

# **KITCHENSYNC**

## **DEPLOYMENT MANUAL**

### **TABLE CONTENTS**

1. Introduction
2. System Requirements
3. Software Requirements
4. Installation of Applications
  - Installing Python
  - Installing Anaconda
  - Installing Visual Studio Code (VSCode)
  - Installing Chrome
5. Clone Our GitHub Repository
6. Starting the server
7. Running the Application
8. Results
9. GitHub Wiki

# 1. INTRODUCTION

Welcome to the Deployment Manual of the KitchenSync Application. This document will provide you with a step-by-step guide for the effective installation of our Software on your system or in your Environment.

## Project Overview

KitchenSync is designed for busy individuals, families, and small firms that struggle to keep their pantries organized and cut down on food waste.

KitchenSync Streamlines your pantry management and meal planning Seamlessly. It is a sophisticated kitchen assistance platform, that enhances the management of your pantry and offers recipe options that fit your available ingredients, organizes meal plans, and reminds you of the shelf-life of your food items.

Unlike other traditional inventory systems and shopping list apps, KitchenSync adjusts to your dietary preferences and schedules by making use of readily available ingredients to simplify the meal preparation process.

## Technology Stack

- ❖ Framework: Java Script
- ❖ Backend: JavaScript, Flask
- ❖ Database: SQLite
- ❖ Version Control: Anaconda
- ❖ Code Editor: Visual Studio Code

## **2. SYSTEM REQUIREMENTS**

### **Hardware requirements (Desktop)**

#### **a. Processor (CPU):**

Equivalent to AMD Ryzen 3 2.4Ghz (4-core)/

Intel Core i-3 2.6Ghz(4-core physical)/

Apple M1 (8-core)

#### **b. Memory (RAM):**

A minimum of 8 Gb DDR4 @ 4200 Mhz

#### **c. Storage:**

5 Gb of SSD space or equivalent for optimal performance

#### **d. GPU:**

Intel Iris Xe with 1.30 Ghz/ Radeon Graphics 2.00 Ghz

#### **e. Network Connectivity:**

Minimum download speed of 1Mpbs

## **3. SOFTWARE REQUIREMENTS**

#### **a. Operating System (windows)**

Windows 8/10/11

#### **b. Operating System (Mac)**

MacOS Monterey / Big Sur / Catalina/Mojave

#### **c. Operating System (Linux)**

Ubuntu Jammy Jellyfish / Mantic Minotaur

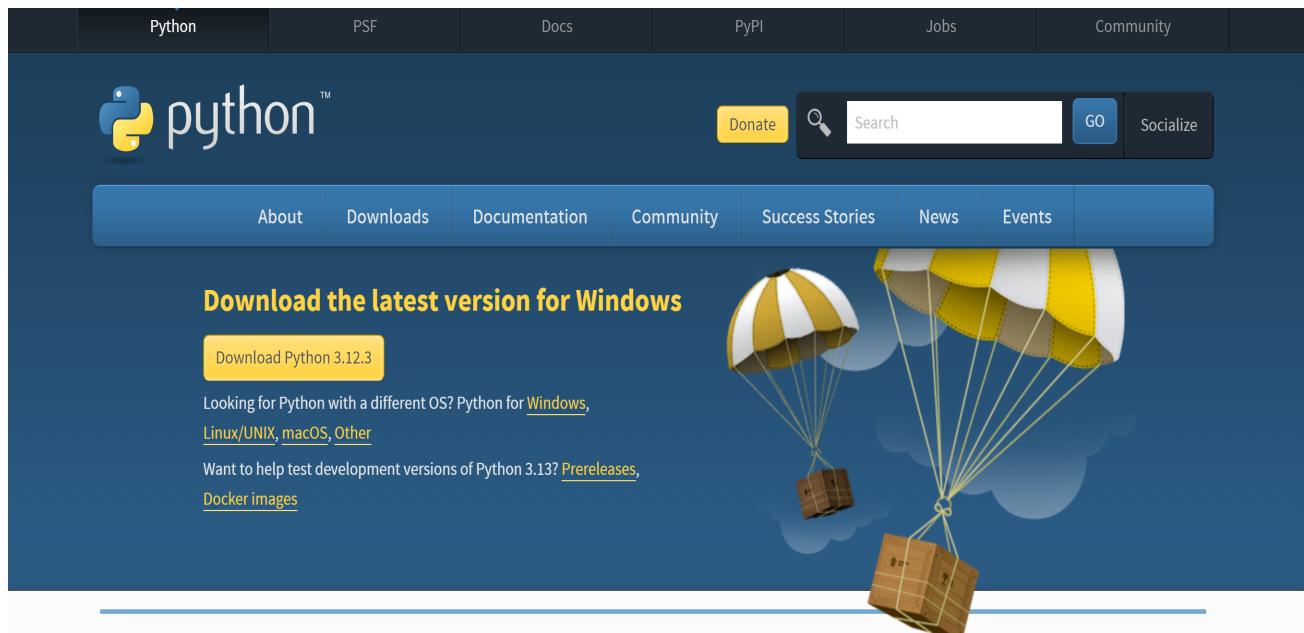
#### **d. Other Requirement: Latest Chrome version**

## 4. INSTALLATION OF APPLICATIONS

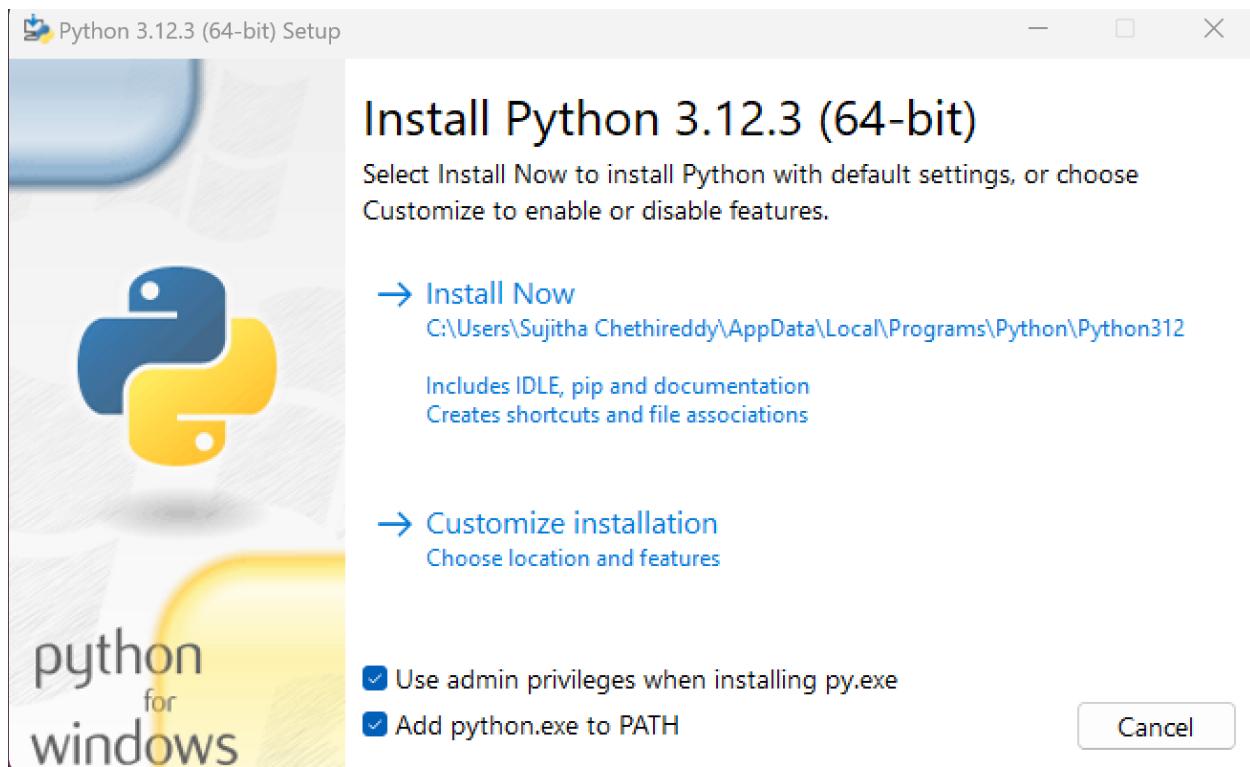
### a. Installing Python

The first and foremost step is to install Python on your local machine or your systems using the Link Below.

