

Auctions Website
Web Development Technologies

Georgios Nikolaou
sdi1800134

Nefeli Tavoulari
sdi1800190

Spring Semester 2022



HELLENIC REPUBLIC
National and Kapodistrian
University of Athens

Contents

1	Introduction	3
2	Installation	3
3	Database	3
4	HTTPS protocol	4
5	Authentication	4
6	Newsletter	4
7	Auctions	4
8	Administration	5
9	Messaging - Notifications	5
10	OpenStreetMap	5
11	Filters	5
12	Image Upload - File Download	5
13	Search Engine	6
14	Recommender System	6
15	Challenges	6
16	Conclusion	7
17	Screenshots	7

1 Introduction

In this assignment we implemented an auctions website, an Ebay clone, called BidIt. In this website, one can create an auction, browse auctions and bid. The main technologies used are:

- React Framework for the frontend
- Express - Node.js for the backend (REST API)
- PostgreSQL as the relational database

We chose the specific technologies as they are among the most popular ones in the field, they are well documented and offer, of course, many possibilities. We created a RESTful API using HTTP requests and JSON for data interchange between frontend and backend, in order to ensure component independence and flexibility, which we tested with Postman. In the following sections we will discuss in more detail all the challenges we faced and the services we implemented.

2 Installation

- `sudo service postgresql start` : to start the db
- copy and paste the queries found in `api/database.sql` into the Postgres console, in order to initialize the database, create the tables etc.
- check the `api/.env` file, so that these configurations match those of the host database.
- `cd api; npm install; npm start` : to install packages and start backend
- `cd ui; npm install; npm start` : to install packages and start frontend
- create a user with "admin" username to have access to the admin pages
- go to `https://localhost:5000/xml` to fill the database with data from the given XML files.

3 Database

The PostgreSQL database we used consists of the following tables:

- `account` : used to login/register/get users/get user etc
- `auction`, `auction_item` : used to create/update/delete/see auction(s) etc
- `auction_category` : intermediate table because of many to many relationship between auction and category
- `bid` : used to create bid/get bids

- category : used to create/delete/get category/categories
- message : used to send message/get inbox/sent messages
- newsletter : used to add an email
- product_view : used to add the auctions a user has viewed, used for the recommender system
- recommendation_bid : recommendations based on user's bids
- recommendation_view : recommendations based on product views

The tables' fields can be seen in detail in `api/database.sql`.

4 HTTPS protocol

The website uses SSL/TLS certificates to ensure security of the transactions.

5 Authentication

The user can login/register in any page of the website opening the modal popup that is found in the navigation bar, in order to be able to bid or sell. When the user signs up, the password they selected is encrypted and saved in the database with the other data and a JSON Web Token is generated that is saved in the session and which is deleted when the user logs out. We tried to validate as much as we could the input of the user in the all the forms and on error we send an appropriate message to the user.

6 Newsletter

We created a simple Newsletter form, where a user can add their email to the database in order to receive emails on new auctions etc.

7 Auctions

The user can create an auction by filling the corresponding form, specifying the auction details. They can also edit an auction before it has been started, they can delete it, they can view it, they can start it etc. All started and non terminated auctions are displayed and can be accessed by all users, even non authenticated ones.

8 Administration

On the installation of the app an admin user is created, who can see all users' data, approve or delete a user registration and download in json or xml format all the auctions of the website. Plus, the admin user pressing a button in the users' page, can trigger the recommender system calculations. Only the specific user has access to the corresponding endpoints.

9 Messaging - Notifications

Using conditional rendering and performing queries to the database we get the inbox/sent/new message page. When a user receives a message it is displayed right away in the inbox, since `setInterval` is used to refresh the page occasionally. Notification for new messages is shown in all the pages until the user goes to their inbox. For this, we store in the account table the number of messages they have and we compare it to the new number of received messages, in order to show notifications. Plus, when a user wins an auction, a message is sent to them automatically by the seller.

