

IIIT-H AUCTION – Hack Daddys

How many times have you seen posts on the lost and found group about some item for sale? How many purchases or sales have you made?

For the average IIITian, who barely pays any attention to the lost and found group, the IIITH auction site is a great opportunity to get great deals on second hand articles owned by fellow IIITians.

Why do we need the auction site?

In the absence of a proper price scheme, sellers often price articles at levels favorable to them, which most buyers will not be willing to pay. Having an auction in place of the current lost and found group ensures the sale of the article at a cost favorable to both the buyer and the seller. It also helps reduce waste, and unsold items are usually trashed.

How will it work?

It's surprisingly simple! Anyone with an account on the site can either bid for articles, or put items up on the block. Articles that are put up for auctioning will remain open to bids for a maximum time frame of 7 days, after which the highest bidder and the seller will be put in contact.

How will the bidding procedure work?

Once an article is put up to be auctioned, there is a 3-day buffer period in which the seller has to provide proof of quality of the product. After this 3-day period, the auction for the article goes live. During this period, anyone is free to make a bid. The only requirement for the bid is that it is a minimum of the previous bid added to the minimum increment defined for the article (by the admins of the website). However, during the last day of the bidding process, bidding will be locked to only those who have made a bid in the previous 24 hours to prevent bid spamming before the auction closes.

How do I put an article up for bidding?

After making an account, navigate to the add an article page and follow the instructions. Provide all relevant details like extent of use, MRP of the item (will be used in determining minimum increment by the admins), and expected lifetime etc. Also provide proof that the article is in the condition specified within 3 days of listing it for the auction.

How will I be ensured of a decent price?

As a seller, you can set a minimum price that you will be satisfied with. If the highest bid during the auction falls below this price, then you are free to cancel the transaction.

What is the guarantee that the delivery of the article will be made?

The payment will be routed through the auction site, and it will act as an escrow till the delivery of the article is made. The seller has a 3-day window to make the delivery, and if the delivery is cancelled for any reason, your money will be refunded after the 3-day period.

Technical details

- **Front-end Framework – ReactJs , react-bootstrap**

The whole web-app is based on a basic React framework. It has the main Js file called **index.js** whose only purpose is to render the `<App/>` instance in the root div in **index.html**.

The `<App/>` instance renders all the html content present in the App class render function into the **App.js**. Similarly Home.js , About.js etc.. all are rendered in the App class.

- **Linking pages – React-Router-DOM**

To link all the pages together, App.js has been used as the central routing point. All the content in the **Switch** tag inside the BrowserRouter tag links to the different pages. Basically the react-router makes it easier as instead of going to different links , different html content is rendered on the same page when the user clicks on the link.

- **Styling – Bootstrap4 , react-bootstrap**

For the css part of bootstrap , bootstrap 4 is used and for the Js part react-bootstrap was used.

