# CS 6314.501

## Web Programming Languages - F16

*Project Report*


**Submitted by:**

**Himanshu Kandwal (hxk154230)**

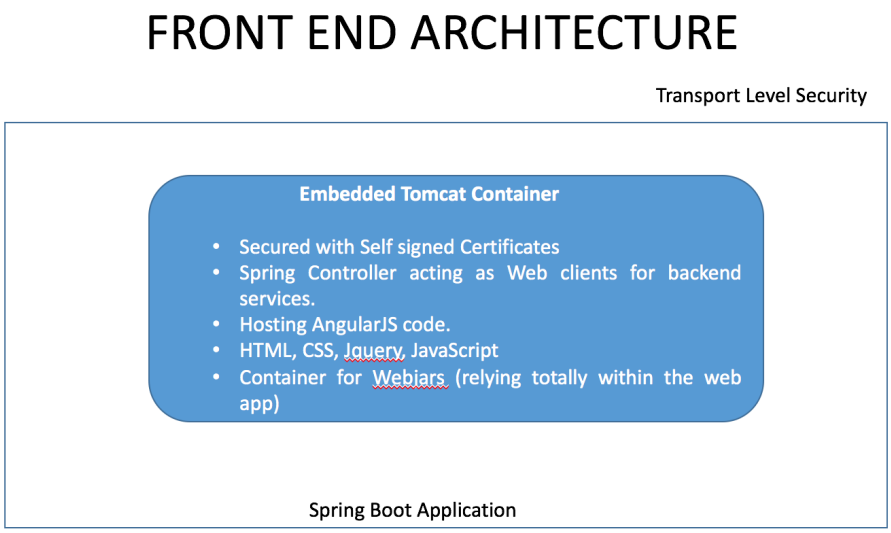
**Anirudh Kuttiyil Valsalan (axk153230)**

**Srinivas Lingamgunta (sxl154630)**

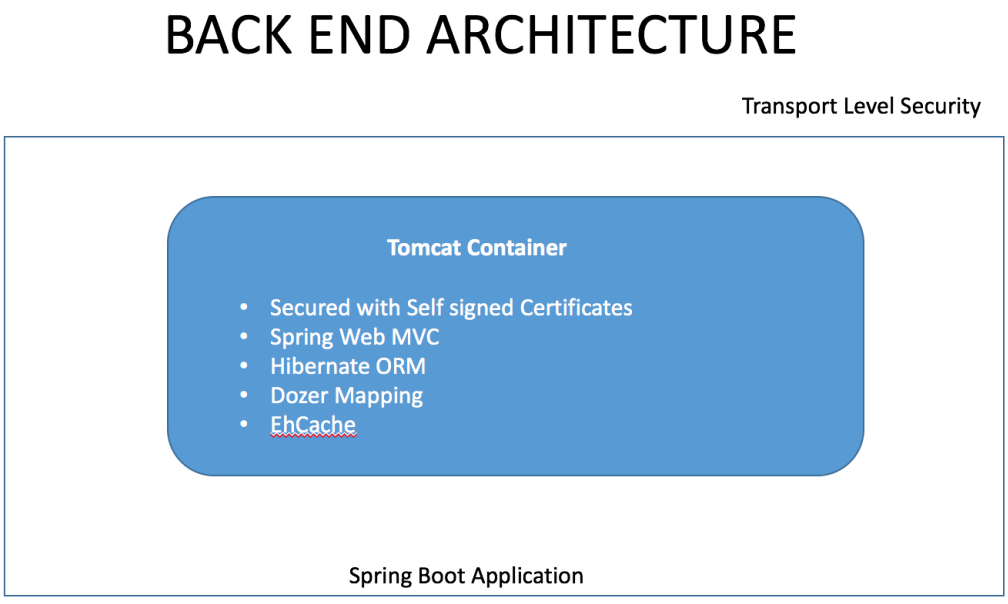
## Objective:

The objective of this project is to develop a web application that can be used for posting a need of apartment to rent rental and prospective bidders will bid for your requirement. The users in the context of this application have role as a customer using the application.