10 OpenStreetMap

For this task we do a GET call to the OpenStreetMap API using the seller's address and city data, in order to get the coordinates of the auction. The we use these coordinates in combination with the React Leaflet library to see this location in the map.

11 Filters

In the page where users browse through the auctions, we have added some filters, so that the user can see what they really want. So, we filter the auctions, using some db queries, according to the category, price and location. For these we used checkboxes, dropdown menu and a range slider in the user interface, where all the possible options are shown from the backend. Every time the user changes the state of one of these, a new 'select' query is created. When the user chooses many filters, the auctions shown should satisfy them all at the same time(intersection).

12 Image Upload - File Download

For the image upload we used React Dropzone in the frontend, along with `FormData` and `multer` in the backend, in order to upload the file. The url of the file is stored in the database, so that it can be easily retrieved when needed. For the auctions export in XML format we used the `exportFromJSON` function with

the appropriate arguments. The admin just presses the corresponding button and downloads a file with all the auctions of the website.

13 Search Engine

We created a simple search engine that helps display the auctions that match a specific description, using Postgres Full-Text Search. First of all, we tokenized and performed stemming on the description of each auction and added the result to a new column. We then created a new column with the Generalized Inverted Index and so we were able to perform queries using the user input.

14 Recommender System

We created two recommender systems, one based on users' bids and one based on users' product views, which are created when the administrator presses a button in the Users Administration Page. First of all, we create a matrix using the information given by the "Bid" table, the account ids of the bidders and the corresponding auction id of each transaction. Then, two random matrices are generated that correspond to the user features and the auction item features. Then, finding the squared distance between the prediction and the original matrix, and minimizing the RMSE, we get the final two matrices and multiplying them we get the recommended auctions for each user. We get the 5 auctions with the highest scores and display them to the user's home page. If a user has no history of bids, the recommended auctions are based on their product view history. If they have not viewed any items either, then some random products are displayed.

15 Challenges

The most challenging parts of this project were:

- inserting the data from the given xml to the website's database, since the size of them was quite large and we had to perform many queries and process the given data
- finding the right library/way to perform some more advanced tasks, such as creating the map, uploading images, exporting json/xml
- performing more advanced queries to filter the auctions shown
- creating the search engine
- restricting access in pages, hide elements from non authenticated users
- getting to know react hooks, handlers, asynchronous functions and conditional rendering

- creating the recommender system algorithm, based on either the user's bids or product views and displaying the result in the user interface. Plenty of requests had to be made and many parameters had to be taken into consideration, thus the procedure was rather complicated.

16 Conclusion

During this assignment, we had the opportunity to do research on plenty of concepts and technologies. We familiarized with web development, coming across realistic scenarios and finding solutions using different libraries and frameworks. All in all, we managed to create a user-friendly website with all the basic and some more advanced functionalities of such an application.

17 Screenshots

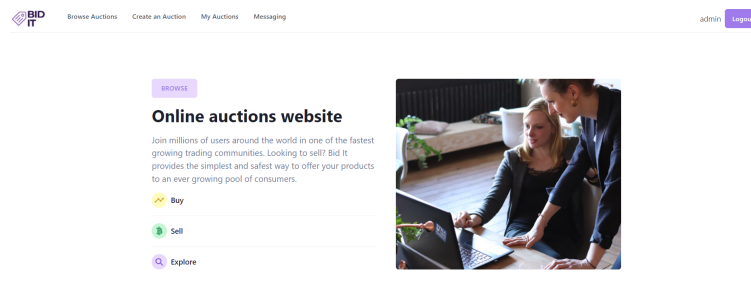


Figure 1: Home Page

Sign up

to enjoy all of our cool features 🙌

Username *

Email address *

First Name *

Last Name *

Country *

City *

Phone number *

Address *

Postcode *

Tax Code *

Password *

Confirm password *

☒ I have read and accept the terms and conditions. *

Sign up

Figure 2: Registration Form

Account created successfully

Your registration has been completed. Now all you have to do is wait for our administrators to verify your information and you can start participating in the Bid It Community. This process is usually very fast, but be sure to browse our expansive catalog while you wait.

