AuctioX

1. Server: Apache
2. Database: MySQL (My Structured Query Language)
3. Technologies stack:
   1. XAMPP ( stands for Cross-Platform(X), Apache(A), MariaDB(M), PHP(P) and Perl(P) )
4. Scripting language: PHP (Hypertext Preprocessor)
5. Architecture: MVC (Model View Controller)
   1. 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.
   2. 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.
   3. The **controller** accepts input and converts it to commands for the model or view.
6. Hash function: MD5
   1. Description:

The md5() function calculates the MD5 hash of a string. The md5() function uses the RSA Data Security, Inc. MD5 Message-Digest Algorithm.

1. SQL Injection
   1. 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.

* 1. Prevention:

What *mysql\_real\_escape\_string()* does is take a string that is going to be used in a MySQL query and return the same string with all SQL Injection attempts safely escaped. Basically, it will replace those troublesome quotes(') a user might enter with a MySQL-safe substitute, an escaped quote \'.

1. Types of users:
   1. Guest:

- can access the website

- can login / register

* 1. 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

* 1. 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
2. PDF 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. Tasks
   1. 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

* 1. Tiron Adrian:

- header of the website

- logo

- search bar

- login / register buttons

- navigation menu

- currencies (RON / 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 / quantity / unit price / total

- select which items to buy / delete

- checkout and delete buttons

- final total price in chosen currency

- footer of the website

- useful links and trademark acknowledgement

* 1. Mancas Mihai:

- product feed page

- active / inactive

- current price

- description

- time remaining

- bid button

- add an item for bidding form (TODO)

- user’s items put up for auctioning page (TODO)

* 1. Manolache Mihaita:

- advanced search page

- drop-down links in nav menu

- about / contact pages

1. USER PERSONAS
   1. 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 FURTHER AHEAD)

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) 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.



