

# Restaurant Reviewer

Software Usage Tutorial

---

Team Mango: Saket Vissapragada, Ameya Gharpure, Shreya Sharma, Sruthi Kilari,  
Amrith Balachander

# How to Install Software

1. Ensure that python is downloaded and installed on machine.
  - a. Type “python” into command prompt. If no prompt opens, utilize the following website to guide your installation on your operating system:  
<https://www.python.org/downloads/>. Otherwise, continue to the next step.
2. Clone the following repository onto your local machine:  
<https://github.com/saketv2/CourseProject.git>

# How to Install Software (cont.)

We utilized Flask to run a backend application for our Chrome extension. The documentation to install Flask is found in <https://flask.palletsprojects.com/en/2.0.x/installation/#>.

To summarize...

1. **Open up a terminal and clone the repository using this link:** <https://github.com/saketv2/CourseProject.git> **and navigate inside this directory**
2. **Place the pipeline.pkl file from** [https://drive.google.com/drive/folders/1hx66MiiFLZFUbvnJv5\\_HuQMYRdQx6vil?usp=sharing](https://drive.google.com/drive/folders/1hx66MiiFLZFUbvnJv5_HuQMYRdQx6vil?usp=sharing) **inside the model directory**
3. **Activate a virtual environment**
  - a. `python3 -m venv venv`
  - b. `source venv/bin/activate`
4. **Install Flask**
  - a. `pip install Flask`
5. **Install requirements from requirements.txt**
  - a. `pip install -r requirements.txt`
6. **Run the Flask app by navigating to the flaskapp directory**
  - a. `$ export FLASK_APP=app`
  - b. `$ flask run`

In order to install our Chrome extension in the browser, a user must navigate to the Manage extensions page of Chrome and turn on developer mode. From there, you can select Load Unpacked and select Restaurant Reviewer from the project's directory. This will install the Chrome extension on your browser. Now, when you click on the extension's icon in the browser menu, you will be able to see the popup window to enter a review.

# Visual workflow of Software Installation

1.

```
((env) (base) wirelessprv-10-194-188-161:~ shreyasharma$ git clone https://github.com/saketv2/CourseProject.git
Cloning into 'CourseProject'...
remote: Enumerating objects: 1567, done.
remote: Total 1567 (delta 0), reused 0 (delta 0), pack-reused 1567
Receiving objects: 100% (1567/1567), 5.32 MiB | 21.13 MiB/s, done.
Resolving deltas: 100% (138/138), done.
((env) (base) wirelessprv-10-194-188-161:~ shreyasharma$ cd CourseProject/
(env) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$
```

2.

```
((env) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$ python3 -m venv venv
(env) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$ source venv/bin/activate
(env) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$
```

3.

```
((env) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$ pip install flask
Collecting flask
  Using cached Flask-2.0.2-py3-none-any.whl (95 kB)
Collecting Jinja2>=3.0
  Using cached Jinja2-3.0.3-py3-none-any.whl (133 kB)
Collecting itsdangerous>=2.0
  Using cached itsdangerous-2.0.1-py3-none-any.whl (18 kB)
Collecting Werkzeug>=2.0
  Using cached Werkzeug-2.0.2-py3-none-any.whl (288 kB)
Collecting click>=7.1.2
  Using cached click-8.0.3-py3-none-any.whl (97 kB)
Collecting MarkupSafe>=2.0
  Using cached MarkupSafe-2.0.1-cp38-cp38-macosx_10_9_x86_64.whl (13 kB)
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, flask
Successfully installed Jinja2-3.0.3 MarkupSafe-2.0.1 Werkzeug-2.0.2 click-8.0.3 flask-2.0.2 itsdangerous-2.0.1
```

# Visual workflow of Software Installation (cont.)

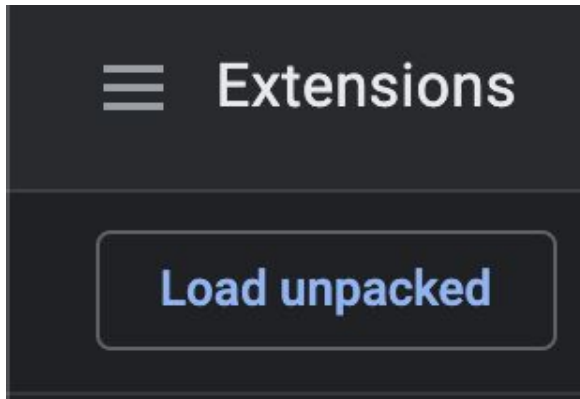
4.

```
(venv) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$ pip install -r requirements.txt
Collecting beautifulsoup4==4.10.0
  Using cached beautifulsoup4-4.10.0-py3-none-any.whl (97 kB)
Collecting blis==0.7.5
  Using cached blis-0.7.5-cp38-cp38-macosx_10_9_x86_64.whl (5.8 MB)
Collecting catalogue==2.0.6
  Using cached catalogue-2.0.6-py3-none-any.whl (17 kB)
Collecting certifi==2021.10.8
```