[Browse](#)



Figure 3: Waiting Room

+ New		Inbox			
Inbox		SENDER	SUBJECT	DATE	
Sent		rulabula	money	2022-05-15T14:47:23.011Z	Delete Open
		mwire	Auction Won	2022-05-15T14:53:14.333Z	Delete Open

Figure 4: Messaging

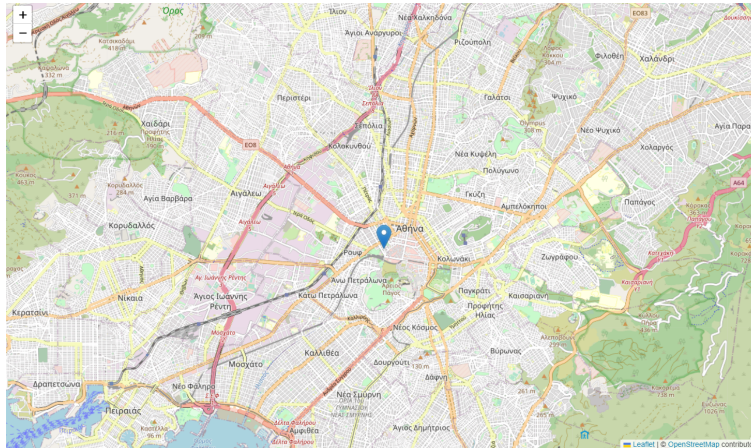
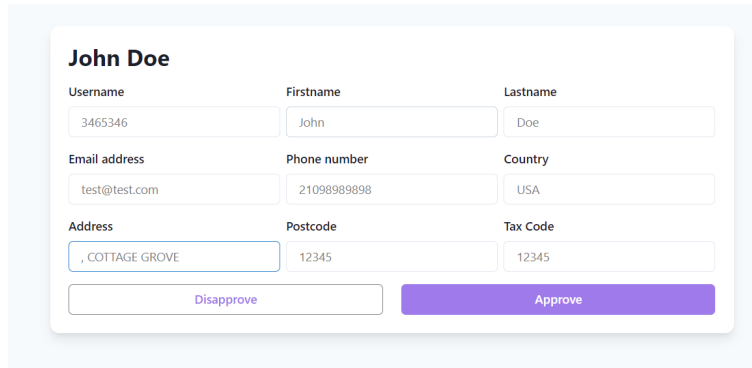


Figure 5: Map

[Recommender System](#)
[All auctions in JSON](#)
[All auctions in XML](#)

Users			
#	USERNAME	APPROVED	
0	3465346	✓	Details →
1	1548349	✓	Details →
2	rulabula	✓	Details →
3	123057	✓	Details →
4	goldcoastvideo	✓	Details →

Figure 6: Users Administration Page



John Doe

Username	Firstname	Lastname
3465346	John	Doe
Email address	Phone number	Country
test@test.com	21098989898	USA
Address	Postcode	Tax Code
, COTTAGE GROVE	12345	12345

Figure 7: User Administration



Search

Category

- ☐ Collectibles
- ☐ Clothing & Accessories
- ☐ Movies & Television
- ☐ Sports
- ☐ Decorative & Holiday
- ☐ Autographs, Paper & Writing
- ☐ Housewares & Kitchenware
- ☐ Pop Culture
- ☐ Art, Animation & Photo Images
- ☐ Miscellaneous (Antiques & More)

Price

750 750

Location

Location



LISTED BY SKEVCO

CINDERELLA 1984 CANADA GLASS (COKE-MCDONALDS)

