User Manual

Beliview

This is user manual of Belieview. Please refer to instructions below.

All the contents are written based on the environment of **Windows** OS.

Google Chrome should be downloaded in your PC**

Git should also be downloaded in your PC**

1. **Code installation**

   A. Open command prompt(cmd).

   B. Move to directory where you want to install:
   our recommended path is: C:\Users\(UserName)
   and you can move to the directory with command:

   cd C:\Users\(UserName)

   since we made all instructions based on such directory

   C. Clone our github's contents with command:

   git clone https://github.com/Adurizz/BlogPostClassify.git

   example)

2. **Python installation**
   : You can follow option A or B as your condition!
   ****Follow only one instruction either A or B!!****

   A. If you never installed python

      i. Go to python download page
      (https://www.python.org/downloads/windows/)

      - Python 3.8.2 - Feb. 24, 2020

      to download **3.8** is recommended, and we developed the software based on **python 3.8.2**, so this version is highly recommended.
      note) This can be **mandatory** since some libraries have high-dependency on python version: tensorflow and tensorflow_text.

   B. If you already installed other version of python
      : create virtual environment for python 3.8.2 is recommended

      i. Open command prompt

      ii. Insert command below:

      conda create -n (any name for virtual environment) python=3.8.2

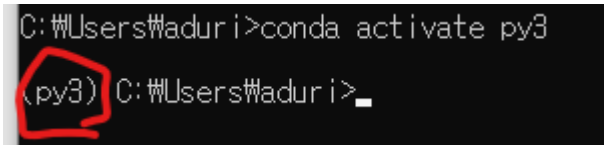
      example)

      iii. Please activate the virtual environment just created
      use command below:

      conda activate (name of virtual environment)

      example)

      note) make sure that the part in red circle indicates your environment's name! → if not, your environment is not activated properly.

3. Library setting with batch file

A. Execute batch file
: we've made the batch file to support your easy-construction of environment

i. Move to flaks_app location : **BlogPostClassify-main\flask_app**

if you follow our recommendation, you can move to the location with commands below:

cd C:\Users\(UserName)\BlogPostClassify-main\flask_app

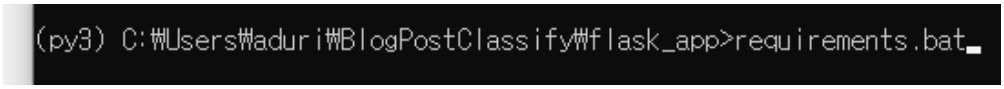
example)

check if the directory is moved to the flask_app.

ii. Execute requirements.bat with this command:

requirements.bat

example)



with the batch file, installation of libraries below included:
flask, flask_cors, beautifulsoup4, pillow, matplotlib, pandas, numpy, requests, tensorflow, tensorflow_text

note) in the case of tensorflow and tensorflow_text, it can takes a long time to install… please wait!
note) installing proper version of python is mandatory since some libraries have high-dependency on python version: tensorflow and tensorflow_text.

4. Setting needed for chrome extension

A. Install node js at download page
(https://nodejs.org/en)

B. Move to chrome extension's location:
BlogPostClassify-main\chrome_extension\popup

if you follow our recommendation, you can move to the location with commands below:

cd C:\Users\(UserName)\BlogPostClassify-main\chrome_extension\popup

example)

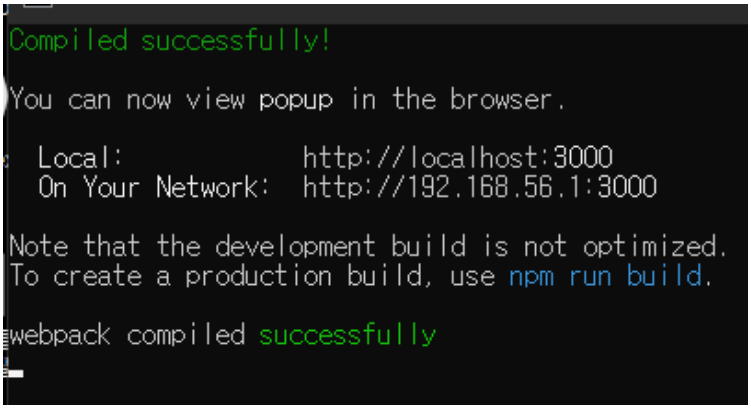
C. Build react UI with commands below:

i. **npm update**
This step can take some time… please wait!