## Architectural Design:

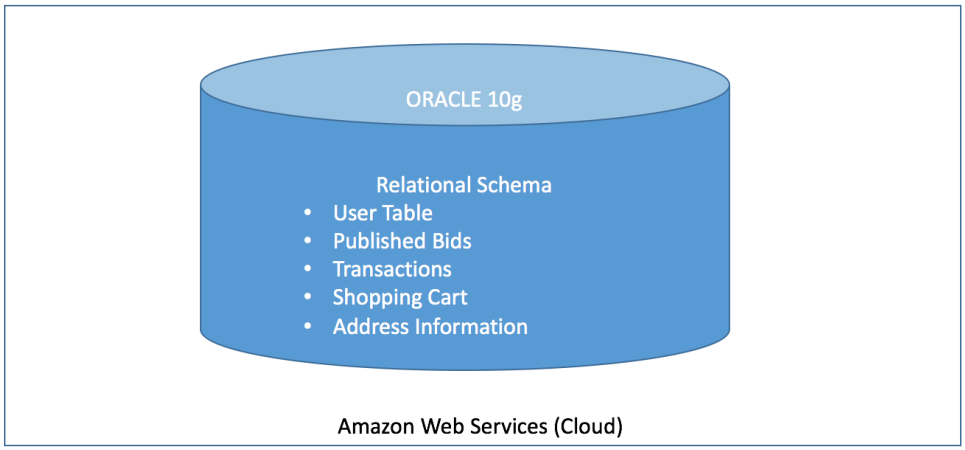



Website is the first point of contact for the user. A user who wishes to use the web application and access all the functionalities offered by the application should do it by using a browser. The user is first displayed with the home page of the website which has an image slider and navigation bar. The user is allowed to navigate through the site only after a valid login or registering as a new user and then logging in with the previously provided credentials.



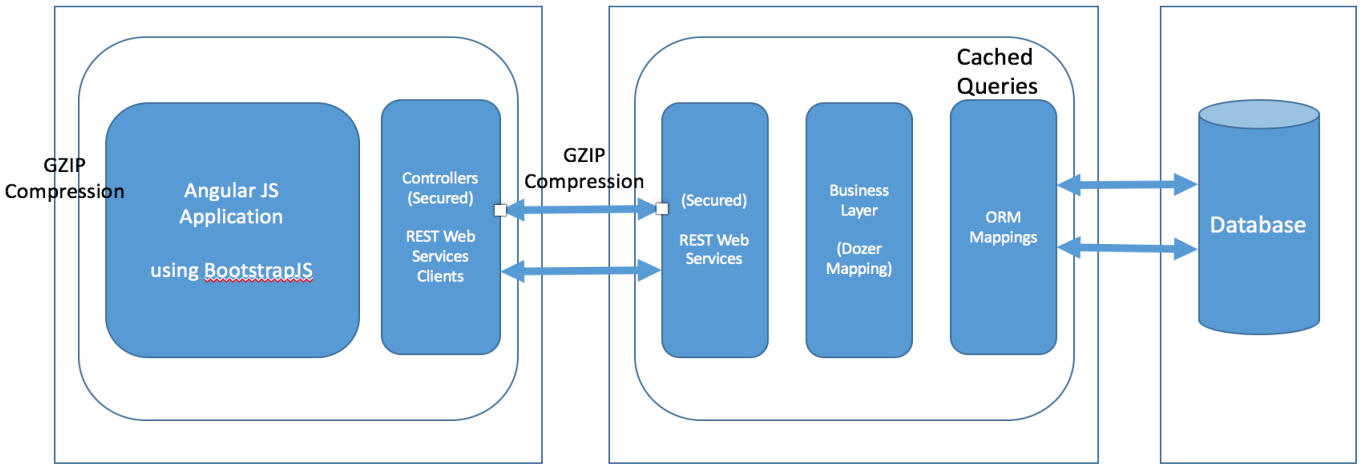
The Database architecture is as follows:

# DATABASE ARCHITECTURE




The complete overview of the application can be seen as:

# COMPLETE ARCHITECTURE




## Technical Design:

We have two tomcat servers running. Any request by the user will be redirected to that particular **servlet** which handles that request. These servlets are resided in the same server as the website.

This rest service resides on another tomcat server(app server). All the necessary database access is done within this rest service and the results are sent back to the servlet that called the service which handles the output and calls the necessary pages to be displayed.

