

User Manual

Beliview

This is user manual of Beliview. 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:

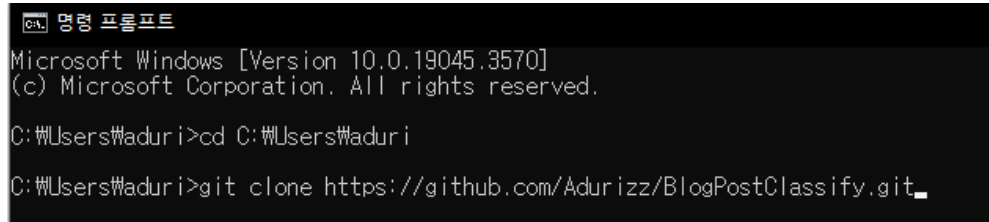
cd C:\Users\(\UserName)

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)

A screenshot of a Windows Command Prompt window. The title bar is black with the text '명령 프롬프트' in white. The window content shows the following text: 'Microsoft Windows [Version 10.0.19045.3570]', '(c) Microsoft Corporation. All rights reserved.', 'C:\Users\Waduri>cd C:\Users\Waduri', and 'C:\Users\Waduri>git clone https://github.com/Adurizz/BlogPostClassify.git_'.

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Waduri>cd C:\Users\Waduri

C:\Users\Waduri>git clone https://github.com/Adurizz/BlogPostClassify.git_
```

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)

```
C:\Users\Waduri>conda create -n py3 python=3.8.2
```

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

conda activate (name of virtual environment)

example)

```
C:\Users\Waduri>conda activate py3
(py3) C:\Users\Waduri>_
```

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 flask_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)

```
(py3) C:\Users\aduri>cd C:\Users\aduri\BlogPostClassify-main\flask_app
(py3) C:\Users\aduri\BlogPostClassify-main\flask_app>_
```

check if the directory is moved to the flask_app.

ii. Execute requirements.bat with this command:

requirements.bat

example)

```
(py3) C:\Users\aduri\BlogPostClassify-main\flask_app>requirements.bat_
```

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)

```
(py3) C:\Users\aduri\BlogPostClassify-main\flask_app>cd C:\Users\aduri\BlogPostClassify-main\chrome_extension\popup
(py3) C:\Users\aduri\BlogPostClassify-main\chrome_extension\popup>_
```

- C. Build react UI with commands below:

- i. **npm update**

This step can take some time... please wait!

example result)

```
(py3) C:\Users\aduri>cd C:\Users\aduri\BlogPostClassify-main\chrome_extension\popup
(py3) C:\Users\aduri\BlogPostClassify-main\chrome_extension\popup>npm update
npm WARN deprecated @babel/plugin-proposal-private-methods@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-private-methods instead.
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser.
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1481 packages, and audited 1482 packages in 2m

248 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

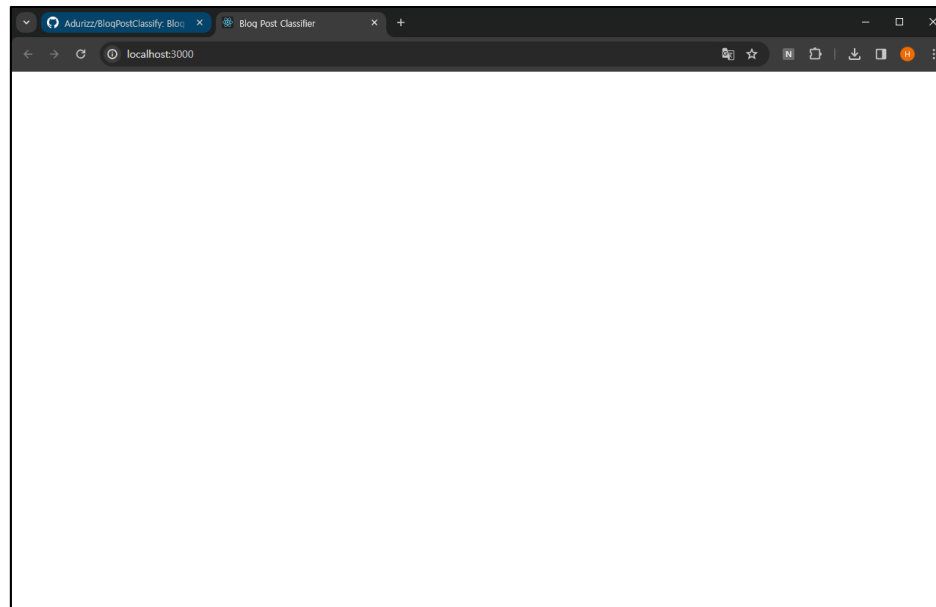
Run `npm audit` for details.
```

- ii. **npm start**
example result)

```
Compiled successfully!  
  