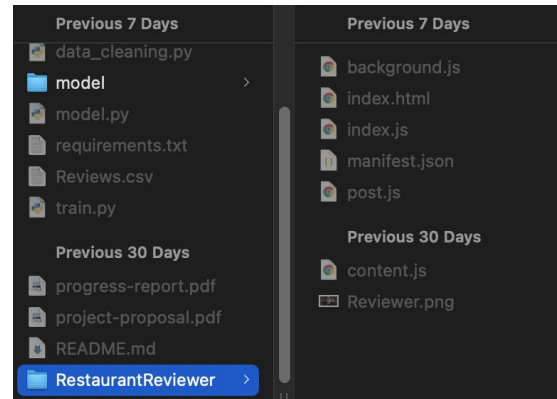
5.

```
(venv) (base) wirelessprv-10-194-188-161:CourseProject shreyasharma$ flask run
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

6.



7.



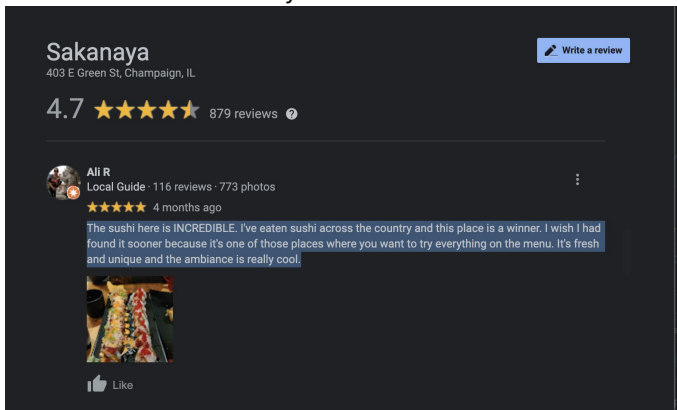
Now, the chrome extension should be ready to use!

# How to Use Software

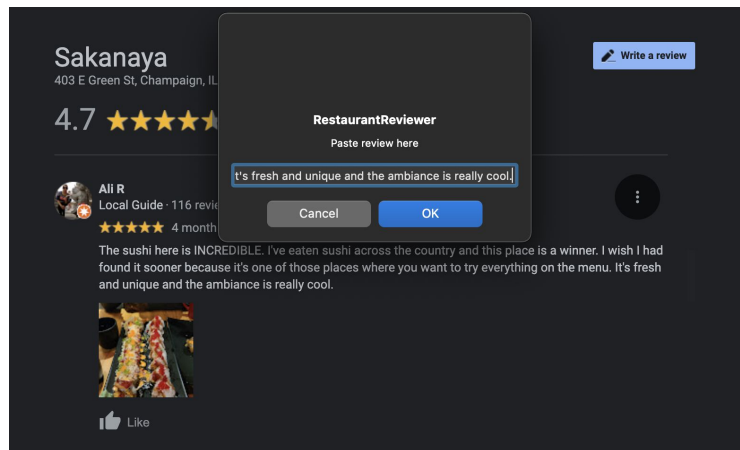
1. User Launches the Chrome Extension.
2. User inputs text into the text field.
3. User can click the “OK” button to receive the standardized review rating
  - a. If the text that the user inputted into the text field is undesired, the user can click the “Cancel” button instead.
4. Review the generated rating after clicking the “OK” button.

# Visual workflow of Software Usage

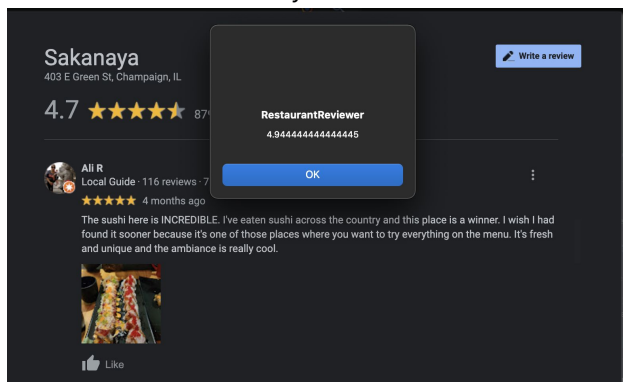
1. Find review that you desire to standardize



2. Activate chrome extension and input text from review into chrome extension



3. Click OK to receive your standardized score!



Now you know if you can trust the review you are reading!! Enjoy

# Use Cases

1. Enter the text *“The sushi here is INCREDIBLE. I've eaten sushi across the country and this place is a winner. I wish I had found it sooner because it's one of those places where you want to try everything on the menu. It's fresh and unique and the ambiance is really cool.”* in the chrome extension's text field.



Thank You

