I aim to build a bid website for the final project. Client can publish some goods which they want to sale on this web. Consumers can input their expectation price for such product. Then the client has right to decide sale it or not. If client decide to sale, they will know each other contacts. But if no one want to buy it after certain period, it will be withdraw from website.

Hard part:

1, get notification when sellers modified the price.

2, The product part will have the records of who want to buy this products.

3, different users can write comments for sellers.

Want to do

## Three status, On shelves, On selling, and selled.

Build A on selling table.

Consummate put, post, delete operations

Home page relative operation

# Home

## 1, Link: “/home”

### Only include get rest request

## 2, Three dynamic slides: Most welcomed products

### Put products by who is valuable to this company

## 3, Other: Prime List

## 4, Top list: recently upload product

### Get data from product controller( sort by lasted visit time): getAlVisitHistory() => getProductByIds([])

## 5, Most people view List: List the product which has most view times.

### 1, Get data from view\_history table( sort by view times) => getProductByIds([]) **Expect can get all products which id is in the array.**

### 

# ProductPage

## 1, Link: “/products”

## 2, Filter:

### Type: phone, computer, furniture etc, product.getAllByType()

### Date: sort by products date. OnShelves. Date => pass id => product.getAllByIdIn()

### Location: Google map api => OnShelves => userid =>userAddress

### productid => product.getAllByIdIn();

## 3, Biding/Saling:

### OnShelves Type

## 4, Top list

### OnShelves sort by date( Angular Pipe)

# ProductDetailPage

## 1, Link: “/products-detail{id}”

## 2, Image/detail list: product.getAllProducts()

## 3, Bid/Bargain: Judge by on \_shelves type.

### Click => save UserId + productid in Operations History ====Wait for seller reaction

# AdminPage

## 1, Link:”/admin”

## 2, Products: product.getAllProducts()

## 3, Users: user.getAllUsers() calculate the length which is the sum

# UserProfilePage

## 1, Link: “/users”

## 2, upload products: product.addProduct()

### Upload images: productImage.addImage(List<ProductImages>);

### Upload onshelves: onshelves.addProduct(onShelvesProduct);