- **Screenshots**

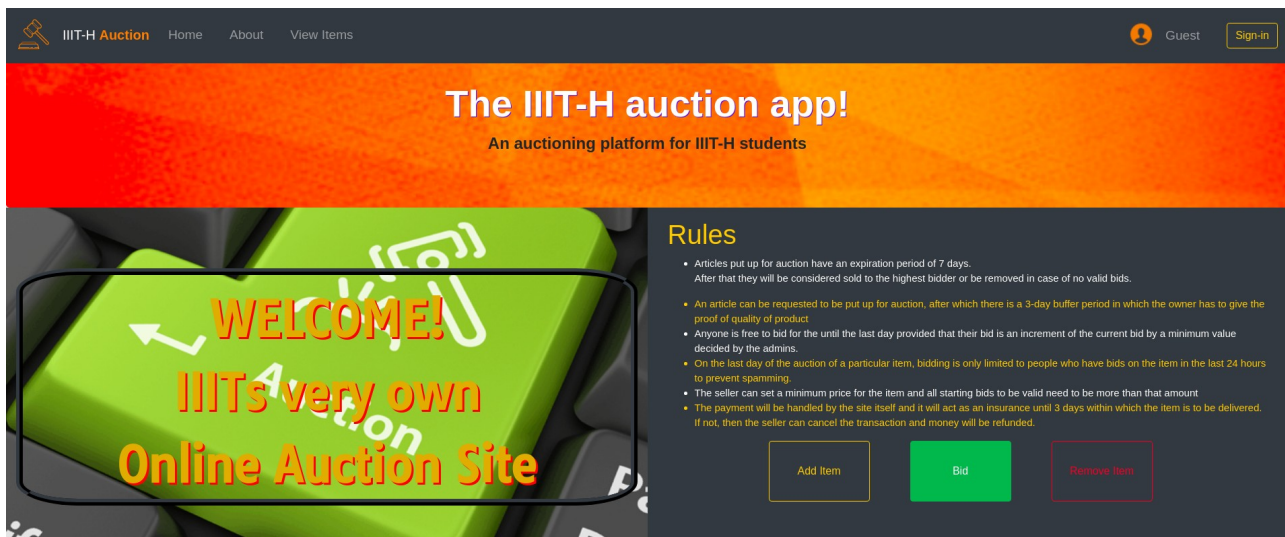
```
src > JS App.js > ...
1  import React, { Component } from 'react';
2  import { BrowserRouter, Route, Switch } from 'react-router-dom';
3
4  import Home from './components/Home';
5  import About from './components/About';
6  import Error from './components/Error';
7  import Nav1 from './components/Nav';
8
9  class App extends Component {
10   render() {
11     return (
12       <BrowserRouter>
13         <div>
14           <Nav1/>
15           <Switch>
16             <Route path="/" component={Home} exact/>
17             <Route path="/about" component={About}/>
18             <Route component={Error}/>
19           </Switch>
20         </div>
21       </BrowserRouter>
22     );
23   }
24 }
25
26 export default App;
```

App.js

Hack-Daddys > public > index.html > html > head > link

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/Logo.png" />
6     <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Tauri" />
7     <meta name="viewport" content="width=device-width, initial-scale=1" />
8     <meta name="theme-color" content="#000000" />
9     <meta
10       name="description"
11       content="Web site created using create-react-app"
12     />
13     <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
14     <!--
15       manifest.json provides metadata used when your web app is installed on a
16       user's mobile device or desktop. See https://developers.google.com/web/fund
17     -->
18     <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
19     <!--
20       Notice the use of %PUBLIC_URL% in the tags above.
21       It will be replaced with the URL of the `public` folder during the build.
22       Only files inside the `public` folder can be referenced from the HTML.
23
24       Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
25       work correctly both with client-side routing and a non-root public URL.
26       Learn how to configure a non-root public URL by running `npm run build`.
27     -->
28     <title>IIITH-Auction</title>
29   </head>
30   <body>
31     <noscript>You need to enable JavaScript to run this app.</noscript>
32     <div id="root"></div>
33     <!--
34       This HTML file is a template.
35       If you open it directly in the browser, you will see an empty page.
36
37       You can add webfonts, meta tags, or analytics to this file.
38       The build step will place the bundled scripts into the <body> tag.
39
40       To begin the development, run `npm start` or `yarn start`.
41       To create a production bundle, use `npm run build` or `yarn build`.
42     -->
43   </body>
44 </html>
```

index.html



Home page screenshot

Additions in future patches

- Adding user ratings for sellers and buyers based on their payment and delivery records (and possibly even buyer/seller reviews).
- Blacklist users who repeatedly fail to pay/deliver articles.
- Add certain functionality to detect genuine buyers, and scout for users who bump prices.
- To add different categories to the listed articles, to help buyers find articles they desire easily
- To add a point/bonus system for users with a high rating.
- Adding an automatic email subscription update service, that will alert users of higher bids on articles they're interested in
- Add a follow option to be notified when an article matching your request goes live
- Add an absentee bidding option, which is automatic. It matches the highest bid until a limit specified by you
- Add an article consignment service, complete with article valuation
- To streamline the escrow and payment process, and remove the need for an admin to physically ensure the validity of the delivery/sale.
- Proper bid verification through bank servers
- Inclusion of other local colleges
- An option to directly buy the product(not auctioned) from an external website (amazon)

WIN BIG, ANYTIME, ANYWHERE.
HEMANTH BIG GAE.