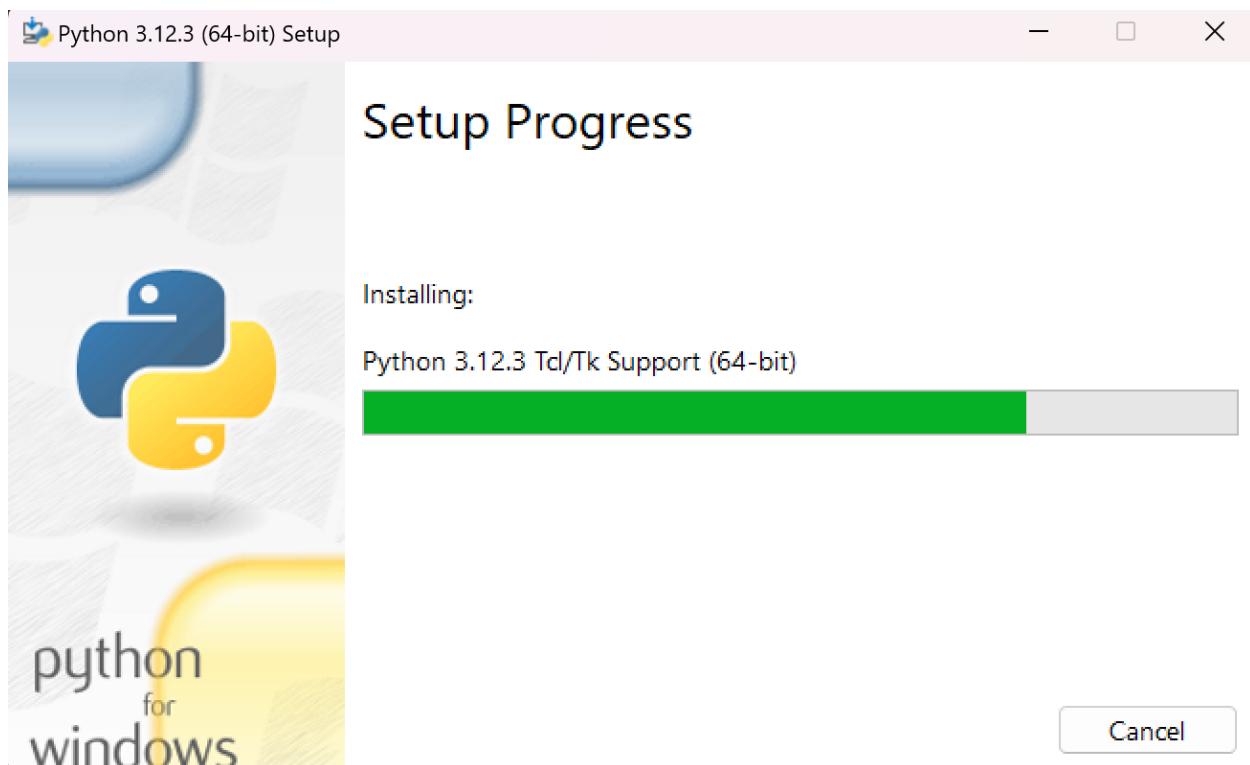
Python Installer: <https://www.python.org/downloads/>



- You need to Download the Latest Version of Python by clicking the button Available there in Your Systems (Windows/Mac/Linux)
- Then you will see this pop-up with two options. Click the checkbox below (Add Python to PATH) and install Python in your system with your desired path(files in your system).



- Then click Install Now And you will see this pop-up.

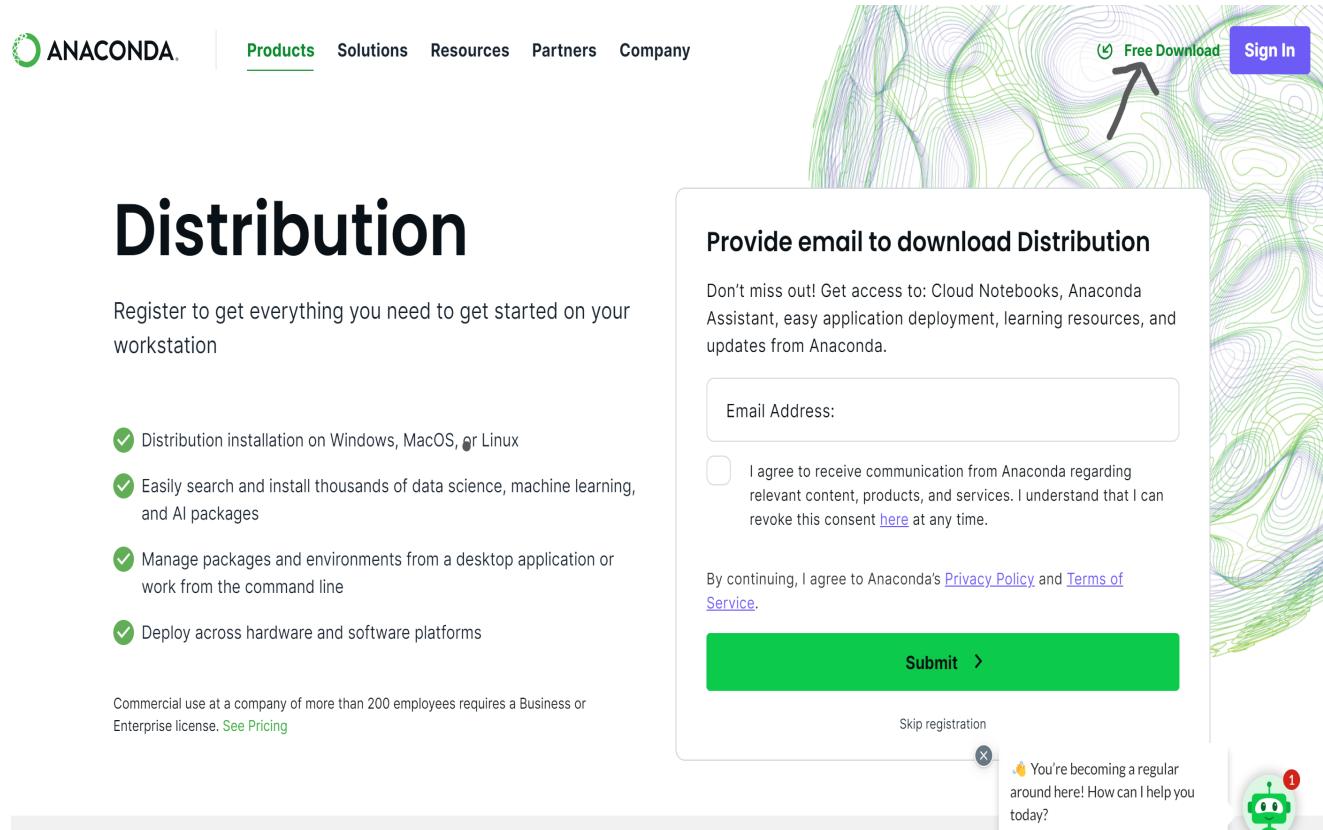


- After that Python is successfully downloaded to your system. Now double click on Python application and run it.