example result)

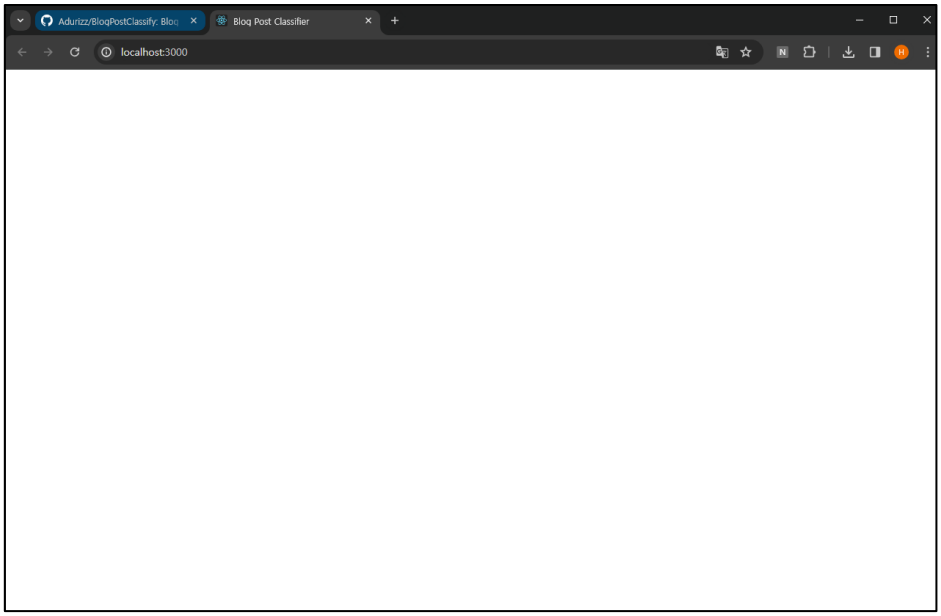
ii. **npm start**

example result)



After this message, you can simply close the prompt window

and then such web tab can revealed:

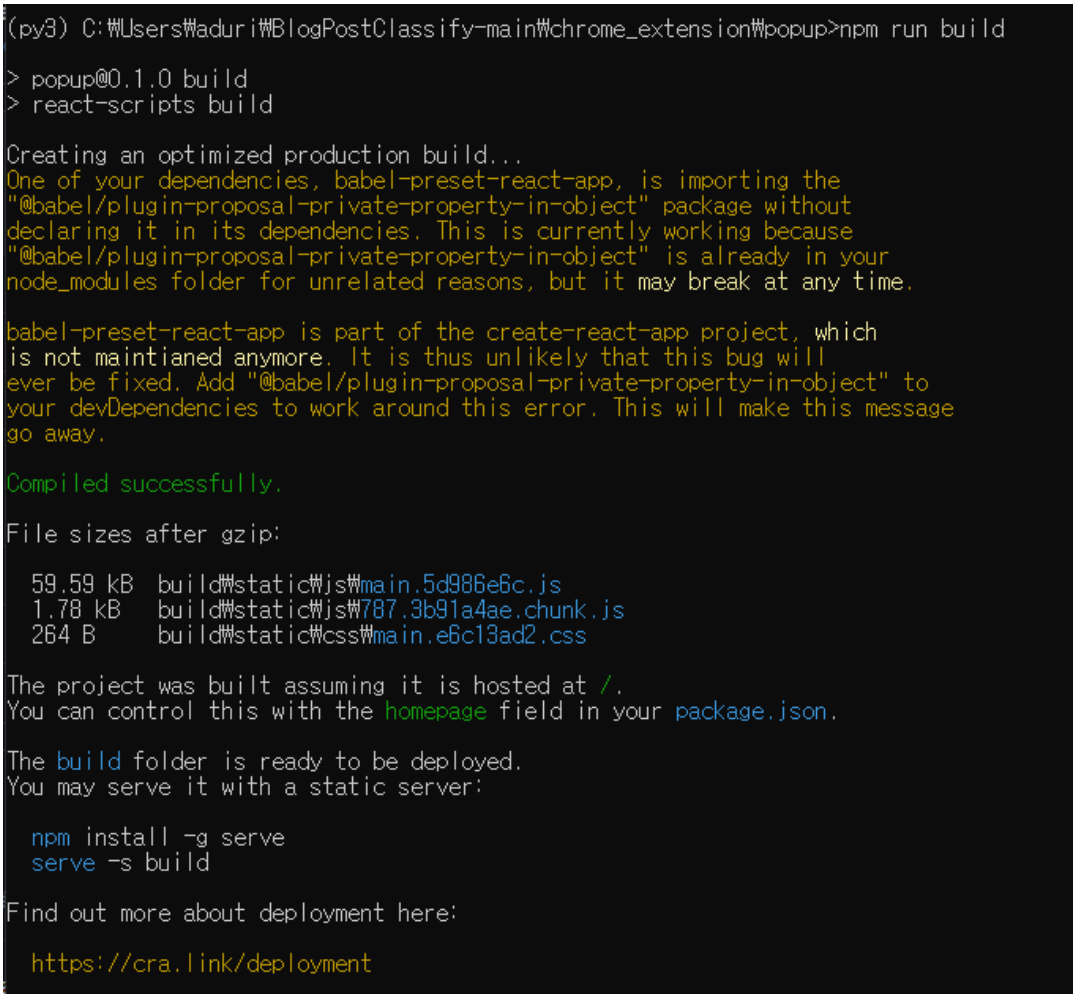


iii. you may have to open another command prompt tab
Also, you should move to the flask_app folder again with commands below:

cd C:\Users\(UserName)\BlogPostClassify-main\chrome_extension\popup

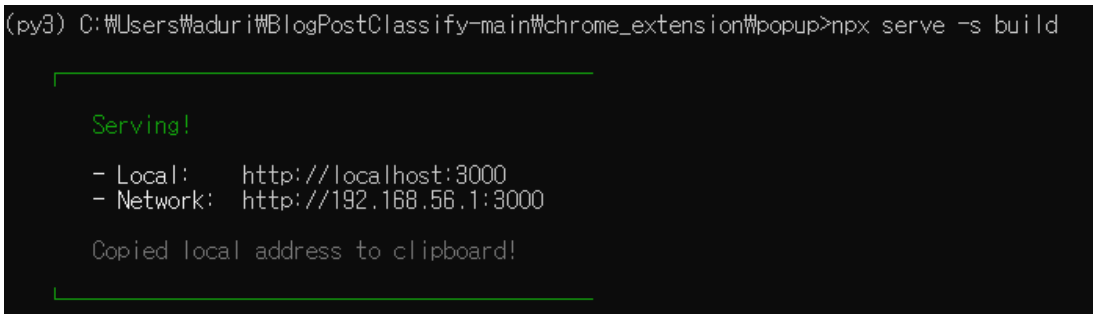
and you can run:
npm run build

example result)



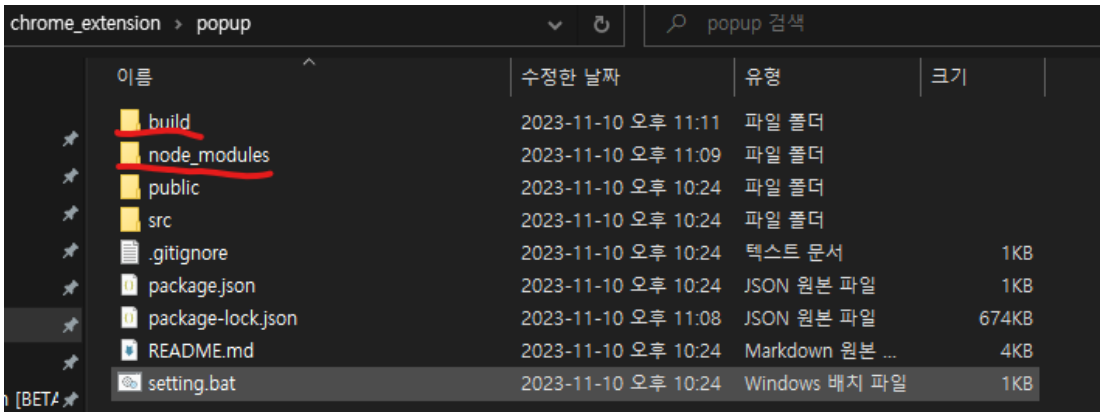
iv. **npx serve -s build**

example result)



you can close this prompt tab now.

