THE END OF THIS DOCUMENT)

1. **FRONTEND DESING**

* For the front-end design we used HTML (HyperText Markup Language) whose purpose is to provide a set of general rules that suggest how content should look when rendered. A markup language doesn't style the website, for that CSS (Cascading Style Sheets) is used.
* The "cascading" in Cascading Style Sheets refers to how property values are applied in the context of the parent/child hierarchy of the Web document. Child elements either inherit or override property values bound to their parent elements. A style sheet is the encapsulation of style rules in a centralized location, either in the head section of the HTML document or in a separate linked file. The Web browser reads these styles and applies the specified formatting rules before displaying the content.

1. **BACKEND DESIGN**

* The back-end is the code that runs on the server, that receives requests from the clients, and contains the logic to send the appropriate data back to the client. The back-end also includes the database, which will persistently store all of the data for the application. Building a new backend depends on the language we are most familiar with like, so we will use PHP, that will run on Apache. The second thing we need to do is design the database schema and pick the database engine that suits our needs. We consider that MySQL(that XAMPP uses) is more than enough for what we need to use it for.
* First off, the website pages will be structured as follows: each main page will include the header (which will be in a separate .php file). The main page will have a feed of products that the user can scroll through indefinitely and bid for. The header provides login / register functionalities and a search bar (which also has an advanced search button for more categories and filters to choose). The user can navigate through the pages using the menu, for example, the user account, which contains information and settings that can be made on the account. The user can also see its won items through bidding and its added items for sale (can also add items through a form). If the user decides so, it can buy the won products through a checkout page where payment info is introduced. The specials / last minute are pages where specific types of items appear.
* For creating the RSS the client needs to complete a form where he specify what kind of products he wants a feed for or some HTML identifiers from the page generating the feed. Also it is allowed to specify the number of desired "objects" in the feed containing the name of the product, link to actual page and a small description.