## b. Installing Anaconda

The next important step is installing Anaconda to your System or your Local Machine

Anaconda Installer: <https://www.anaconda.com/download>



The screenshot shows the Anaconda website's download page. At the top, there's a navigation bar with the Anaconda logo, a search bar, and links for Products, Solutions, Resources, Partners, and Company. A prominent green button labeled "Free Download" is highlighted with a large black arrow. Below the button, a section titled "Provide email to download Distribution" asks for an email address and a consent checkbox. A note below the form says: "Don't miss out! Get access to: Cloud Notebooks, Anaconda Assistant, easy application deployment, learning resources, and updates from Anaconda." At the bottom of the form, there's a "Submit >" button and a "Skip registration" link. A small AI bot icon in the bottom right corner says: "You're becoming a regular around here! How can I help you today?"

- After Clicking the link you will see this screen, click skip registration.



## Download Now

For installation assistance, refer to [Troubleshooting](#).

Download Distribution by choosing the proper installer for your machine.



 [Download](#)

### Anaconda Installers



#### Windows

##### Python 3.11

 [64-Bit Graphical Installer \(904.4M\)](#)



#### Mac

##### Python 3.11

 [64-Bit Graphical Installer \(728.7M\)](#)

 [64-Bit Command Line Installer \(731.2M\)](#)

 [64-Bit \(M1\) Graphical Installer \(697.4M\)](#)

 [64-Bit \(M1\) Command Line Installer \(700 M\)](#)



#### Linux

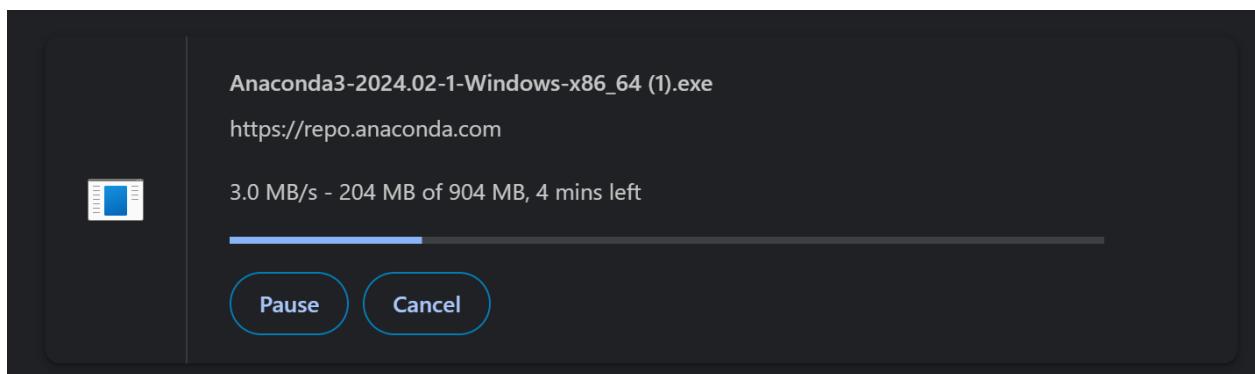
##### Python 3.11

 [64-Bit \(x86\) Installer \(997.2M\)](#)

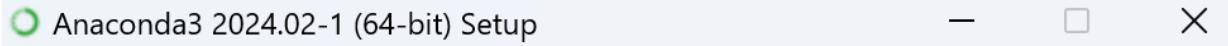
 [64-Bit \(AWS Graviton2 / ARM64\) Installer \(798.5M\)](#)

 [64-bit \(Linux on IBM Z & LinuxONE\) Installer \(91.8M\)](#)

- Then you will see the Above page, Now Click “Download” Button on Anaconda Installer for your Operating system (Windows / Mac / Linux)
- Python is modified to the Necessary version from the Anaconda installer.



- Double click on the above .exe file after downloading it.



## Welcome to Anaconda3 2024.02-1 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2024.02-1 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

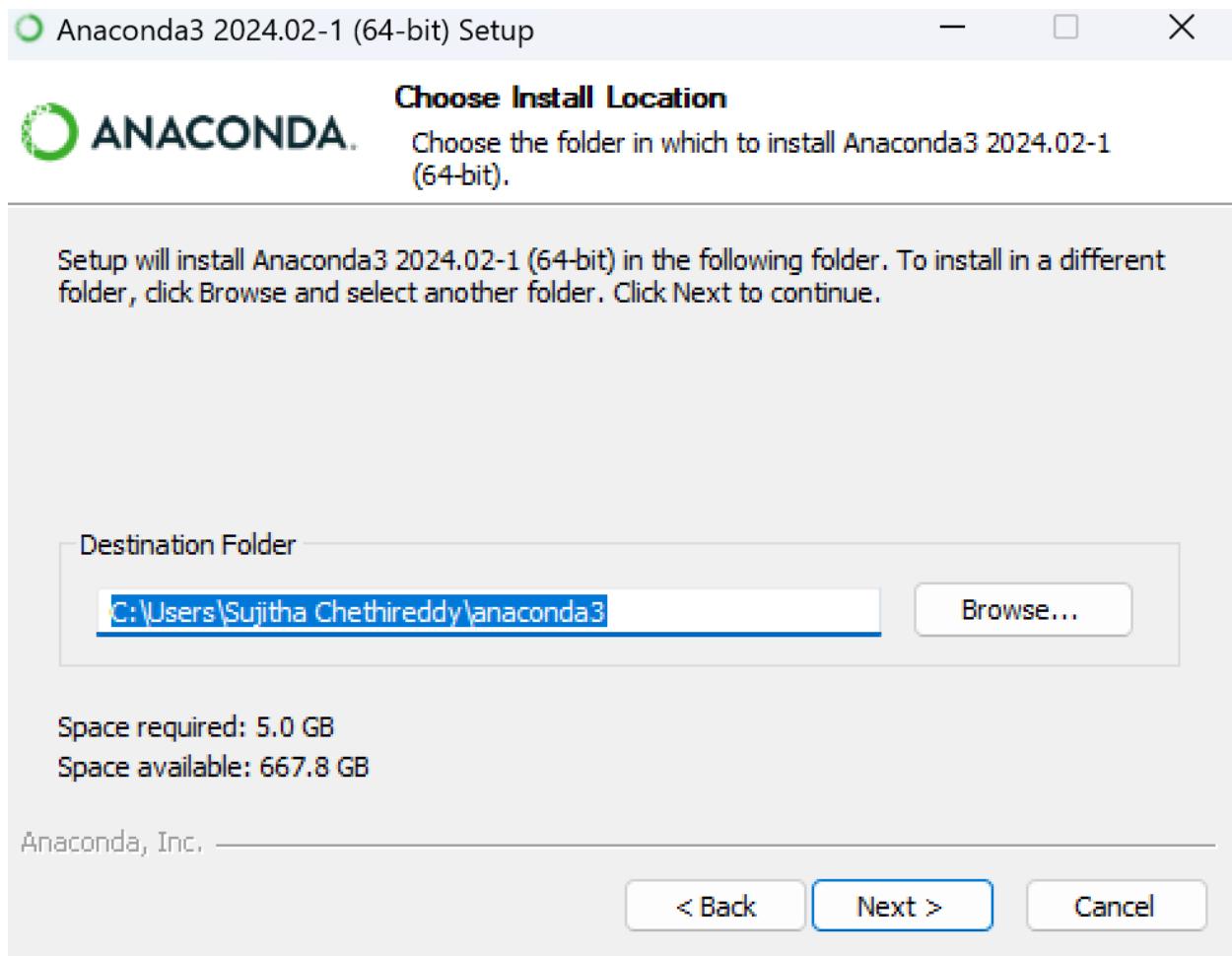
Click Next to continue.



[Next >](#)

[Cancel](#)

- You get this pop up, click Next and Then Agree to License page which comes next.
- Then will You get pop for type of installation you would like to perform Anaconda Install for Just Me or All Users. Select 'Just Me' option.
- Then you will see below pop up, now give your desired file path / location to install and Click Next.



- After That Anaconda is installed in your system. You can open Anaconda prompt shell and run your commands.