note)After four steps above, you can check two new folders are created in popup folder:

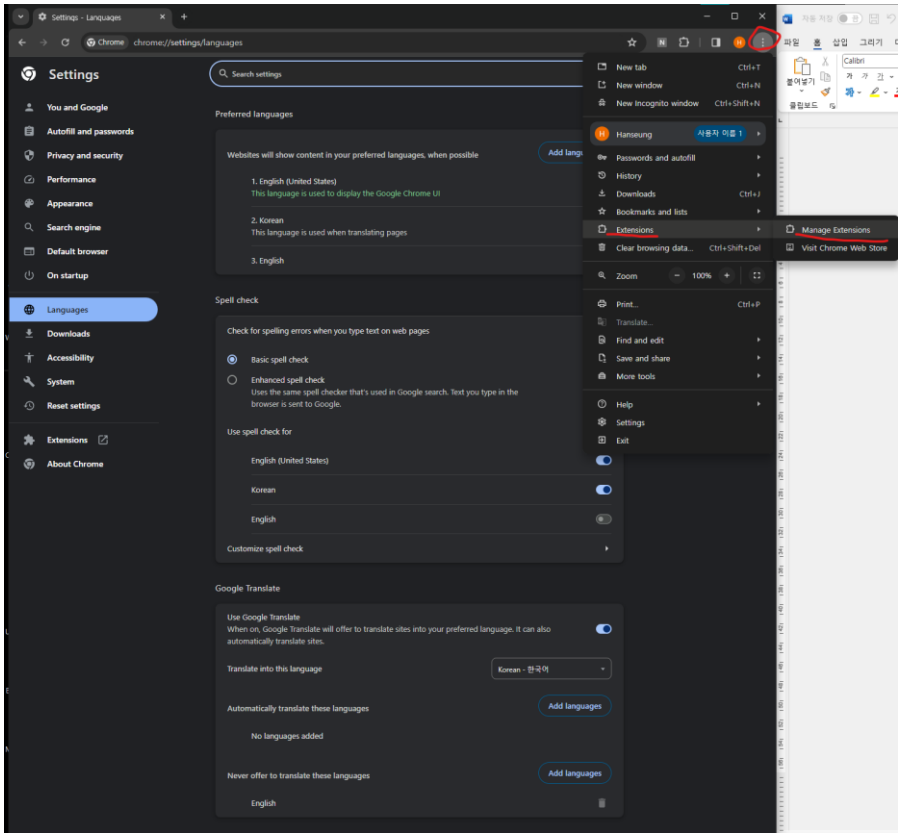


This means build was successfully done!

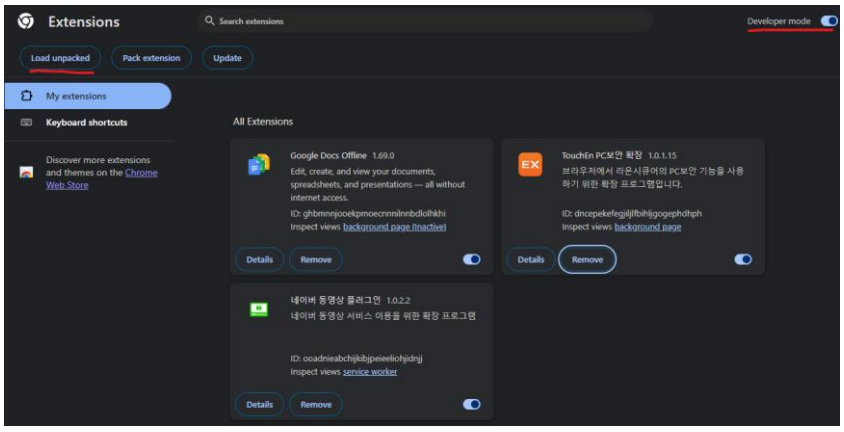
D. Now you should upload the chrome extension file on your chrome environment.