You can now view popup in the browser.  
  
Local:      http://localhost:3000  
On Your Network:  http://192.168.56.1:3000  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
webpack compiled successfully
```

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)

```
(py3) C:\Users\aduri\BlogPostClassify-main\chrome_extension\popup>npm run build

> popup@0.1.0 build
> react-scripts build

Creating an optimized production build...
One of your dependencies, babel-preset-react-app, is importing the
"@babel/plugin-proposal-private-property-in-object" package without
declaring it in its dependencies. This is currently working because
"@babel/plugin-proposal-private-property-in-object" is already in your
node_modules folder for unrelated reasons, but it may break at any time.

babel-preset-react-app is part of the create-react-app project, which
is not maintained anymore. It is thus unlikely that this bug will
ever be fixed. Add "@babel/plugin-proposal-private-property-in-object" to
your devDependencies to work around this error. This will make this message
go away.

Compiled successfully.
File sizes after gzip:

 59.59 kB  build\static\js\main.5d986e6c.js
 1.78 kB   build\static\js\787.3b91a4ae.chunk.js
 264 B     build\static\css\main.e6c13ad2.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment
```

iv. **npx serve -s build**

example result)

```
(py3) C:\Users\Waduri\BlogPostClassify-main\chrome_extension\popup>npx serve -s build
```

```
Serving!  
- Local:    http://localhost:3000  
- Network:  http://192.168.56.1:3000  
  