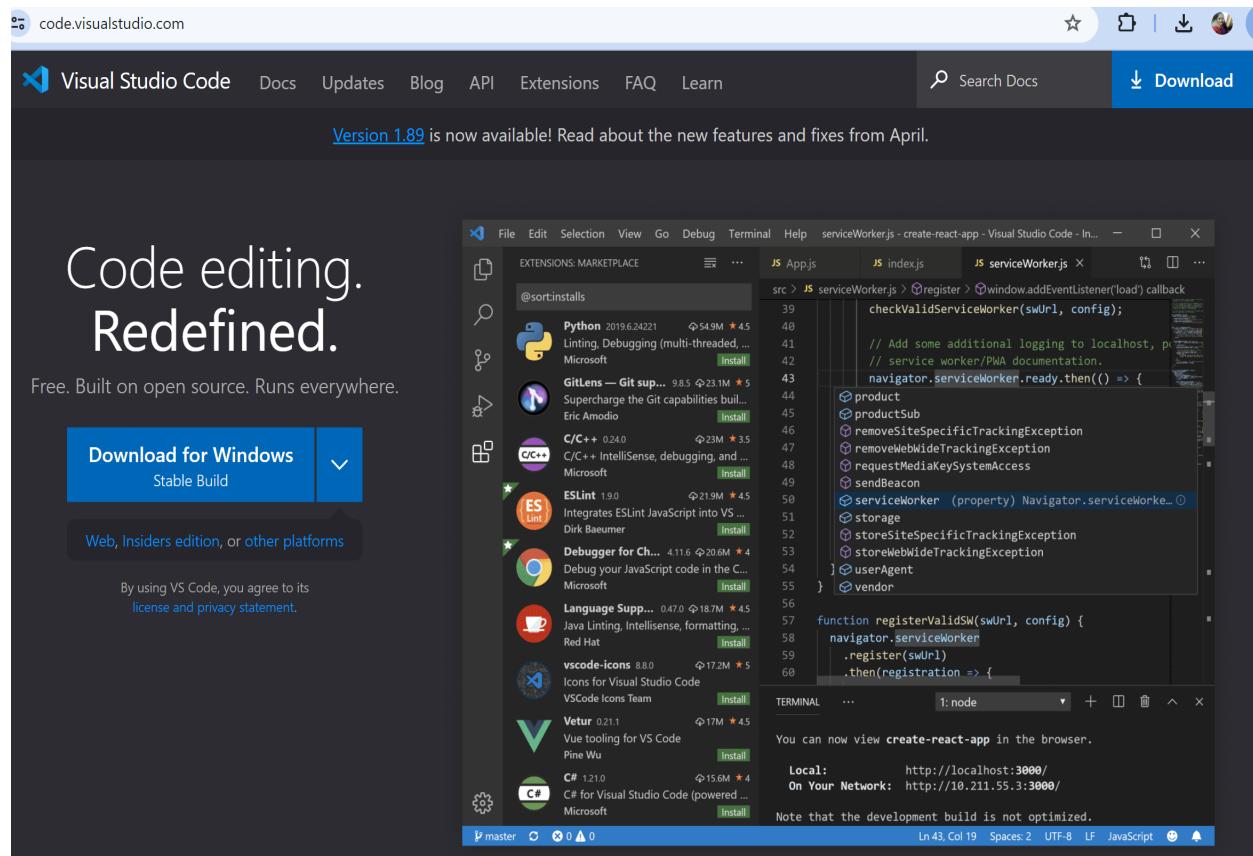
### c. Installing Visual Studio Code (VS Code):

The next step is to Download and install Visual Studio Code to your environment.

Download VS Code:

- Visit the official Visual Studio Code website at

<https://code.visualstudio.com/>



- o Click on the "Download" button to download the installer for your operating system (e.g., Windows, macOS, or Linux).

## Install VS Code:

- Once the installer is downloaded, run the installer executable.
- Follow the on-screen instructions to install VS Code, including accepting the license agreement. (same as the above installations)
- Choose your preferred installation settings and location (you can usually leave the default settings as they are).

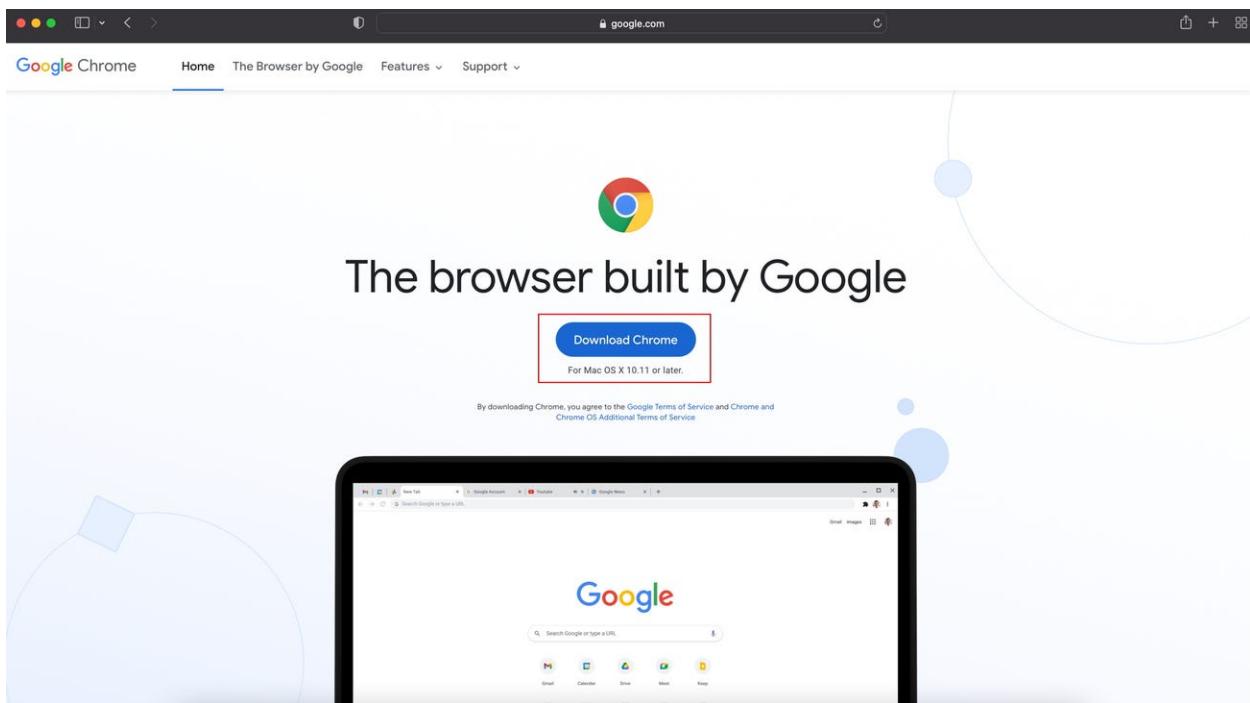
Launch VS Code:

- After installation, you can launch VS Code from your system's application menu or desktop shortcut.