i. Open new tab on chrome

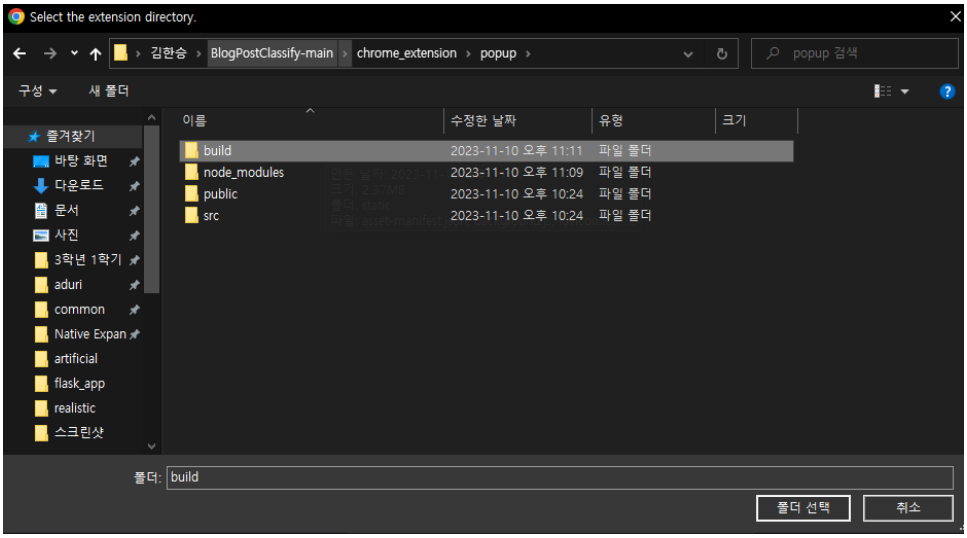
ii. Go to ‘Manage Extensions’ menu



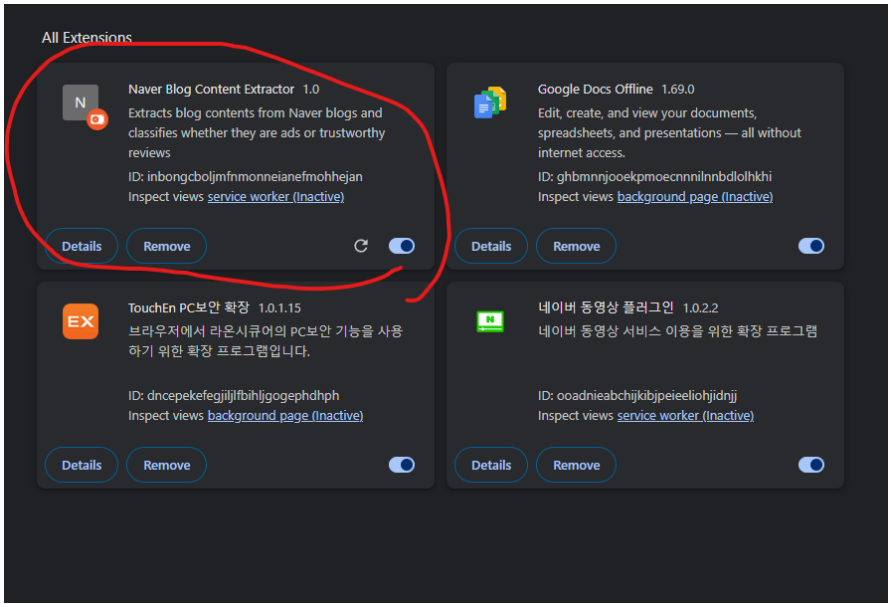
iii. Turn on developer mode and press ‘Load unpacked’ button.



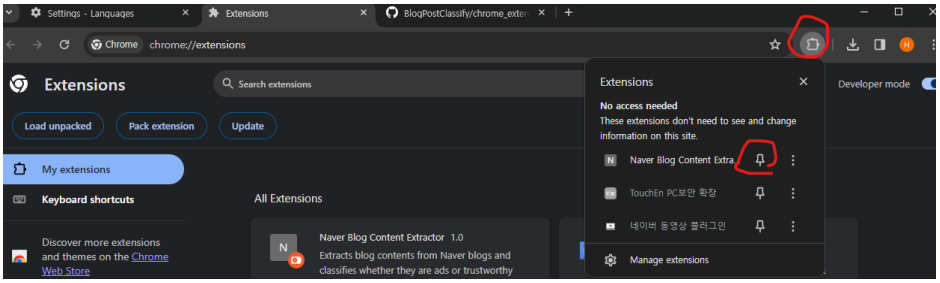
iv. Select ‘build’ folder:
BlogPostClassify-main\chrome_extension\popup\build



example result)



Recommended) pin our extension on your exploration bar!