Copied local address to clipboard!
```

you can close this prompt tab now.

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

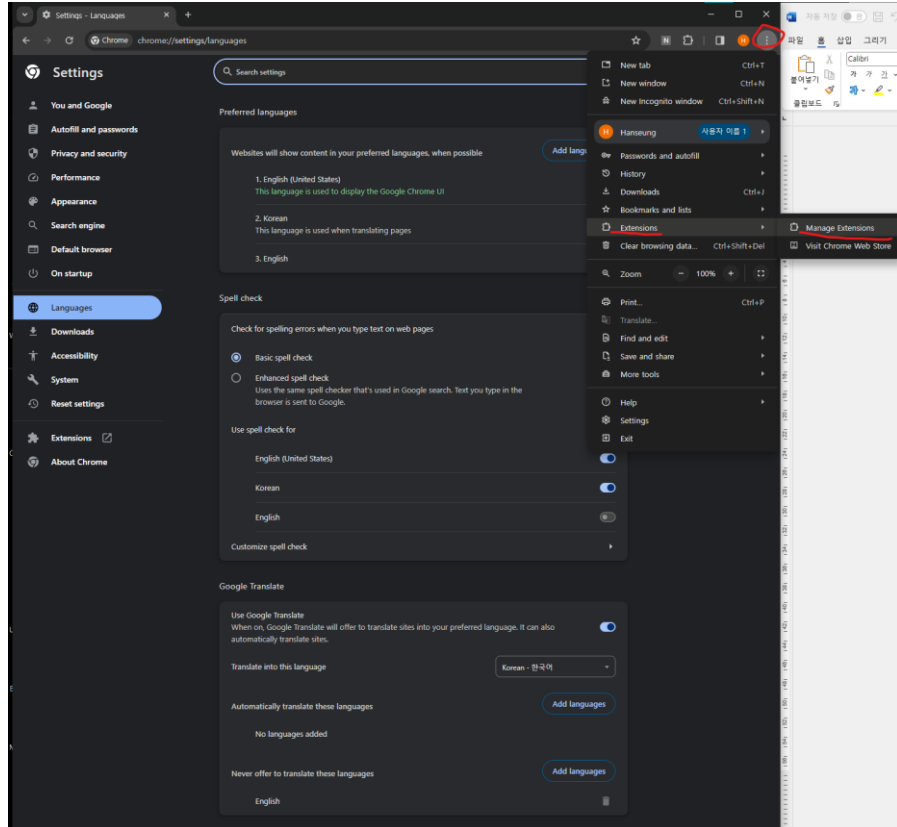
chrome_extension > popup		popup 검색		
이름	수정된 날짜	유형	크기	
build	2023-11-10 오후 11:11	파일 폴더		
node_modules	2023-11-10 오후 11:09	파일 폴더		
public	2023-11-10 오후 10:24	파일 폴더		
src	2023-11-10 오후 10:24	파일 폴더		
.gitignore	2023-11-10 오후 10:24	텍스트 문서	1KB	
package.json	2023-11-10 오후 10:24	JSON 원본 파일	1KB	
package-lock.json	2023-11-10 오후 11:08	JSON 원본 파일	674KB	
README.md	2023-11-10 오후 10:24	Markdown 원본 ...	4KB	
setting.bat	2023-11-10 오후 10:24	Windows 배치 파일	1KB	

This means build was successfully done!

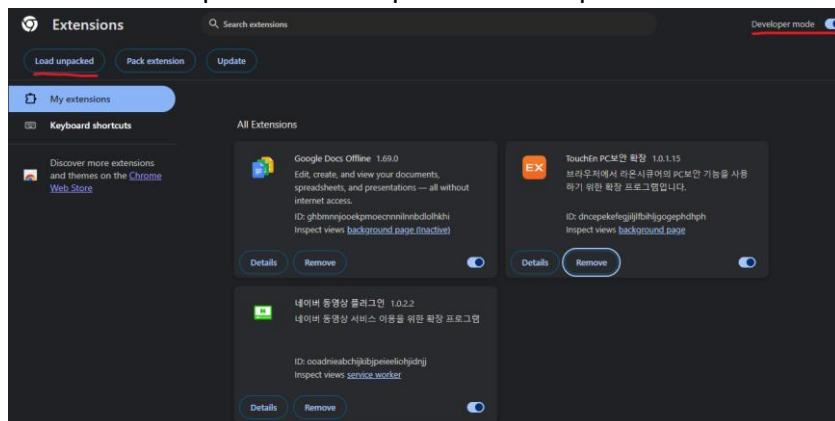
D. Now you should upload the chrome extension file on your chrome environment.

i. Open new tab on chrome

ii. Go to 'Manage Extensions' menu

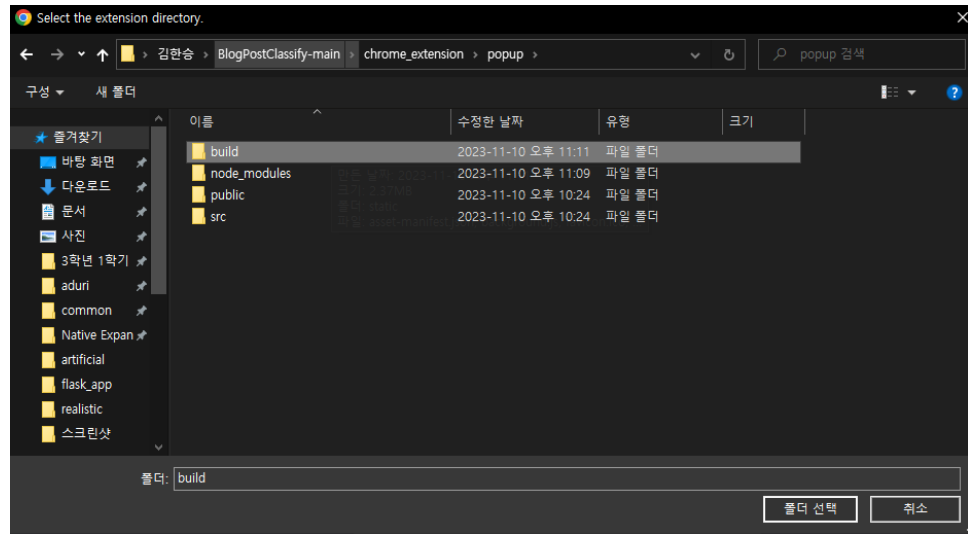


iii. Turn on developer mode and press 'Load unpacked' button.

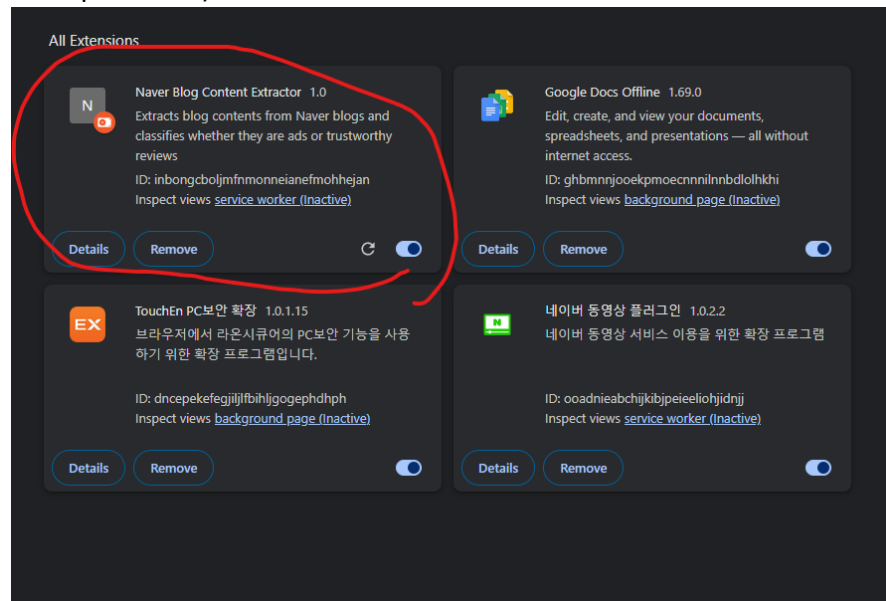


iv. Select 'build' folder:

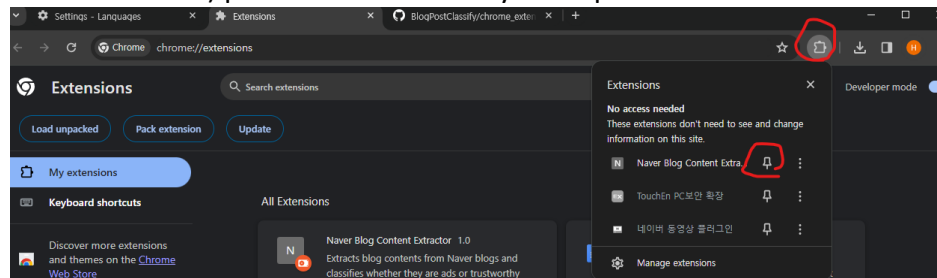
BlogPostClassify-main\chrome_extension\popup\build



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)