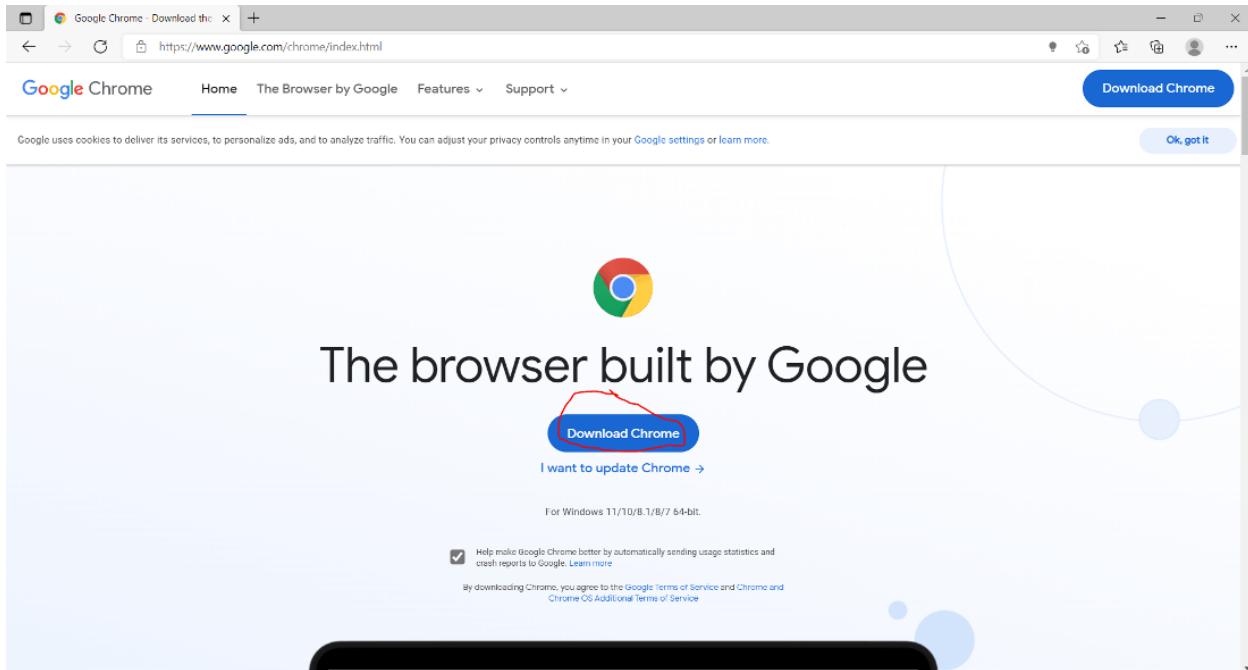
## d. Installing Chrome

The next step is Downloading the Chrome Browser in your system (it makes running of application easy)

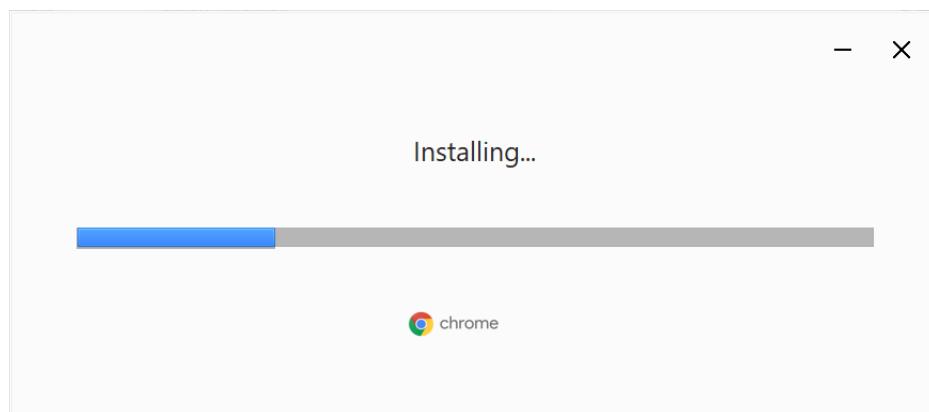
- For Mac OS or Windows: Visit the official website using any web browser.  
[google.com/chrome/](http://google.com/chrome/).



- Once there Click on the “Download Chrome” button, chrome start downloading in your system.



- Now the dialogue box will open then click on Accept and Install. Then you will find the executable file in the downloads folder in your system and run it.
- It will prompt confirmation to make changes to your system. Click on Yes. After this the installation process will start and will take 5 – 7 minutes depending on your Internet speed to complete the installation.



- After downloading it automatically, the Google Chrome browser will open. Be sure to sign in to your Google account so your content can start automatically syncing across devices.

## **5. CLONE OUR GITHUB REPOSITORY**

### **Open a Terminal or Command Prompt:**

- On your local computer, open a terminal (Linux/macOS) or a command prompt (Windows). You'll use this to run Git commands.

### **Navigate to the Directory Where You Want to Clone the Repository:**

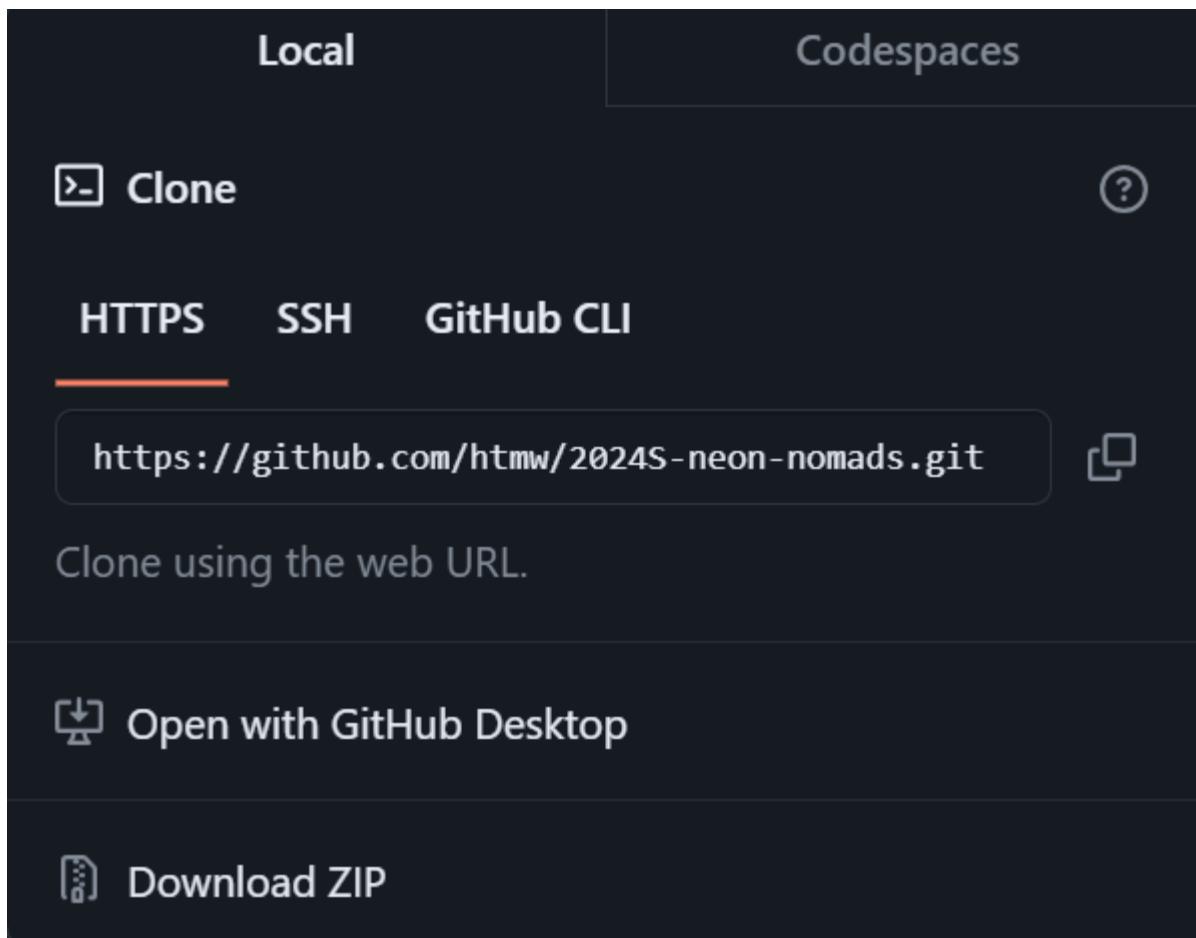
- Use the cd (change directory) command to navigate to the location (desired file path) where you want to clone the repository. For example, to clone the repository in your home directory.