5. **Execute program on flask server**

   Now you can execute our program with command:

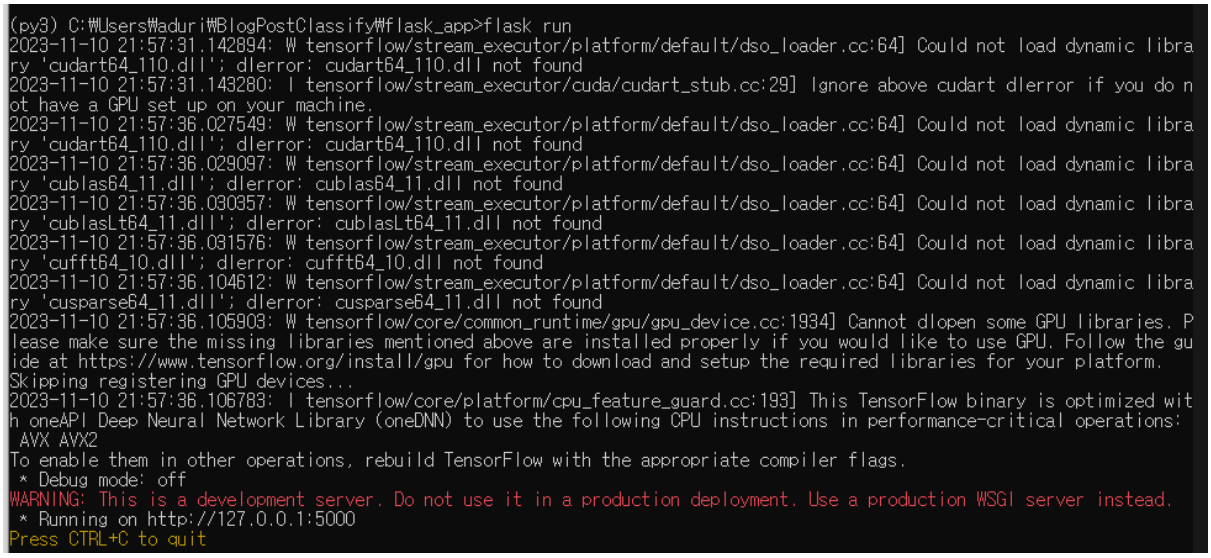
   if you newly opened the prompt window,
   activate (virtual environment name)
   cd C:\Users\(UserName)\BlogPostClassify-main\flask_app

   Above commands should be executed

   then,

   flask run

   example)