Over all the technologies used were:

- Bootstrap
- HTML
- CSS
- JavaScript
- Spring-Boot
- Angular JS
- Hibernate
- Java

We have also implemented the following required functionalities:

1. **SSL encryption**:
a. We enabled SSL in Apache Tomcat by using the following configuration:
b. We used keytool for generating the keystore file by using a keystorepass.
c. After adding the configuration, we can access the website by using https protocol

**2. Compression:**
**a.** We used compression for the requests between servers. We also used this for webservices.
**b.** Configuration details:

**3. Session based access for API's**
**a.** In website every page is protected from access by using a session Key.
**b.** Initially when user will login to the website, a session Key is generated by the login service.
**c.** If user logs out of the website, the session key will be deleted.

**WebServices:**

Web services are hosted on another tomcat server. These receive requests from the webpage. Web services access data from the Database using hibernate and send back relevant data to the user when user requests them. Each major functionality like new user registration, search, login etc are implemented as an individual service.
**Technologies:**
We used jersey framework for implementing web services. Following are the technologies used on the server side:
1. Jackson FasterXML
2. REST Web Services for Java.

**Features:**
In this section the functionalities of few of the implemented web services:
**1. Login Service:**

Authentication of the user provided credentials are done here. In this service the user's last login will be updated and in case of failed attempt, the number of failed attempts will also be

updated. Cache is implemented here. As soon as a request for login is made to the login service, the Cache is checked to see whether the user email is present. In case the user email is already present, then it is a hit. Otherwise it is a miss. In case of a miss, the database is accessed and the retrieved username and password are entered into the cache. Once the user is authenticated, the database is accessed to update the last login. Also, in case of failure, the database is accessed in order to update the number of failed login attempts.

**2. Registration Service:**

Registration service registers the user to the website with their email as the identity. Mandatory fields include firstname, lastname, password and Email. These details are stored to the database.

### 3. Bidding Service:

Bidding service provides the user with the capacity to post a requirement. It asks for information about how much quantity, From Date, To Date, Price and apartment type. Once the bid is being posted, it will be visible for all the users including the users who are also currently logged into the system. The user who posts a bid will see a confirmation message saying the bid is posted. The user who posted the bid can be contacted by any person who is bidding for it and will be shown the messages sent by that bidder.

### 4. Search Bids:

The user will be displayed with all the bids posted. Further it provides you with the functionality for searching among already posted bids, and you can search the displayed bids to see a bid posted by a particular user, price, posted date and also Bid ID. Bid Detail can also be seen with a button provided for the same. **Active Bids** is a special functionality provided to the user which comprises of those bids which have their To-date not yet expired.

### 5. Checkout:

The users can add the bids of their choice to their shopping cart. A confirmation with a pop-up will be shown that the item was added to their shopping cart. Once they are done with their shopping, they can use the check out functionality which helps the user to check out the bids and thereafter the database is updated.

### 6. My Bids:

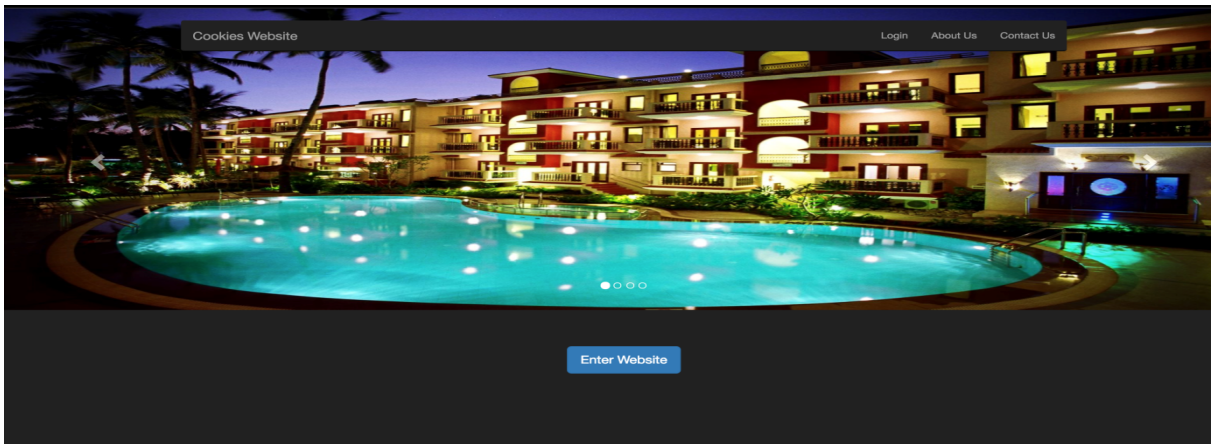
It is a functionality which displays all the bids posted by the user. A functionality is also provided to see only the active bids posted by the user. This also provides the search functionality similar to the Search Bids functionality.