### **Clone the Repository:**

- Use the git clone command followed by the GitHub repository. GitHub repository command will be:

<https://github.com/htmw/2024S-neon-nomads.git>

- Users can either clone our repository by using the above link or download a ZIP of our application. To Download the ZIP, the user has to click on the “Download ZIP” button as shown below. It will download in the user’s specified location. After that, the user has to extract that ZIP.



- Once it is Cloned in your location pull our project from the below link or download that zip file and extract it in your specific location.  
<https://github.com/shashank-sde/Capstonekitchensync>
- Go to the folder where you extracted the code.
- Open the Anaconda PowerShell prompt now.
- Copy that file path/ location where you extracted the code, use the cd command, and paste the location of it in the PowerShell prompt.

cd C:\ user\Sujitha Chethireddy\Desktop\project\Capstonekitchensync (Don't copy this use your own file path/ location)

## 6. STARTING THE SERVER

**Now follow the below commands step by step:**

- ❖ conda create -n "reciperecognizer" python=3.10
  - ❖ conda activate reciperecognizer
  - ❖ pip install -r requirements.txt
- 
- Now you will see this screen like this:

```
(base) PS C:\Users\Sujitha Chethireddy> cd C\Desktop\sujitha_chethireddy\Capstonekitchensync
(base) PS C:\Users\Sujitha Chethireddy\Desktop\sujitha_chethireddy\Capstonekitchensync> conda activate reciperecognizer
(reciperecognizer) PS C:\Users\Sujitha Chethireddy\Desktop\sujitha_chethireddy\Capstonekitchensync> pip install -r requirements.txt
Collecting beautifulsoup4==4.8.2 (from -r requirements.txt (line 1))
  Using cached beautifulsoup4-4.8.2-py3-none-any.whl.metadata (4.1 kB)
Collecting bs4==0.0.1 (from -r requirements.txt (line 2))
  Using cached bs4-0.0.1.tar.gz (1.1 kB)
  Preparing metadata (setup.py) ... done
Collecting Flask==2.0.1 (from -r requirements.txt (line 3))
  Using cached Flask-2.0.1-py3-none-any.whl.metadata (3.8 kB)
Collecting gevent (from -r requirements.txt (line 4))
  Downloading gevent-24.2.1-cp310-cp310-win_amd64.whl.metadata (13 kB)
Collecting Jinja2==3.0 (from -r requirements.txt (line 5))
  Using cached Jinja2-3.0.0-py3-none-any.whl.metadata (3.5 kB)
Collecting numpy (from -r requirements.txt (line 6))
  Downloading numpy-1.26.4-cp310-cp310-win_amd64.whl.metadata (61 kB)
  ██████████ 61.0/61.0 kB 651.3 kB/s eta 0:00:00
Collecting pandas (from -r requirements.txt (line 7))
  Downloading pandas-2.2.2-cp310-cp310-win_amd64.whl.metadata (19 kB)
Collecting Pillow (from -r requirements.txt (line 8))
  Downloading pillow-10.3.0-cp310-cp310-win_amd64.whl.metadata (9.4 kB)
Collecting splinter==0.13.0 (from -r requirements.txt (line 9))
  Using cached splinter-0.13.0.tar.gz (25 kB)
  Preparing metadata (setup.py) ... done
Collecting Werkzeug==2.0 (from -r requirements.txt (line 10))
  Using cached Werkzeug-2.0.0-py3-none-any.whl.metadata (4.4 kB)
Collecting gunicorn==19.0.0 (from -r requirements.txt (line 11))
  Using cached gunicorn-19.0.0.tar.gz (382 kB)
  Preparing metadata (setup.py) ... done
Collecting requests (from -r requirements.txt (line 12))
  Using cached requests-2.31.1-py3-none-any.whl.metadata (13 kB)
```

- Then you will see the following on the PowerShell prompt

```
Anaconda Powershell Prompt + ▾

Downloading click-8.1.2-py3-none-any.whl (96 kB) 96.6/96.6 kB 459.2 kB/s eta 0:00:00
Downloading dnspython-2.2.1-py3-none-any.whl (269 kB) 269.1/269.1 kB 473.2 kB/s eta 0:00:00
Downloading email_validator-1.1.3-py2.py3-none-any.whl (18 kB)
Downloading Flask_Bcrypt-1.0.1-py3-none-any.whl (6.0 kB)
Downloading Flask_Login-0.6.0-py3-none-any.whl (16 kB)
Downloading Flask_SQLAlchemy-2.5.1-py2.py3-none-any.whl (17 kB)
Downloading Flask_WTF-1.0.1-py3-none-any.whl (12 kB)
Downloading greenlet-1.1.2-cp310-cp310-win_amd64.whl (101 kB) 102.0/102.0 kB 490.6 kB/s eta 0:00:00
Downloading idna-3.3-py3-none-any.whl (61 kB) 61.2/61.2 kB 462.9 kB/s eta 0:00:00
Downloading importlib_metadata-4.11.3-py3-none-any.whl (18 kB)
Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Downloading MarkupSafe-2.1.1-cp310-cp310-win_amd64.whl (17 kB)
Downloading pycparser-2.21-py3-none-any.whl (118 kB) 118.7/118.7 kB 257.0 kB/s eta 0:00:00
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading SQLAlchemy-1.4.35-cp310-cp310-win_amd64.whl (1.6 MB) 1.6/1.6 MB 329.2 kB/s eta 0:00:00
Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB) 42.3/42.3 kB 1.0 kB/s eta 0:00:00
Downloading WTForms-3.0.1-py3-none-any.whl (136 kB) 136.5/136.5 kB 537.1 kB/s eta 0:00:00
Downloading zipp-3.8.0-py3-none-any.whl (5.4 kB)
Downloading gevent-21.12.0-cp310-cp310-win_amd64.whl (1.6 MB) 1.6/1.6 MB 350.3 kB/s eta 0:00:00
Downloading numpy-1.26.4-cp310-cp310-win_amd64.whl (15.8 MB) 9.4/15.8 MB 448.3 kB/s eta 0:00:15
```

```
Anaconda Powershell Prompt + ▾

ca96b3e2d422ef665f6b43f6f45ab119b7
Stored in directory: c:\users\suji thira chethireddy\appdata\local\pip\cache\wheels\ae\5f\98\9fc9854ebab023ddc97e516294b5
e870bd0c7ccfdef98fba49
Building wheel for gunicorn (setup.py) ... done
Created wheel for gunicorn: filename=gunicorn-19.0.0-py2.py3-none-any.whl size=100331 sha256=bfa0f1748baea4d5b6bf32d7
4e671dd27628ac2301490dd49b2d402b3d32e51
Stored in directory: c:\users\suji thira chethireddy\appdata\local\pip\cache\wheels\9d\17\6c\de711df8c78a9e7b76f4a9fd3ff7
399bda6b0a3418c3c5c918
Successfully built bs4 splinter gunicorn
Installing collected packages: sortedcontainers, pytz, namex, libclang, gunicorn, flatbuffers, zope.interface, zope.event,
zipp, wrapt, Werkzeug, urllib3, tzdata, typing-extensions, termcolor, tensorflow-io-gcs-filesystem, tensorboard-data-
server, sqlparse, soupsieve, sniffio, six, pysocks, pygments, pycparser, protobuf, Pillow, packaging, numpy, mdurl, Mark
upSafe, markdown, itsdangerous, idna, h11, grpcio, greenlet, gast, exceptiongroup, dnspython, colorama, charset-normaliz
er, certifi, attrs, asgiref, absl-py, WTForms, wsproto, tensorboard, SQLAlchemy, requests, python-dateutil, outcome, opt
ree, opt-einsum, ml-dtypes, markdown-it-py, Jinja2, importlib-metadata, h5py, google-pasta, email-validator, click, ffi
, BeautifulSoup4, astunparse, trio, rich, pandas, gevent, Flask, bs4, bcrypt, trio-websocket, keras, Flask-WTF, Flask-SQ
LAlchemy, Flask-Login, Flask-Bcrypt, tensorflow-intel, selenium, tensorflow, splinter
Successfully installed Flask-2.0.1 Flask-Bcrypt-1.0.1 Flask-Login-0.6.0 Flask-SQLAlchemy-2.5.1 Flask-WTF-1.0.1 Jinja2-3.
0.0 MarkupSafe-2.1.1 Pillow-10.3.0 SQLAlchemy-1.4.35 WTForms-3.0.1 Werkzeug-2.0.0 absl-py-2.1.0 asgiref-3.5.0 astunparse
-1.6.3 attrs-23.2.0 bcrypt-3.2.0 BeautifulSoup4-4.8.2 bs4-0.0.1 certifi-2024.2.2 cffi-1.15.0 charset-normalizer-3.3.2 cl
ick-8.1.2 colorama-0.4.6 dnspython-2.2.1 email-validator-1.1.3 exceptiongroup-1.2.1 flatbuffers-24.3.25 gast-0.5.4 geven
t-21.12.0 google-pasta-0.2.0 greenlet-1.1.2 grpcio-1.62.2 gunicorn-19.0.0 h11-0.14.0 h5py-3.11.0 idna-3.3 importlib-meta
data-4.11.3 itsdangerous-2.1.2 keras-3.3.2 libclang-18.1.1 markdown-3.6 markdown-it-py-3.0.0 mdurl-0.1.2 ml-dtypes-0.3.2
namex-0.0.8 numpy-1.26.4 opt-einsum-3.3.0 optree-0.11.0 outcome-1.3.0.post0 packaging-24.0 pandas-2.2.2 protobuf-4.25.3
pycparser-2.21 pygments-2.17.2 pysocks-1.7.1 python-dateutil-2.9.0.post0 pytz-2024.1 requests-2.31.0 rich-13.7.1 seleni
um-4.20.0 six-1.16.0 sniffio-1.3.1 sortedcontainers-2.4.0 soupsieve-2.5 splinter-0.13.0 sqlparse-0.4.2 tensorboard-2.16.
2 tensorboard-data-server-0.7.2 tensorflow-2.16.1 tensorflow-2.16.1 tensorflow-io-gcs-filesystem-0.31.0 termcolor-
2.4.0 trio-0.25.0 trio-websocket-0.11.1 typing-extensions-4.11.0 tzdata-2024.1 urllib3-2.2.1 wrapt-1.16.0 wsproto-1.2.0
zipp-3.8.0 zope.event-5.0 zope.interface-6.3
(reciperecognizer) PS C:\Users\Sujitha Chethireddy\Desktop\suji thira_chethireddy\Capstonekitchensync>
```