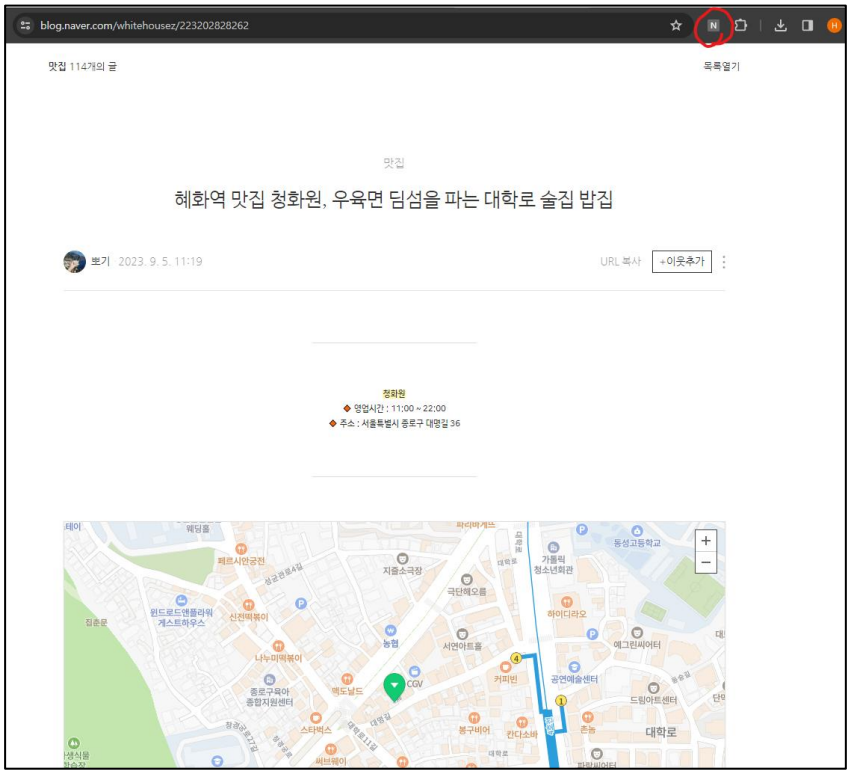


   note) make sure executing location is directory of flask_app folder
   note) make sure your virtual environment is activated
   note) you can ignore messages above since it is related with setting of GPU: we do not use GPU in classification
   note) if message 'Running on http://127.0.0.1:5000' revealed, now you can use our program!

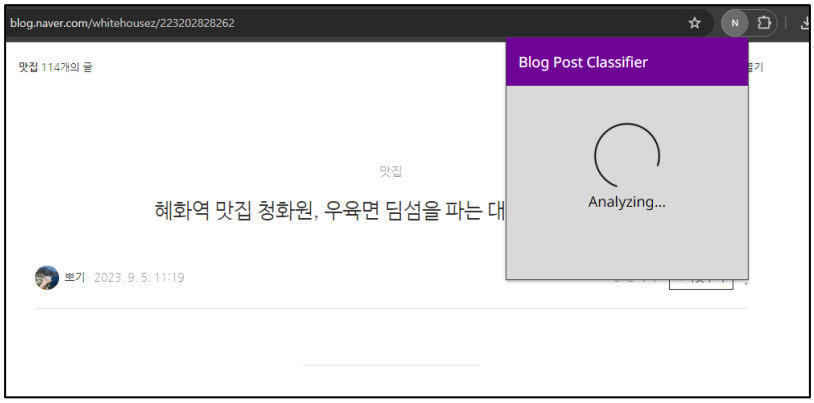
6. **Test**

after you run our program, just access to any websites with **google chrome**
(Since our program developed aimed to chrome extension, you should use google chrome)

A. How to use:
Access to any restaurant review on naver blog, and press our extension's icon



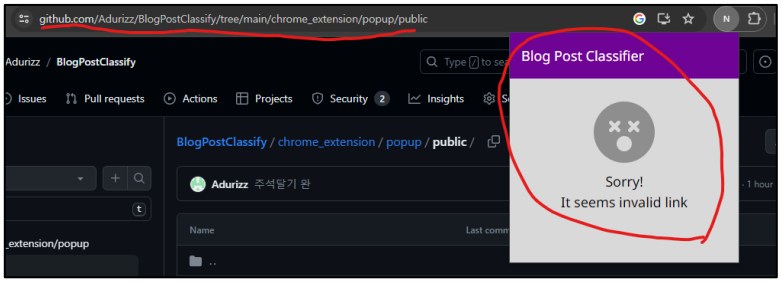
Then analysis starts!



B. Use cases

i. Case 1: if it is not valid candidate

if the link is not a naver blog: restaurant review post, you can see message below:



ii. Case 2: Advertisement

Test link: https://blog.naver.com/whitehousez/223202828262

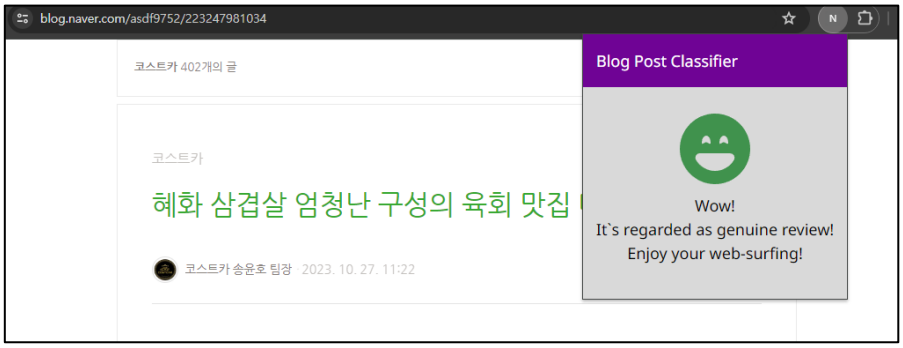
After analyzing,

you can see this window.

iii. Case 3: Non-Advertisement

Test link: https://blog.naver.com/asdf9752/223247981034

After analyzing,



you can see this window.