\$5.95



LISTED BY GRANNYGRUNTIZ

FELIX the CAT Magnets

\$9.99



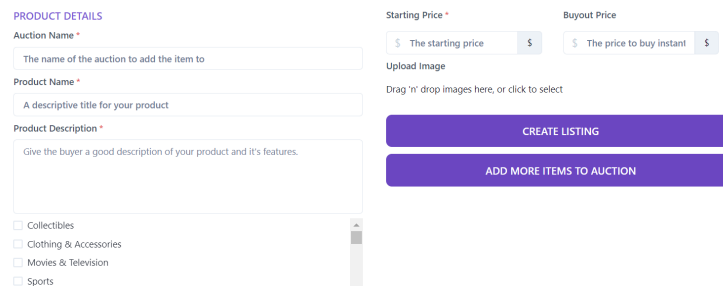
LISTED BY RIJABULIA

christopher radko | fritz n. frosty sledding

\$30.00

Figure 8: Browse Auctions, Filters, Search Bar

Create a listing



PRODUCT DETAILS

Auction Name *

The name of the auction to add the item to

Product Name *

A descriptive title for your product

Product Description *

Give the buyer a good description of your product and it's features.

☐ Collectibles
☐ Clothing & Accessories
☐ Movies & Television
☐ Sports

Starting Price * Buyout Price
 \$ The starting price \$ The price to buy instant

Upload image

Drag 'n' drop images here, or click to select

Figure 9: Auction Creation Form

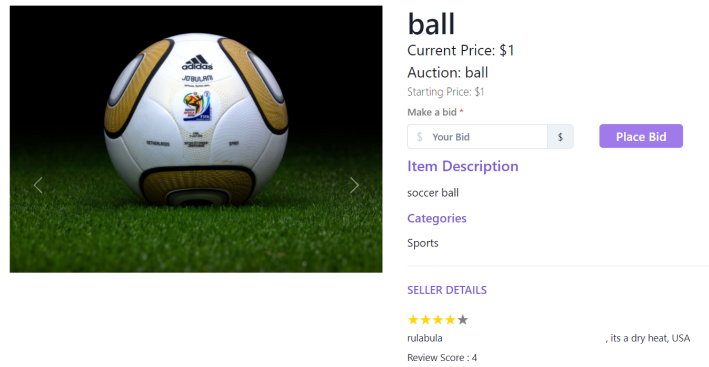


Figure 10: Auction View

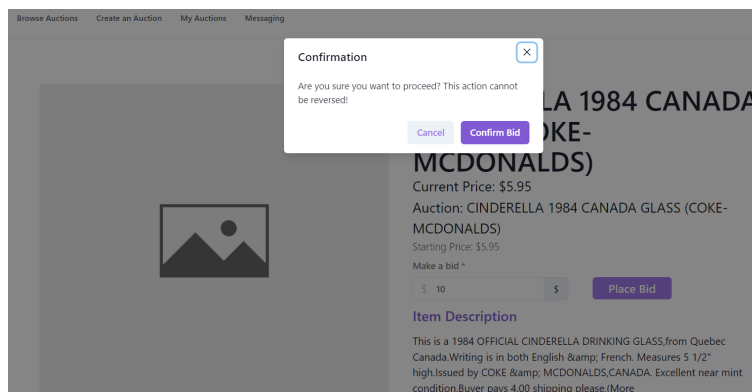


Figure 11: Bid Confirmation

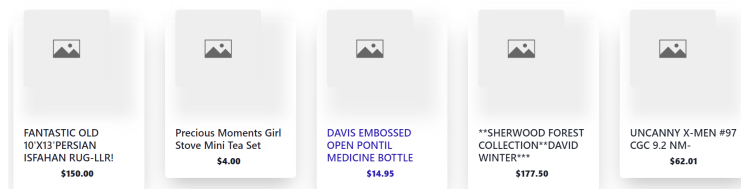


Figure 12: Recommended auctions in user's Home Page