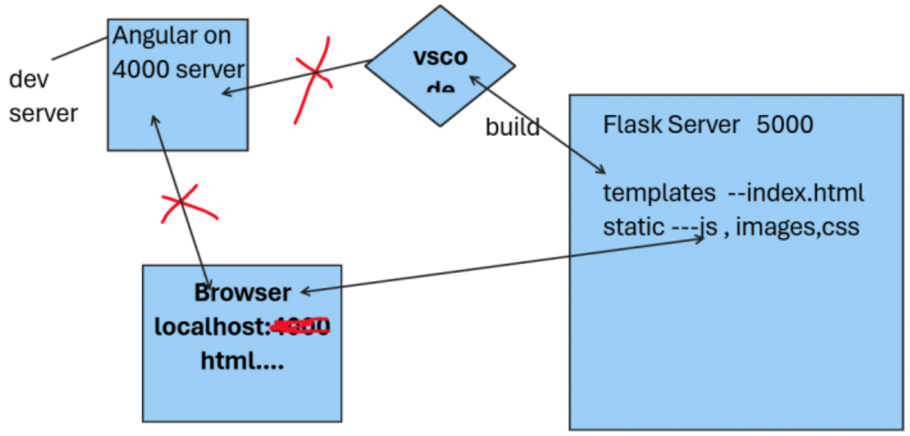
20 February 2025 10:21



This leads to CORS error

IF the angular build is deployed on the Flask server then this will not lead to CORS error

---

1. Put your project images into a folder static, so that each time you add an image in your code, it looks like this:
   <img src="static/any/path/to/img.jpg">

   For example, my project looks like that: project structure (this is an ugly work-around and maybe there is a better way to "translate" the source path on build, please let me know if you find any)
2. When building your app, specify the --deploy-url flag. This will add the prefix to all the src and href properties in the generated index.html:
   ng build --configuration production --deploy-url static/
3. Put all the files except for index.html from the generated dist directory into the static directory of your flask project. Also putt your images here according to how they are organised in your angular project (static/assets/img for me)
4. Put your index.html file into templates directory of your flask app. I put it into templates/ng_app/index.html.
5. Call your angular page like that:
   ```
   @app.route('/ng-app')
   def browsertest():
     return render_template('ng_app/index.html')
   ```

From <https://stackoverflow.com/questions/54147782/how-to-use-angular-build-assets-inside-python-flask-static>

CORS error occurs when the origin of the html in browser is localhost:4000
And the AJAX call from the browser is going to localhost:5000

Template Designer Document

https://jinja.palletsprojects.com/en/stable/templates/

Session Management ---------

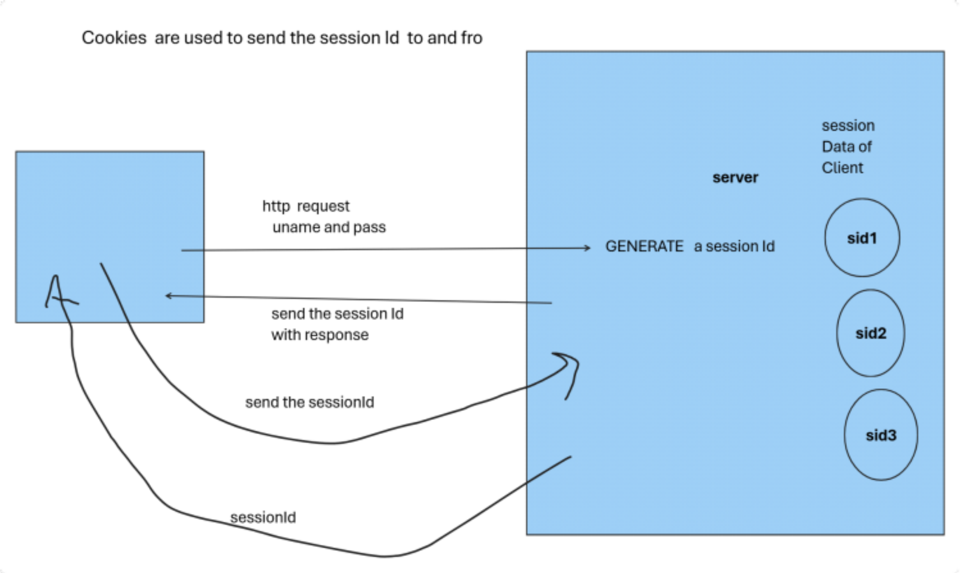
The server should remember the client

The server should maintain client data separately for each client

----

Web Server = Http Server = It is a STATELESS server (http is a stateless protocol)

the http only remembers current request and response



HW ----

1. Create angular build and deploy on server
2. Write code of show product list as discussed in class