```
(py3) C:\Users\maduri\BlogPostClassify\flask_app>flask run
2023-11-10 21:57:31.142894: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-11-10 21:57:31.143280: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2023-11-10 21:57:36.027549: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2023-11-10 21:57:36.029097: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2023-11-10 21:57:36.030357: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2023-11-10 21:57:36.031576: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2023-11-10 21:57:36.104612: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cuspars64_11.dll'; dlerror: cuspars64_11.dll not found
2023-11-10 21:57:36.105903: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2023-11-10 21:57:36.106783: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

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

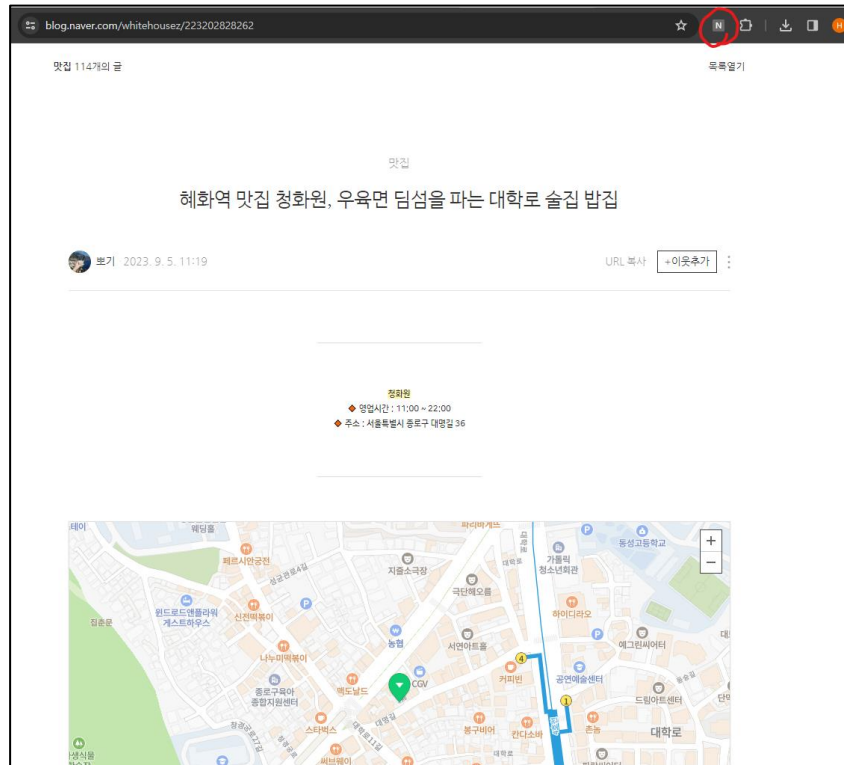
note) if message 'Running on <http://127.0.0.1:5000>' revealed, now you can use our program!

6. Test

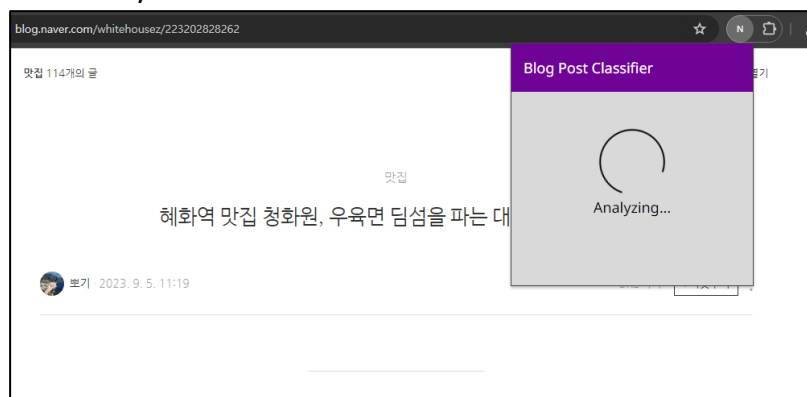
after you run our program, just access to any websites with **google chrome**
(Since our program developed aimed to chrome extension, you should use google chrome)

A. How to use:

Access to any restaurant review on naver blog, and press our extension's icon

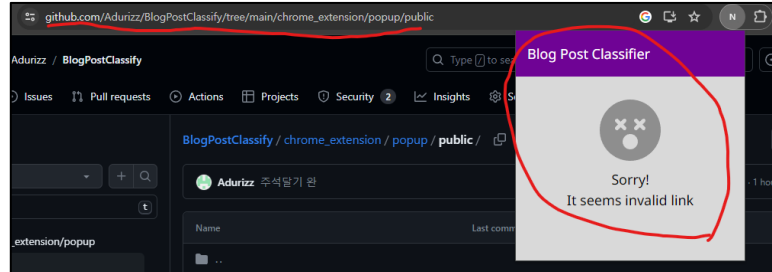


Then analysis starts!



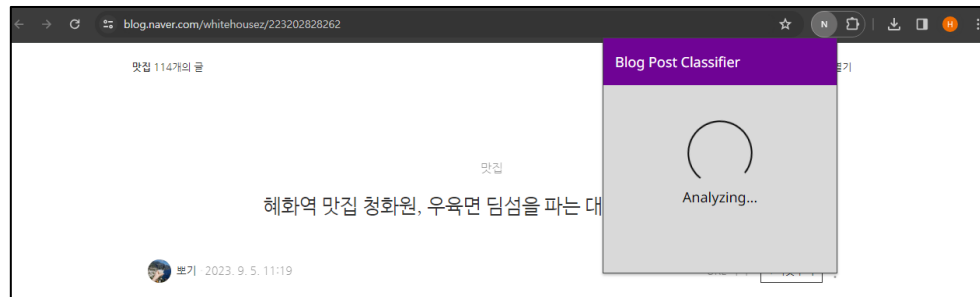
B. Use cases

- 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:

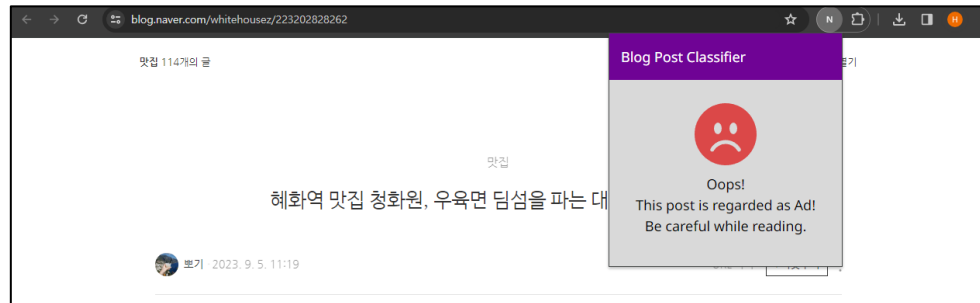


- ii. Case 2: Advertisement

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.