## ❖ pip install flask-migrate

```
(reciperecognizer) PS C:\Users\Sujitha Chethireddy\Desktop\sujitha_chethireddy\Capstonekitchensync> pip install flask-migrate
Collecting flask-migrate
  Downloading Flask_Migrate-4.0.7-py3-none-any.whl.metadata (3.1 kB)
Requirement already satisfied: Flask>=0.9 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from flask-migrate) (2.0.1)
Requirement already satisfied: Flask-SQLAlchemy>=1.0 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from flask-migrate) (2.5.1)
Collecting alembic>=1.9.0 (from flask-migrate)
  Downloading alembic-1.13.1-py3-none-any.whl.metadata (7.4 kB)
Requirement already satisfied: SQLAlchemy>=1.3.0 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from alembic>=1.9.0>flask-migrate) (1.4.35)
Collecting Mako (from alembic>=1.9.0->flask-migrate)
  Downloading Mako-1.3.3-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: typing-extensions>=4 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from alembic>=1.9.0->flask-migrate) (4.11.0)
Requirement already satisfied: Werkzeug>=2.0 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Flask>=0.9->flask-migrate) (2.0.0)
Requirement already satisfied: Jinja2>=3.0 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Flask>=0.9->flask-migrate) (3.0.0)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Flask>=0.9->flask-migrate) (2.1.2)
Requirement already satisfied: click>=7.1.2 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Flask>=0.9->flask-migrate) (8.1.2)
```

## ❖ python app.py

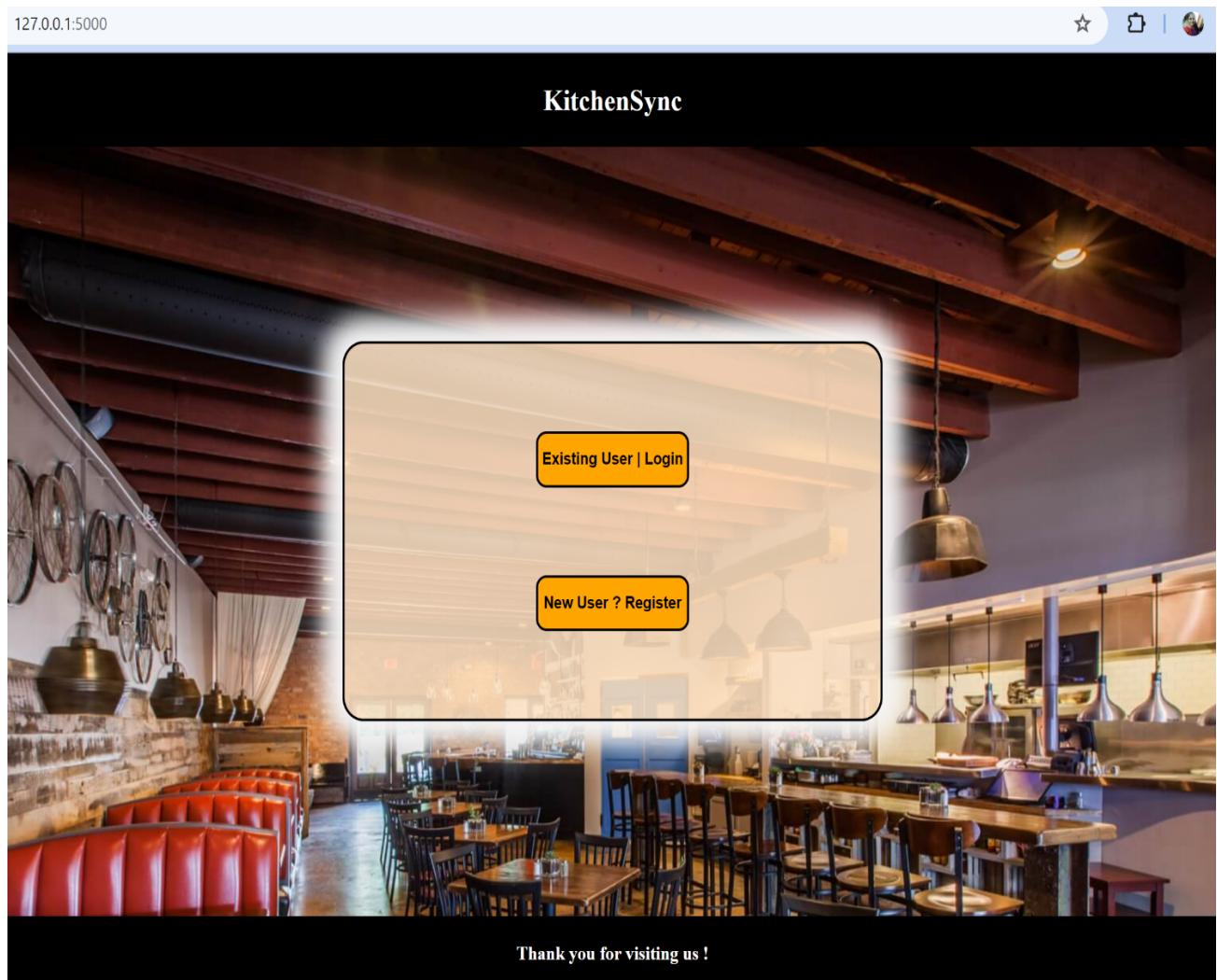
```
R>migrate) (3.0.0)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Flask>=0.9->flask-migrate) (2.1.2)
Requirement already satisfied: click>=7.1.2 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Flask>=0.9->flask-migrate) (8.1.2)
Requirement already satisfied: colorama in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from click>=7.1.2->Flask>=0.9->flask-migrate) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0.0rc2 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from Jinja2>=3.0->Flask>=0.9->flask-migrate) (2.1.1)
Requirement already satisfied: greenlet>=0.4.17 in c:\users\sujitha_chethireddy\new_folder\envs\reciperecognizer\lib\site-packages (from SQLAlchemy>=1.3.0->alembic>=1.9.0->flask-migrate) (1.1.2)
Downloading Flask_Migrate-4.0.7-py3-none-any.whl (21 kB)
Downloading alembic-1.13.1-py3-none-any.whl (233 kB)
  233.4/233.4 kB 2.0 MB/s eta 0:00:00
Downloaded alembic-1.13.1-py3-none-any.whl (78 kB)
  78.8/78.8 kB 4.3 MB/s eta 0:00:00
Installing collected packages: Mako, alembic, flask-migrate
Successfully installed Mako-1.3.3 alembic-1.13.1 flask-migrate-4.0.7
(reciperecognizer) PS C:\Users\Sujitha Chethireddy\Desktop\sujitha_chethireddy\Capstonekitchensync> python app.py
2024-04-25 17:36:54.614113: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2024-04-25 17:36:56.054495: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
C:\Users\Sujitha Chethireddy\New folder\envs\reciperecognizer\lib\site-packages\flask_sqlalchemy\__init__.py:851: UserWarning: Neither SQLALCHEMY_DATABASE_URI nor SQLALCHEMY_BINDS is set. Defaulting SQLALCHEMY_DATABASE_URI to "sqlite:///memory:".
  warnings.warn(
C:\Users\Sujitha Chethireddy\New folder\envs\reciperecognizer\lib\site-packages\flask_sqlalchemy\__init__.py:872: FSDeprecationWarning: SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
  warnings.warn(FSDeprecationWarning(
2024-04-25 17:37:00.066076: I tensorflow/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
C:\Users\Sujitha Chethireddy\New folder\envs\reciperecognizer\lib\site-packages\keras\src\optimizers\base_optimizer.py:32: UserWarning: Argument 'decay' is no longer supported and will be ignored.
  warnings.warn(
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile_metrics' will be empty until you train or evaluate the model.
WARNING:absl:Error in loading the saved optimizer state. As a result, your model is starting with a freshly initialized optimizer.
Model loaded successfully !! Check http://127.0.0.1:5000/
* 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.
```