## 7. Contact Us:

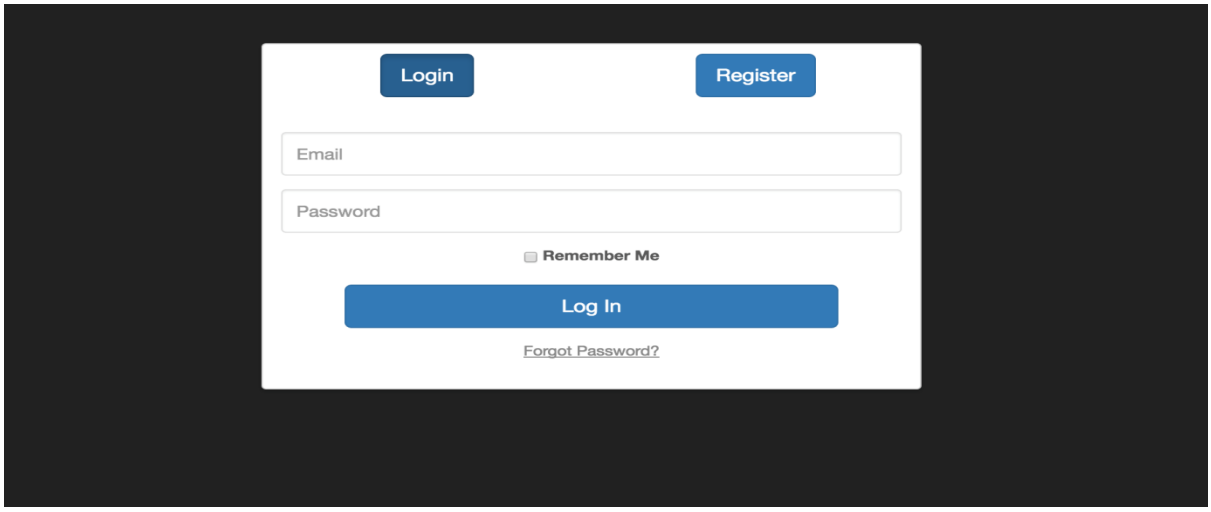
A contact us functionality is provided where the users of the application can contact the website admin with any discrepancy or difficulty about the application or about the content on the website. The users are required to provide a valid mail id and phone number with a brief description about their issue.

Screens:

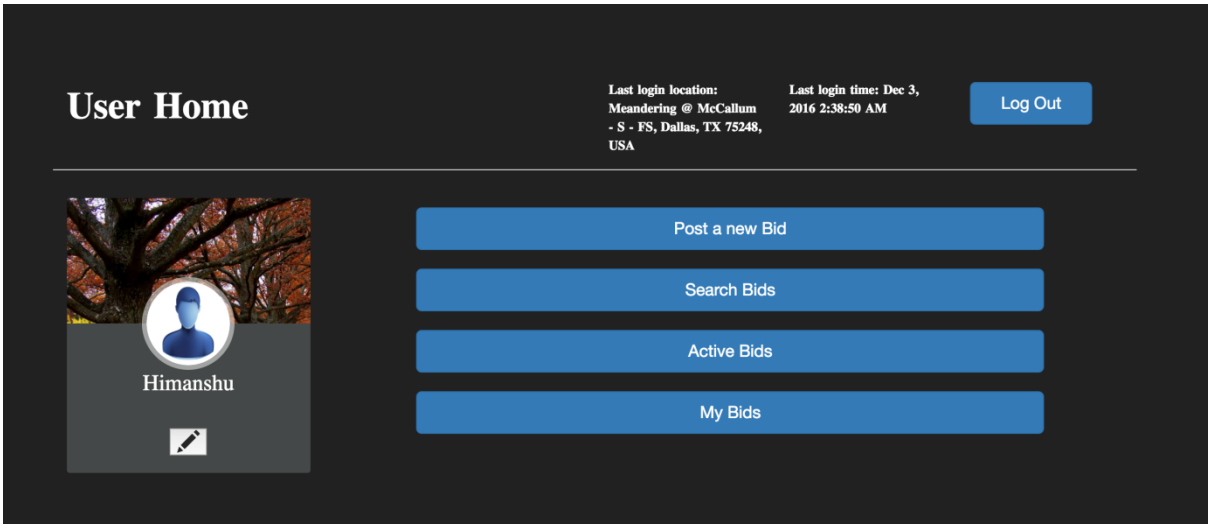
Home Screen



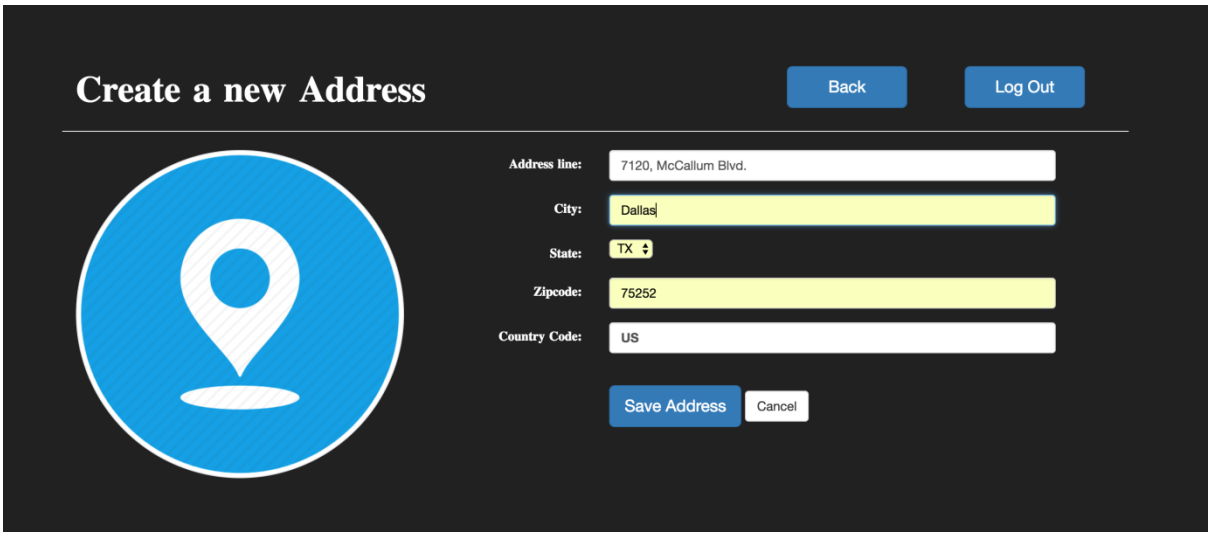

Login Screen



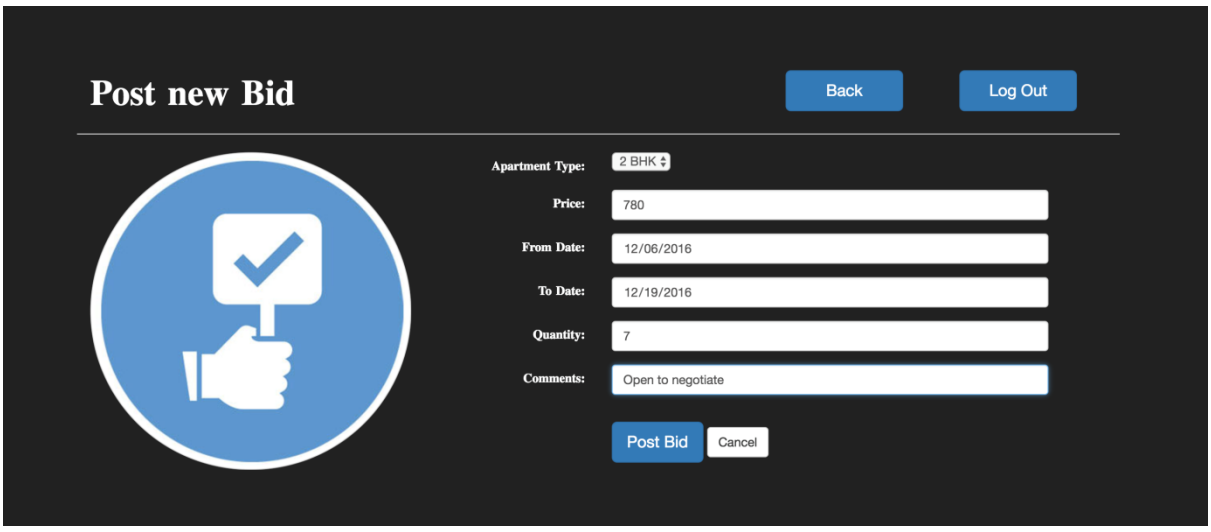
