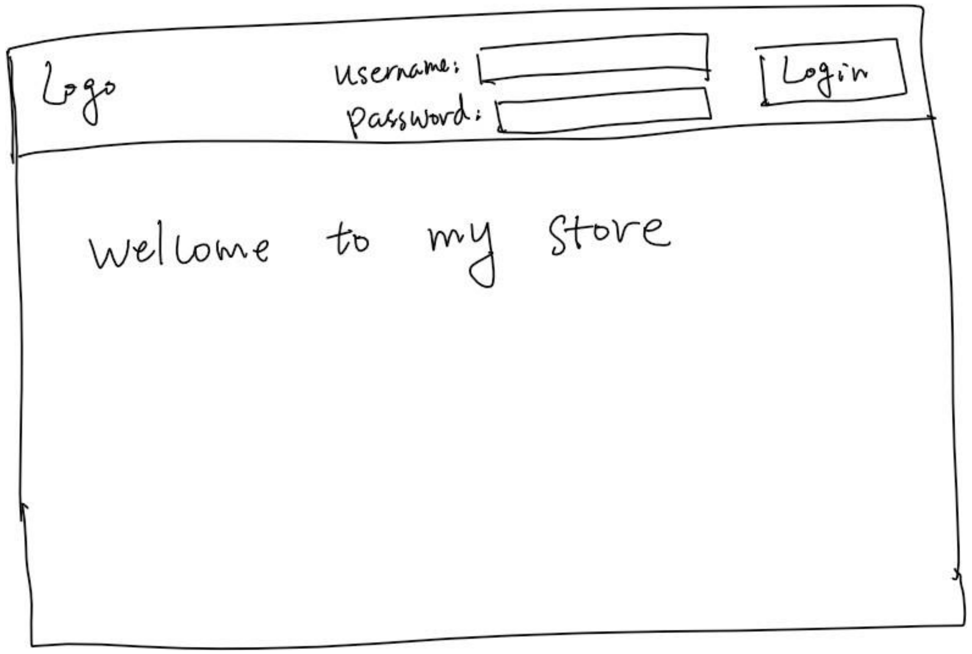
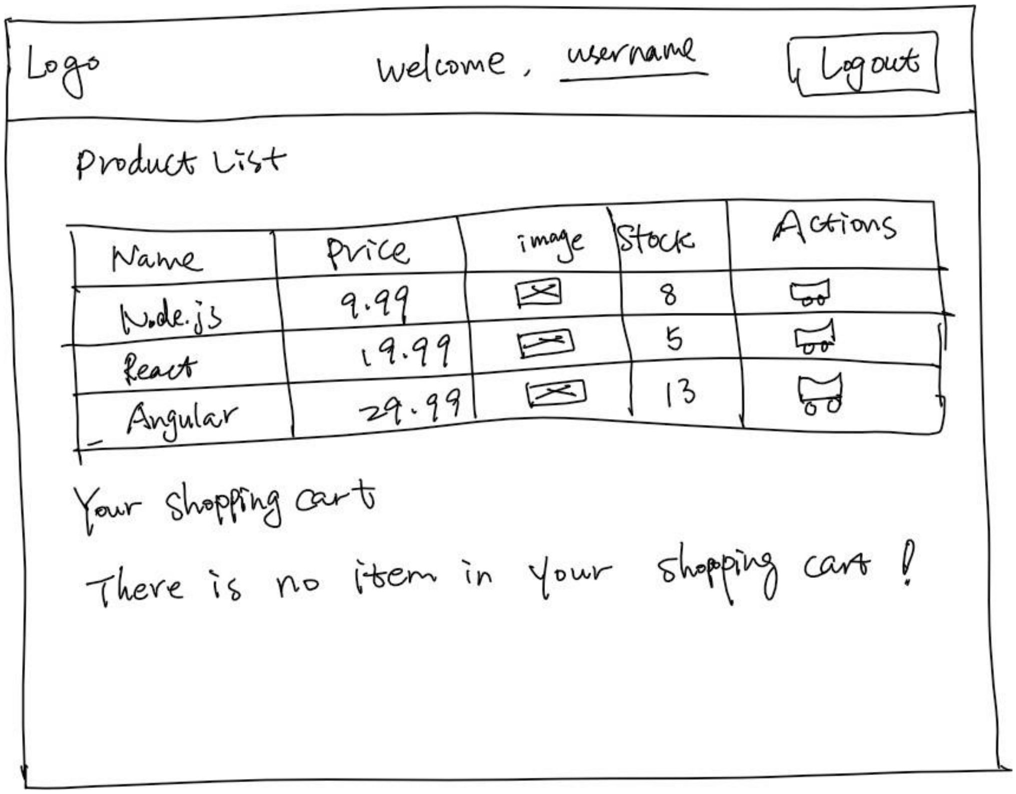
This is a single person project connecting all parts of the content in the course.

The project has the following features.

1. The client and server side are separated project(your client-side code is in 1 folder, your server side code is in another folder). We'll use in-memory database(array, map, set, etc) for this project.
2. You need to make the client project design like the pictures. **You'll lose points if the design doesn't match the requirements. You can choose different colors as you want or add extra style.**
3. There's a login page, user can put username/password to login. All other requested need to be authenticated**. You can hard code several users in DB.** If username and password doesn't match, display "Error" message to user.
    a. Here is the idea on how you implement the login feature:
        i. When use clicks login button with username and password, if you can find the user with correct password, generated a string with current datetime with username, then send back to client side
        ii. In the client side, get the generated string, store it in session storage.
        iii. For all the remaining requests, always pass the generated string located in session storage (about it from https://javascript.info/localstorage) to server side. And server side verify if it has a string with the request. If has, allow access, if not, redirect to login page.



4. After login, it’ll display a page with all products in your store along with items in your shopping carts. **The products can be hardcoded in DB.**
    1) The design below is when there’s no item in the shopping cart



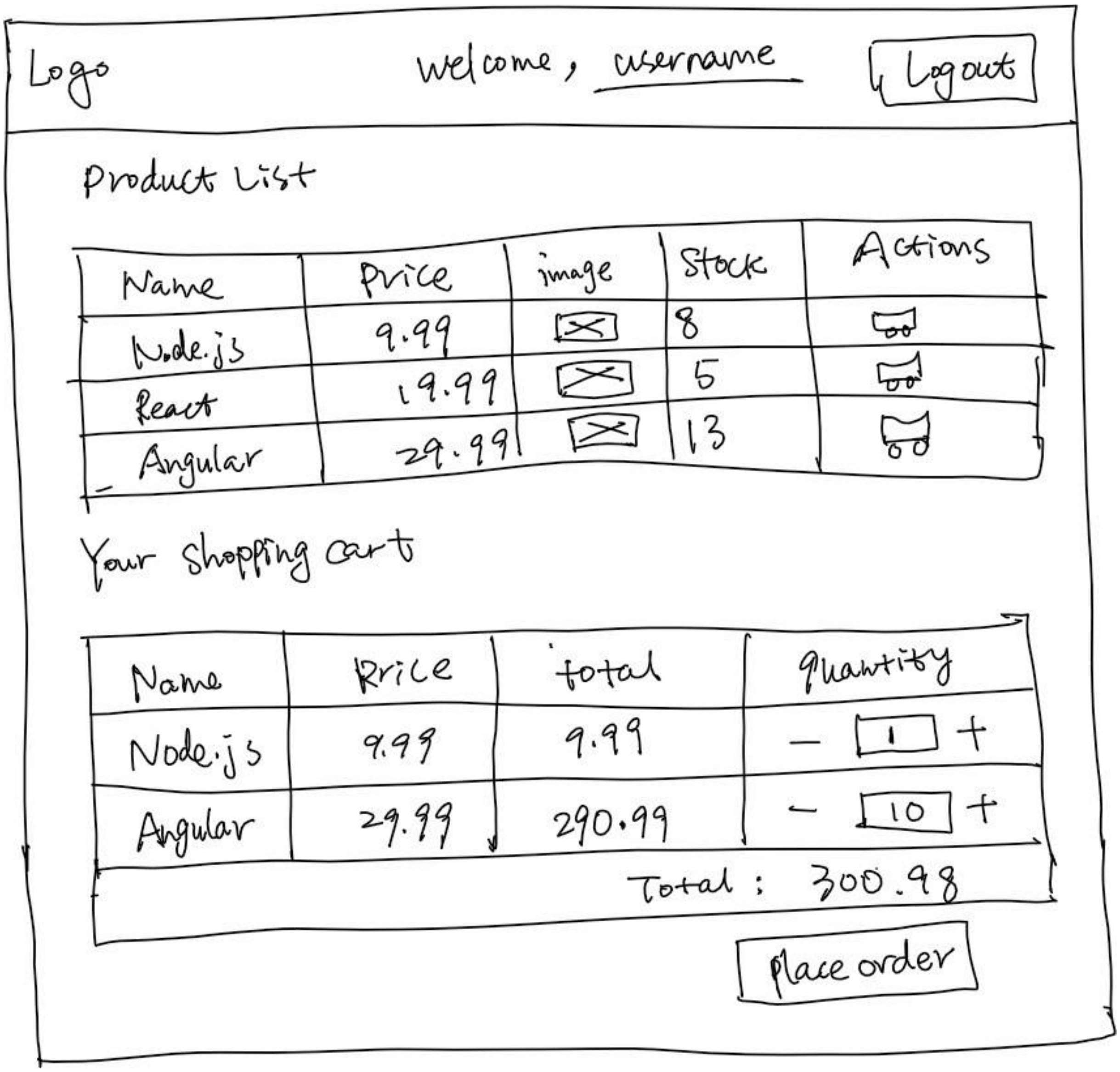
Product List

Name	Price	image	Stock	Actions
Node.js	9.99	☒	8	🛒
React	19.99	☒	5	🛒
Angular	29.99	☒	13	🛒

Your shopping cart

There is no item in your shopping cart !

    2) If you have items in your shopping cart:



Name	Price	image	Stock	Actions
Node.js	9.99	☒	8	🛒
React	19.99	☒	5	🛒
Angular	29.99	☒	13	🛒

Name	Price	total	quantity
Node.js	9.99	9.99	− 1 +
Angular	29.99	290.99	− 10 +

Total : 300.98

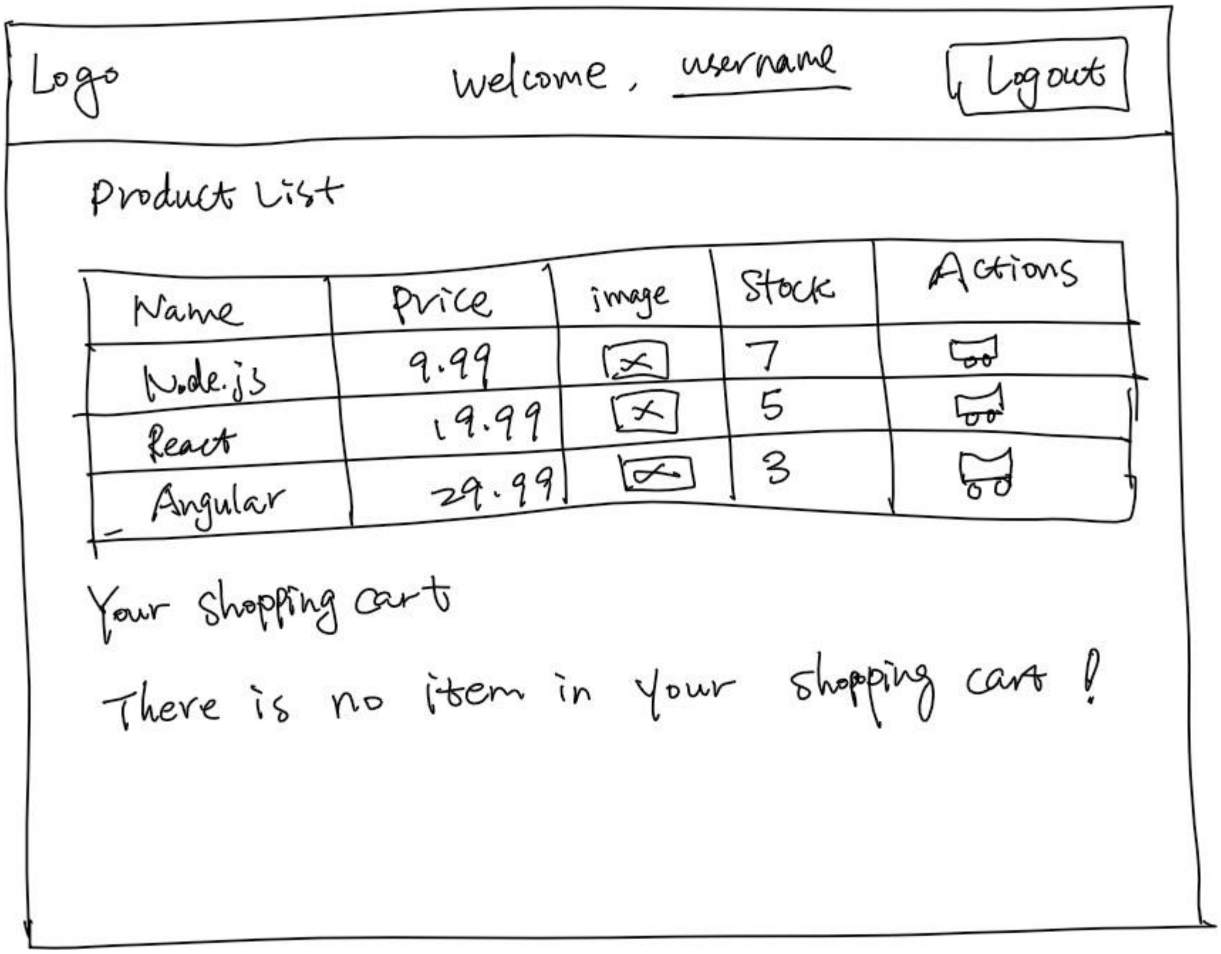
a) Different users have different shopping carts.

b) When you click shopping cart icon in the product list, it adds this product into user’s shopping cart. You can click more than once but cannot exceed the stock limit.

c) In the shopping cart list, you can also click “-“ or “+” to reduce or add more quantity of the product into shopping cart. The minimum value is 1, the max value is the stock. If user reduce to 0, the item should be removed from the shopping cart list. While change the quantity of the product, the total price should be automatically updated. **Need to do client and server side validation.**

d) The stock value is only changed after use place an order. It’s not updated if only add in shopping cart.

5. When user clicks “Place order” button, **it’ll reduce the quantity of products in product list. You MUST do validation checking on Server Side. The quantity cannot be more than stock.** Then return to screen below:



Product List

Name	Price	image	Stock	Actions
Node.js	9.99	🖼	7	🛒
React	19.99	🖼	5	🛒
Angular	29.99	🖼	3	🛒

Your shopping cart

There is no item in your shopping cart!

Grade:

1. Your project grade is based on the project you presented in the video. If you miss some features or some features don't work during presentation, you'll lose points, I won't accept another presentation video.
2. You need to follow the best practices:
    a. the URL naming on the server side
    b. follow the best practices methods for different request.
    c. Your client and server code MUST be separated in different folders, not tangled in 1 folder.
    d. All images of the products must be saved in server side.
3. Name your project with studentIDStudentName e.g: 610001JohnSmith, then zip it to submit. Inside your project, only direct two folders: client, server
4. I'll only run "npm install" on server side, if it doesn't work, need to debug, lose points.
5. NOTE: Don't share your project with other students, you may end up 0.

What to submit:

- Your github link of your project source code
- studentIDStudentName.zip
- presentation video link or file

Extra score: to use "*Swagger*" for the documentation, 1 extra point.