## 7. RUNNING THE APPLICATION

```
C:\Users\Sujitha Chethireddy\New folder\envs\reciperecognizer\lib\site-packages\flask_sqlalchemy\__init__.py:872: FSADeprecationWarning: SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
  warnings.warn(FSADeprecationWarning)
2024-04-25 17:37:00.066076: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
C:\Users\Sujitha Chethireddy\New folder\envs\reciperecognizer\lib\site-packages\keras\src\optimizers\base_optimizer.py:32: UserWarning: Argument 'decay' is no longer supported and will be ignored.
  warnings.warn(
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile_metrics' will be empty until you train or evaluate the model.
WARNING:absl:Error in loading the saved optimizer state. As a result, your model is starting with a freshly initialized optimizer.
Model loaded successfully !! Check http://127.0.0.1:5000/
* 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: on
```

- Then its shows loaded Successfully and give check ‘Link’
- Now open Chrome Browser and paste the Link:  
<http://127.0.0.1:5000/>
- Enter the URL box of Chrome Brower and hit the enter key. Now User can run our application.

## 8. RESULTS



- You will get this screen when you run the application.

# Manage Your Inventory



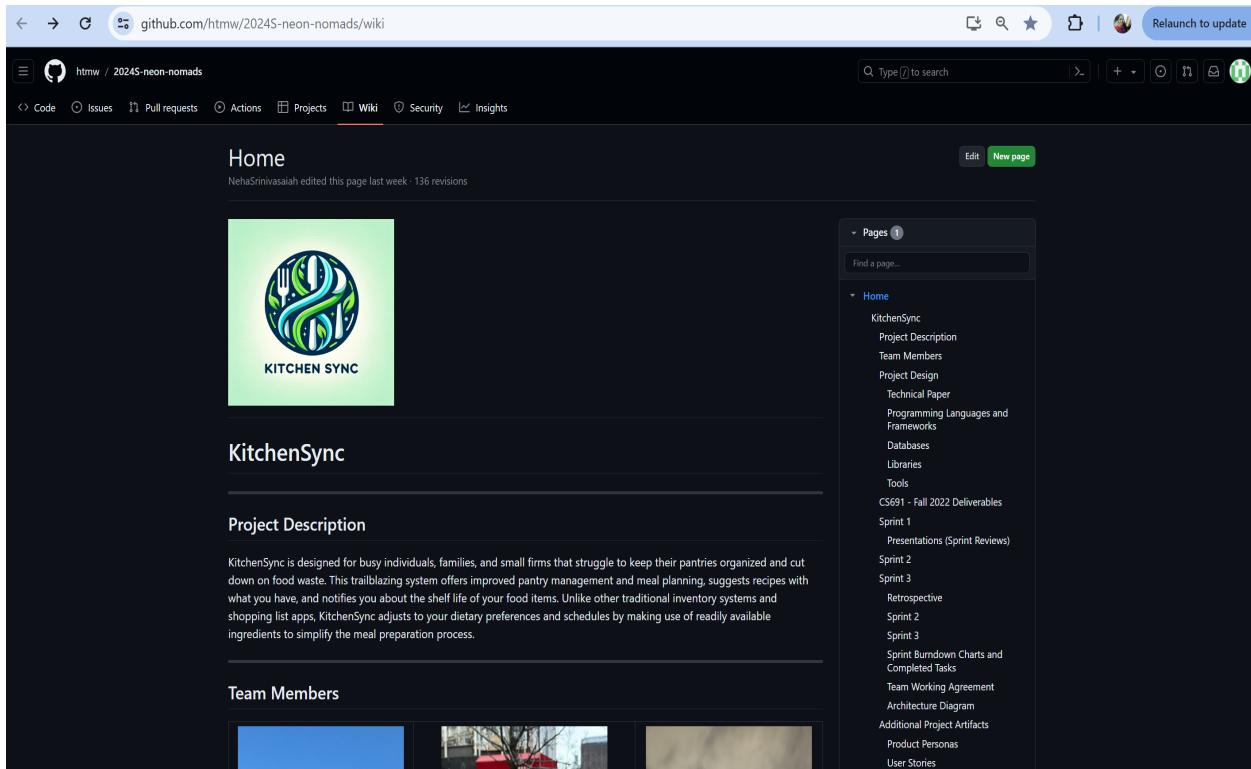
[Press here to logout](#)

## Your Inventory List :

Item Name	Expiry Date	
tomato	2024-05-02	<input type="button" value="Delete"/>
capsicum	2024-05-02	<input type="button" value="Delete"/>
orange	2024-04-30	<input type="button" value="Delete"/>
banana	2024-04-30	<input type="button" value="Delete"/>
onions	2024-05-08	<input type="button" value="Delete"/>
apple	2024-05-04	<input type="button" value="Delete"/>

- This is how User's Inventory List will be after they enter their Pantry Items and its Expiry Dates.

# 9. GITHUB WIKI



<https://github.com/htmw/2024S-neon-nomads/wiki>

This is the Link to Our GitHub Wiki Page. The user can understand more about our KitchenSync Application from here. Here we have provided the User Manual to Comprehend each functionality and every feature of Our application and provided all the documents for a better understanding of each aspect of our application. Also, we have provided a Demo video of KitchenSync to Understand Step by step use of our application.