Home Screen



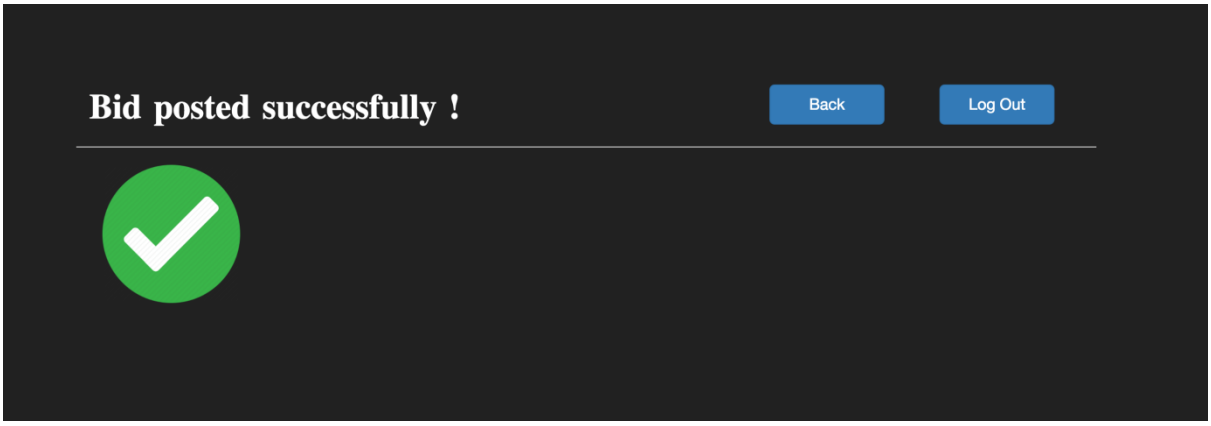

Creating address for Bid:



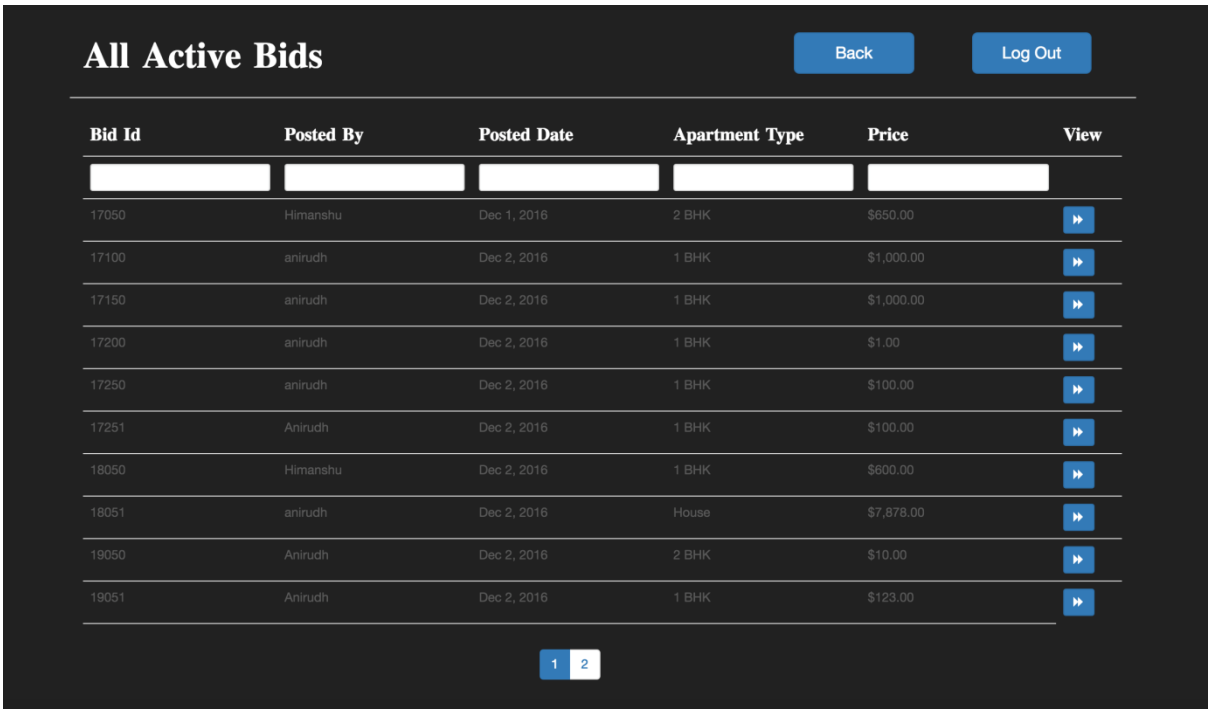

Posting bid screen:



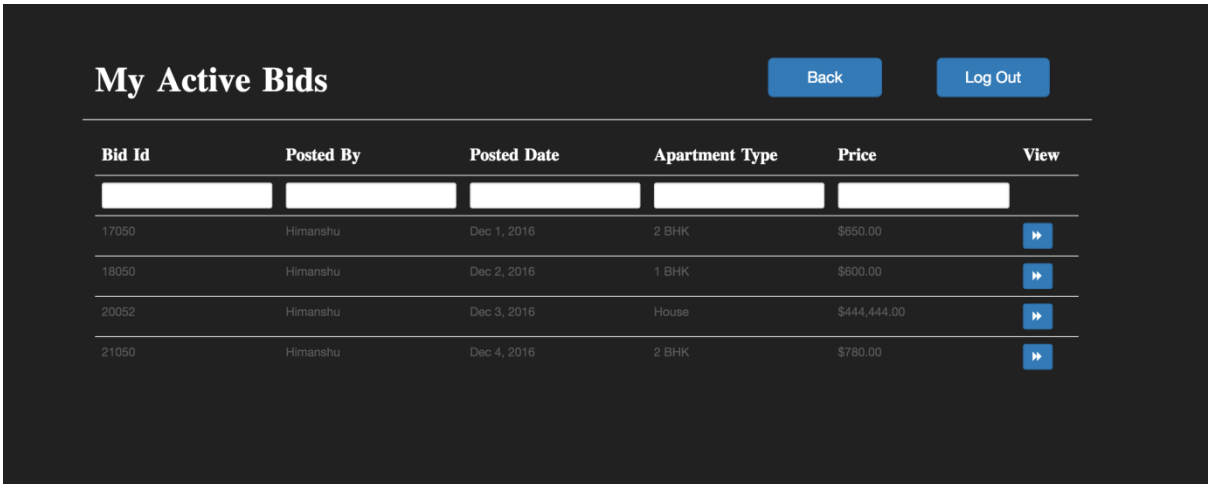
Success Confirmation:



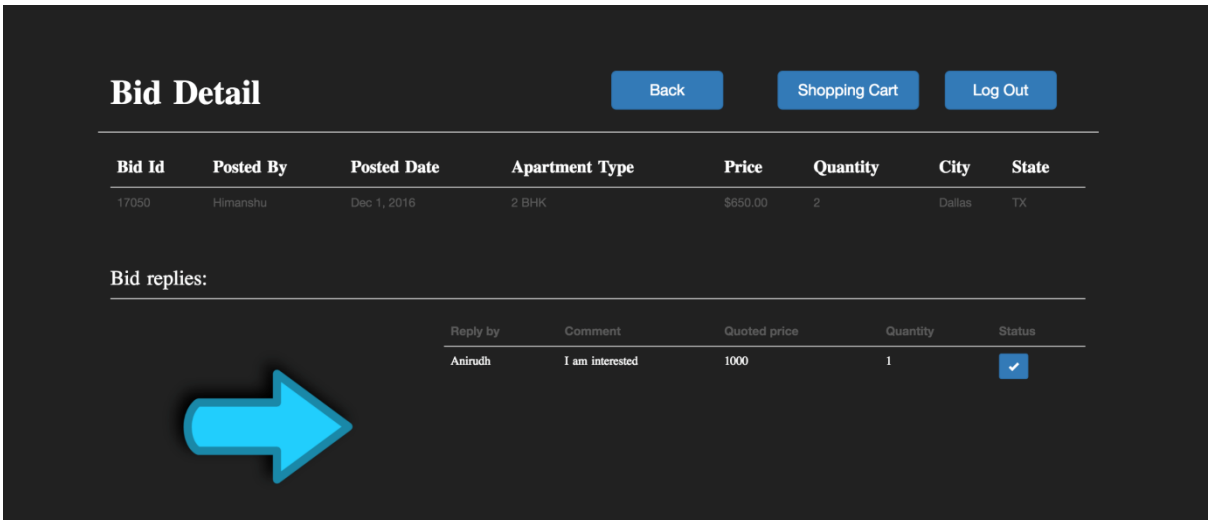

Active Bids Screen:



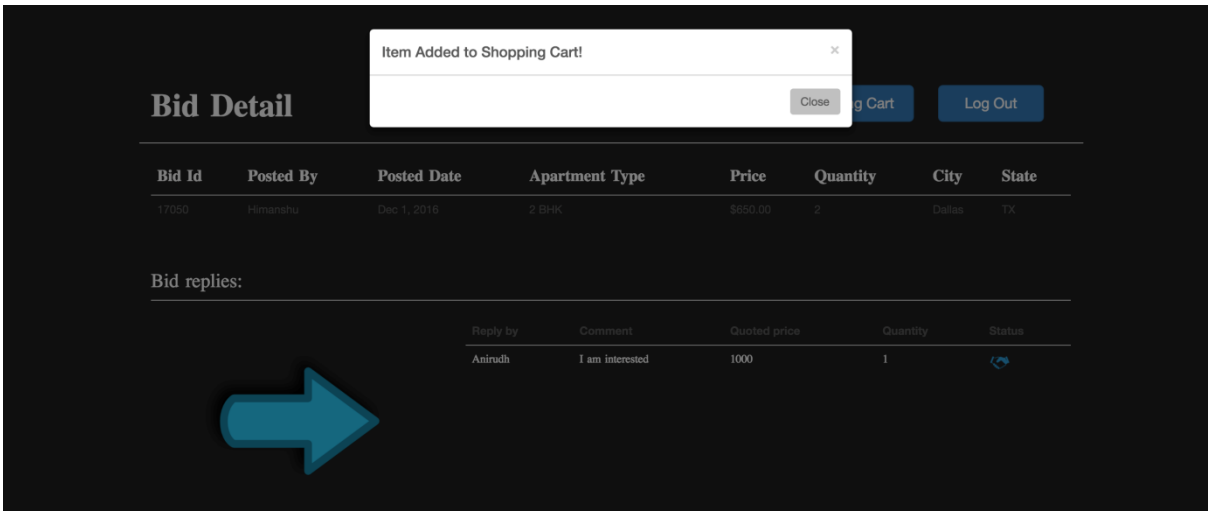
My Active Bids:



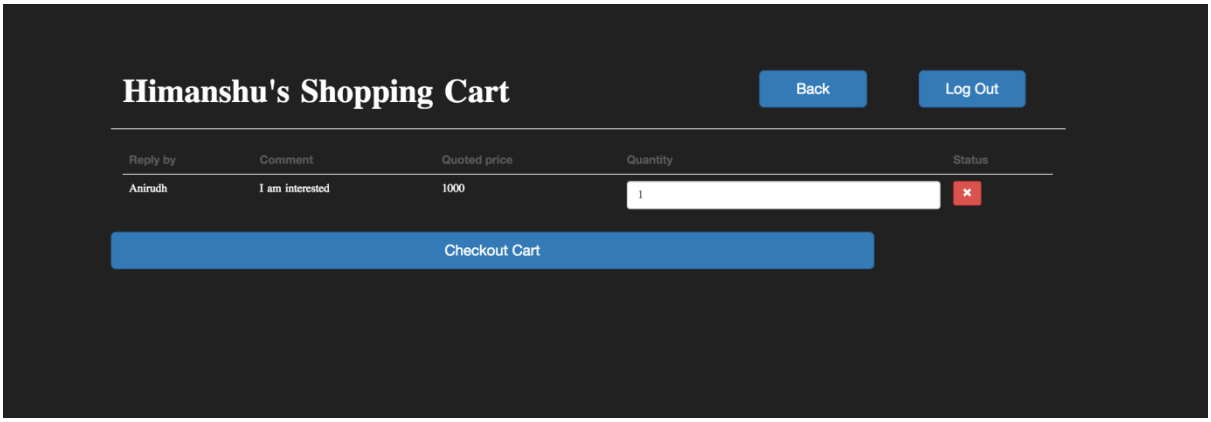

Bid Detail Screen:




Item added Confirmation:



Shopping Cart screen:




Removing an Item from Cart: