**INTRODUCTION**

**OVERVIEW:**

Chatbots are conversational tools that perform routine tasks efficiently. People like them because they help them get through those tasks quickly so they can focus their attention on high-level, strategic, and engaging activities that require human capabilities that cannot be replicated by machines.

**PURPOSE:**

The purpose of public welfare chatbot is to enhance public access to essential services and information, promote efficiency in government operations, and improve the well-being of individuals and communities.

**PROPOSED SOLUTION:**

The proposed solution involves leveraging IBM Watson Assistant in combination with Python to develop an intelligent chatbot. It will be designed to assist users with inquiries, provide product information, and route complex issues to human agents. The chatbot will use Watson Assistant's natural language processing capabilities to understand user intents and entities, and Python will facilitate seamless communication between users and the chatbot. The solution includes a robust escalation mechanism to transfer users to human agents when necessary, and it will be continuously improved through testing, training, and monitoring for optimal customer support.

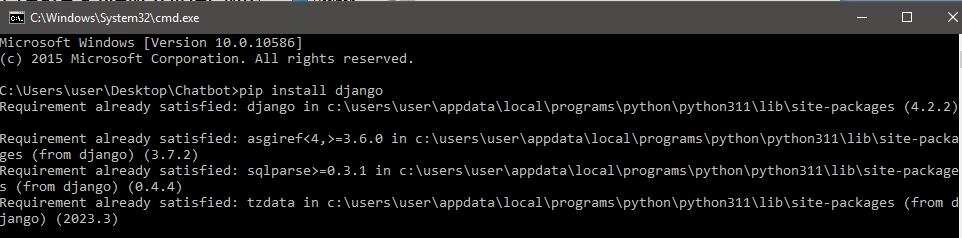
**PROJECT CREATION , PATH SETUP AND APP**

**CREATION IN COMMAND PROMPT**

1.Create a New folder with a name

2.Get into folder and open command prompt

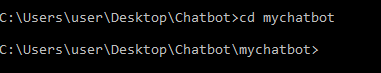
3.Django installation



4.Project creation



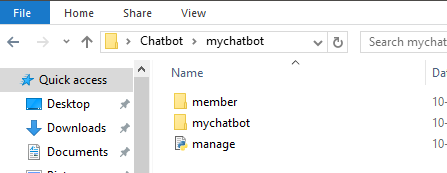
5.Path setup



6.App creation



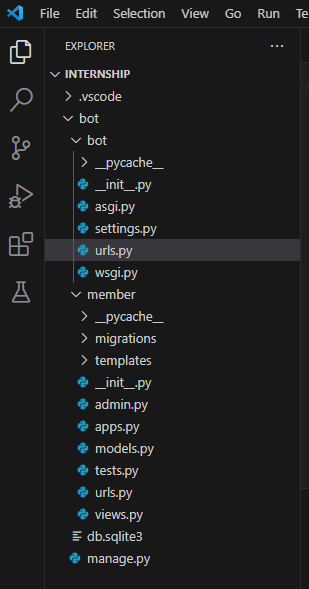
7.Check app and project are created inside the folder



**IMPORTING INCLUDE() FUNCTION AND ADDING**

**URLPATTERNS**

1.Open visual studio code and check whether these are created inside the folder



2.Import the function and add urlpatterns in the project urls.py

from django.contrib import admin

from django.urls import include,path

urlpatterns = [

    path('',include('member.urls')),

    path('admin/',admin.site.urls),

]

3. Import the function and add urlpatterns in the app urls.py

from django.urls import path

from . import views

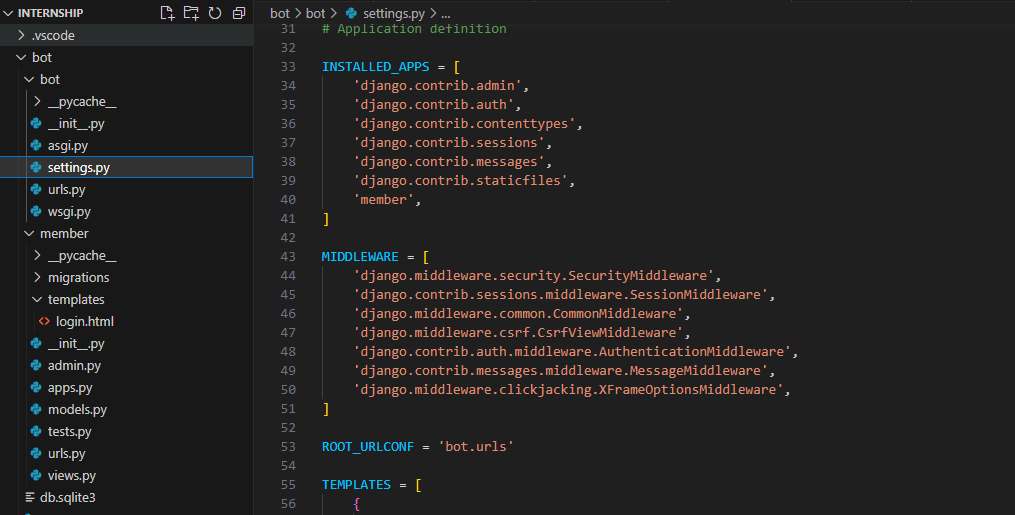
from django.contrib import admin

urlpatterns=[

    path('',views.index),

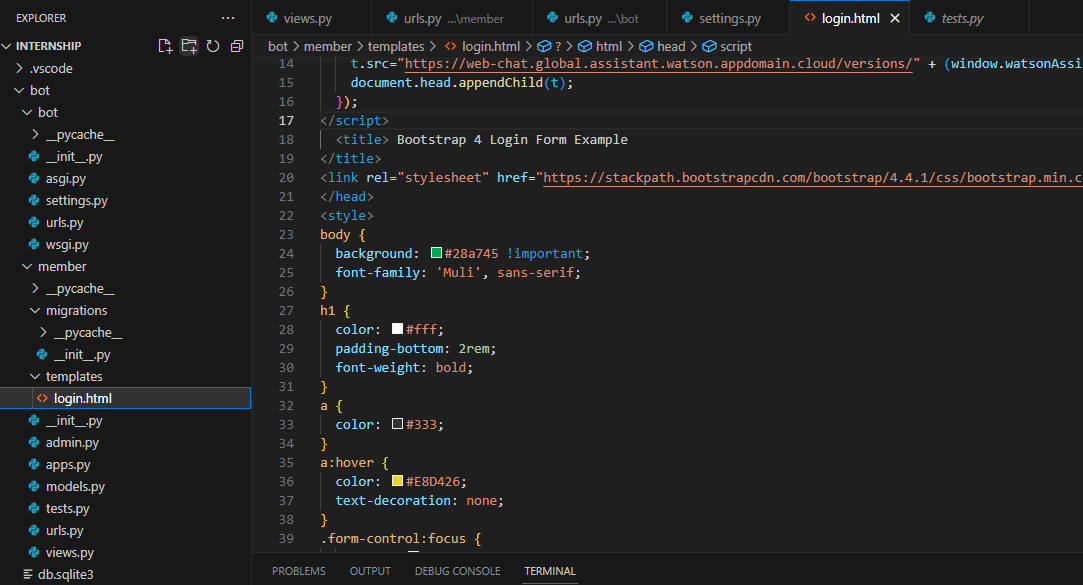
]

4.Add the app name in the settings.py

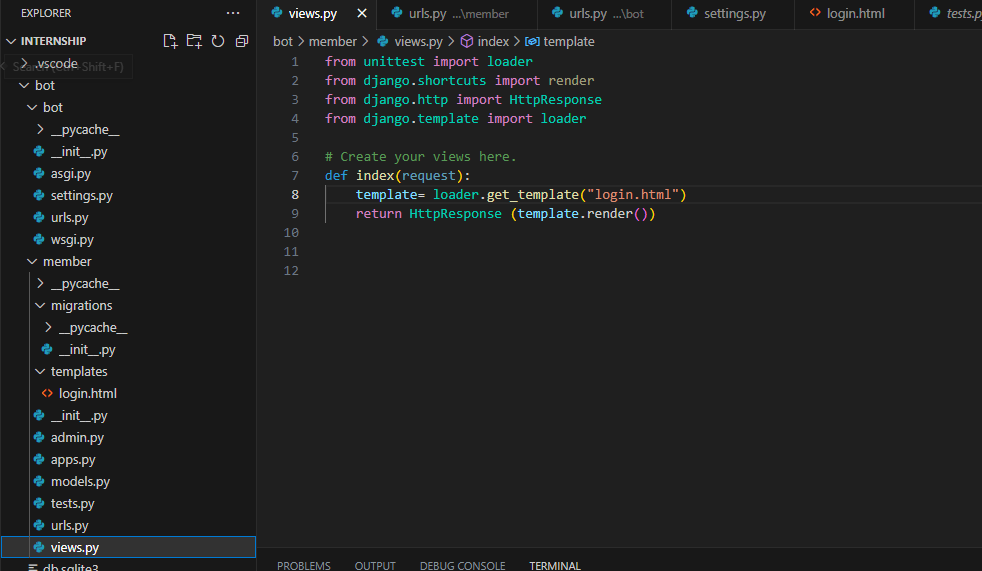


**CREATING VIEWS FOR LOGIN PAGE CODE**

1.Create a file template inside the app and add the login page code



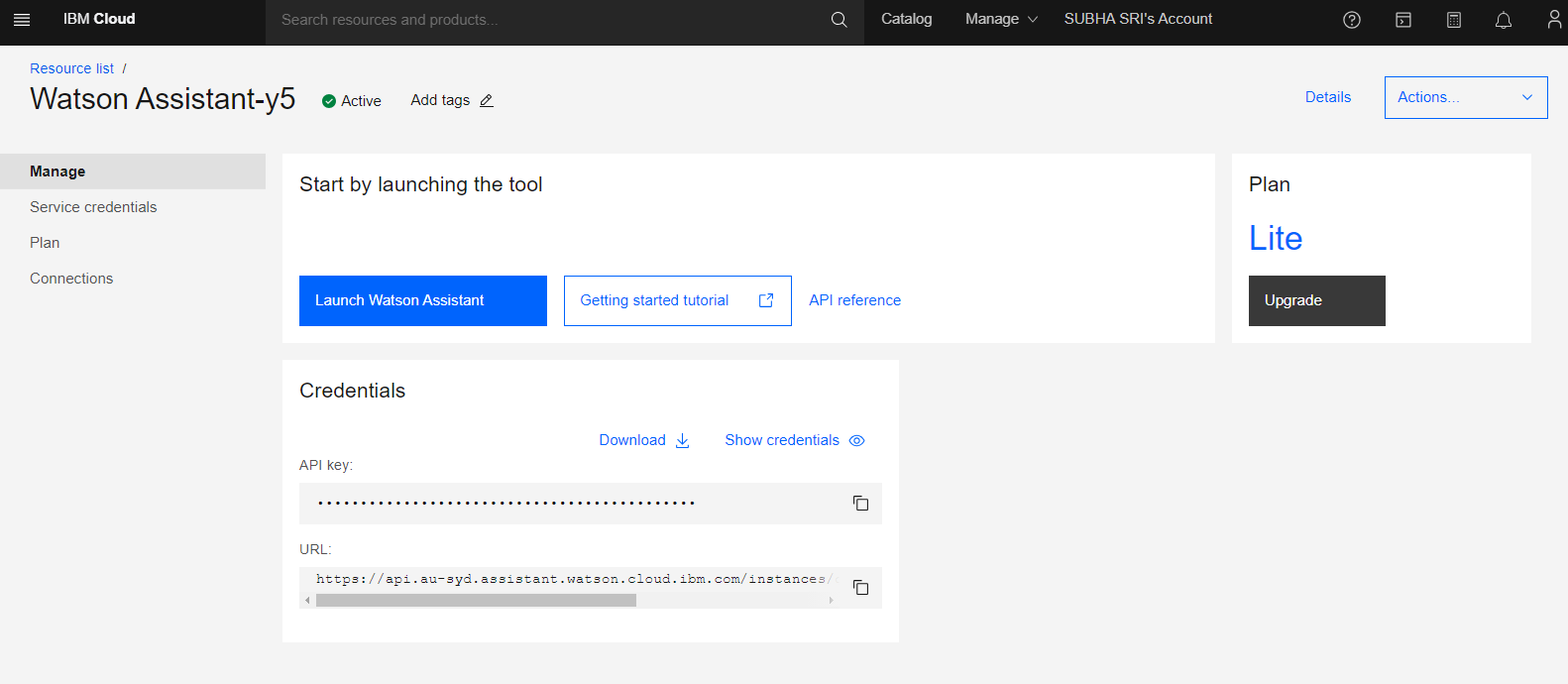
2.Create the views



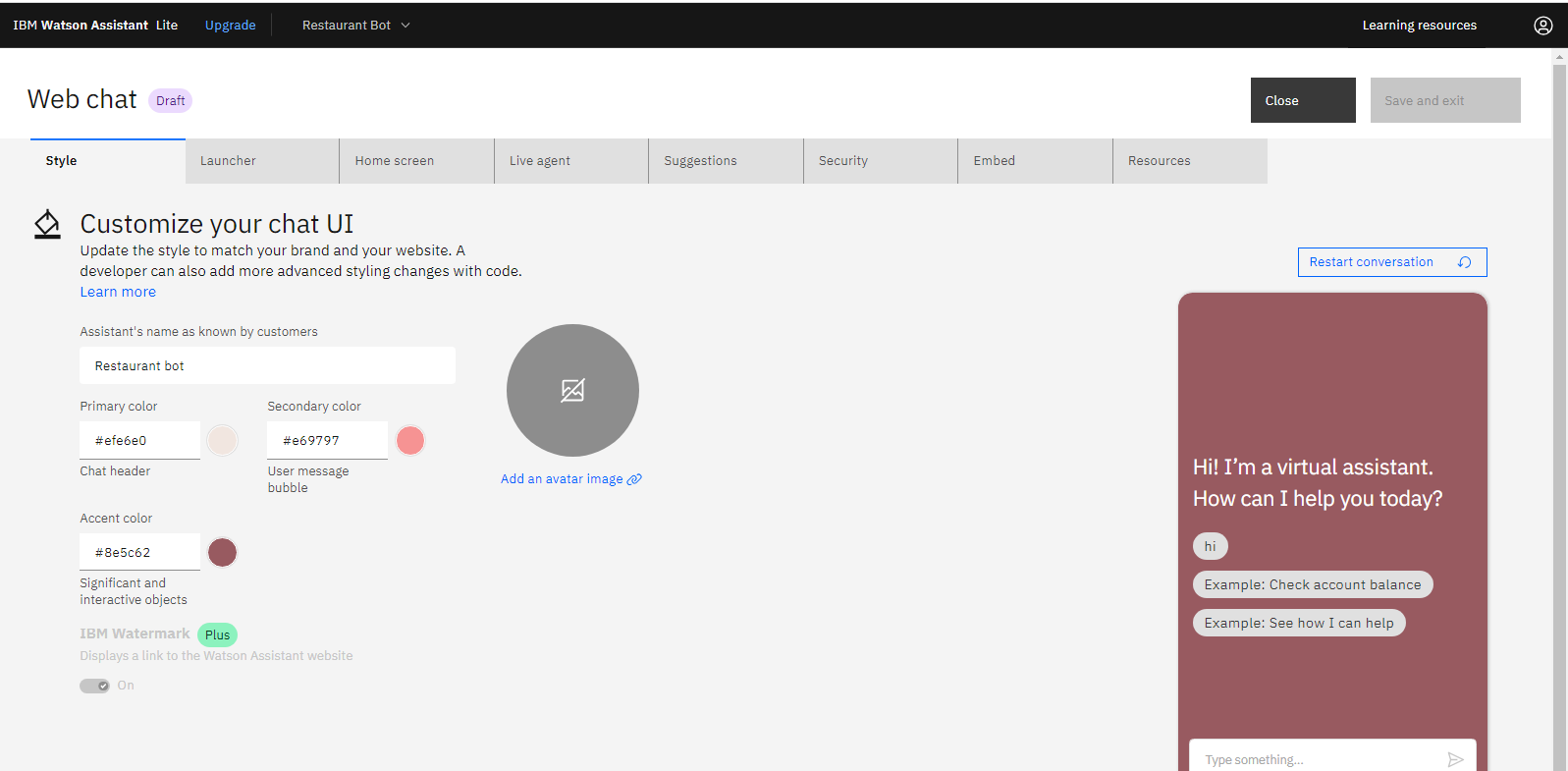
**CHATBOT CREATION THROUGH WATSON**

**ASSISTANT CATALOG**

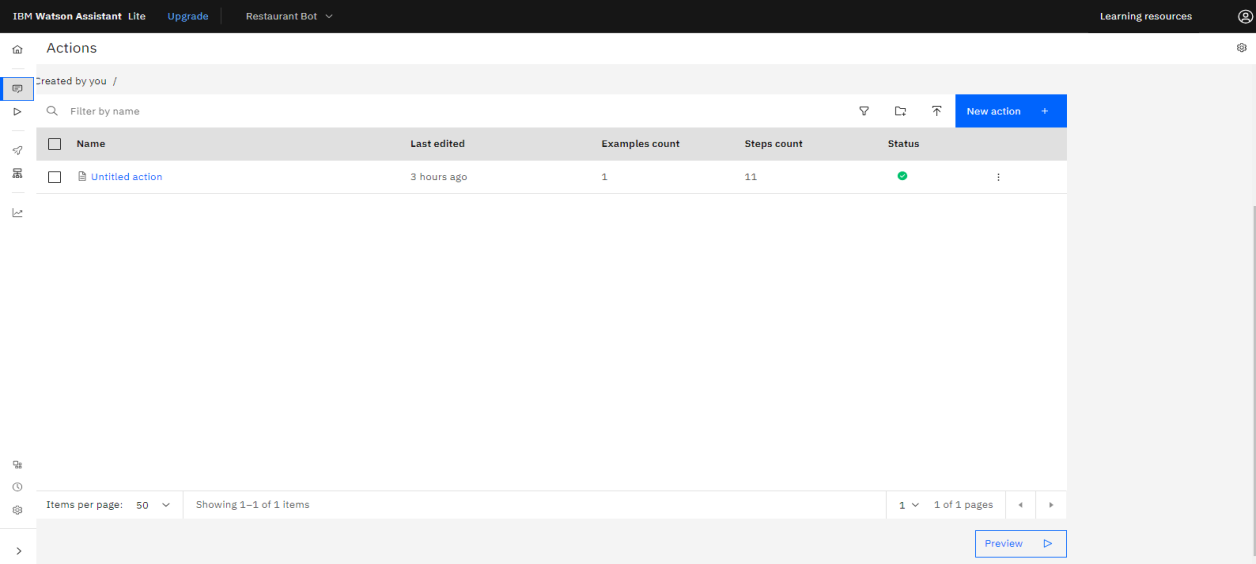
1.Login into your cloud account and search Watson Assistant in Catalog and launch it and create actions.



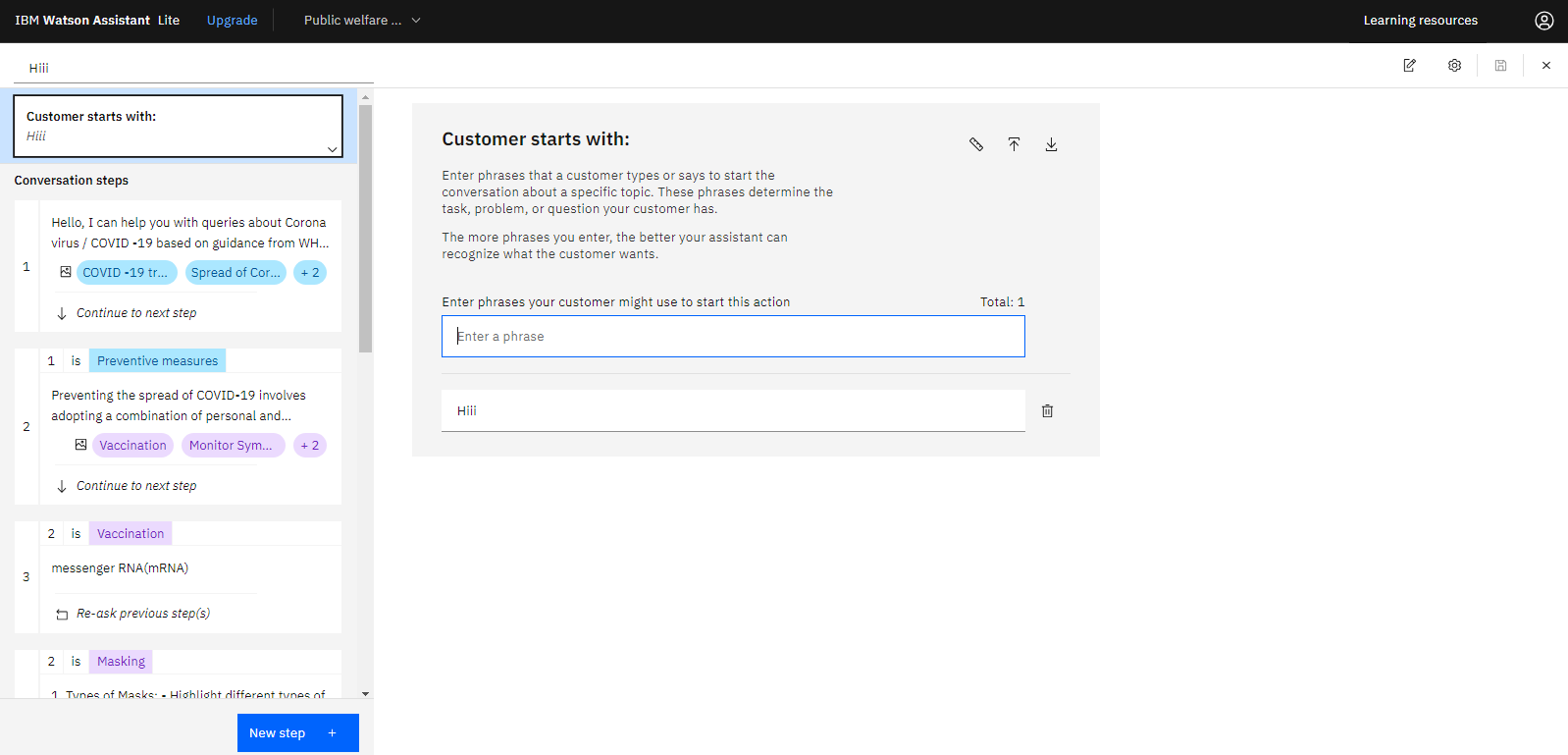
2.Customize your chatbot



3.Select Actions

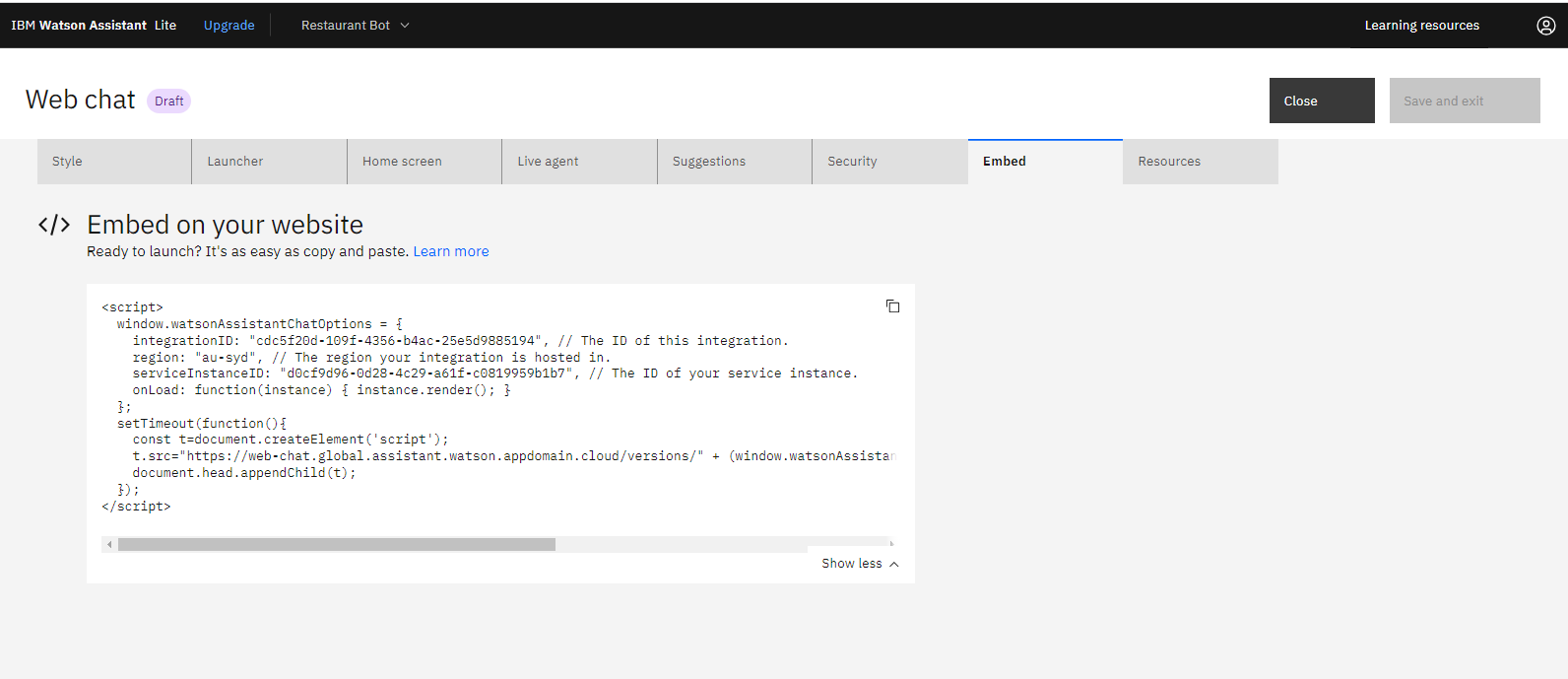


4.Insert the actions for your chatbot

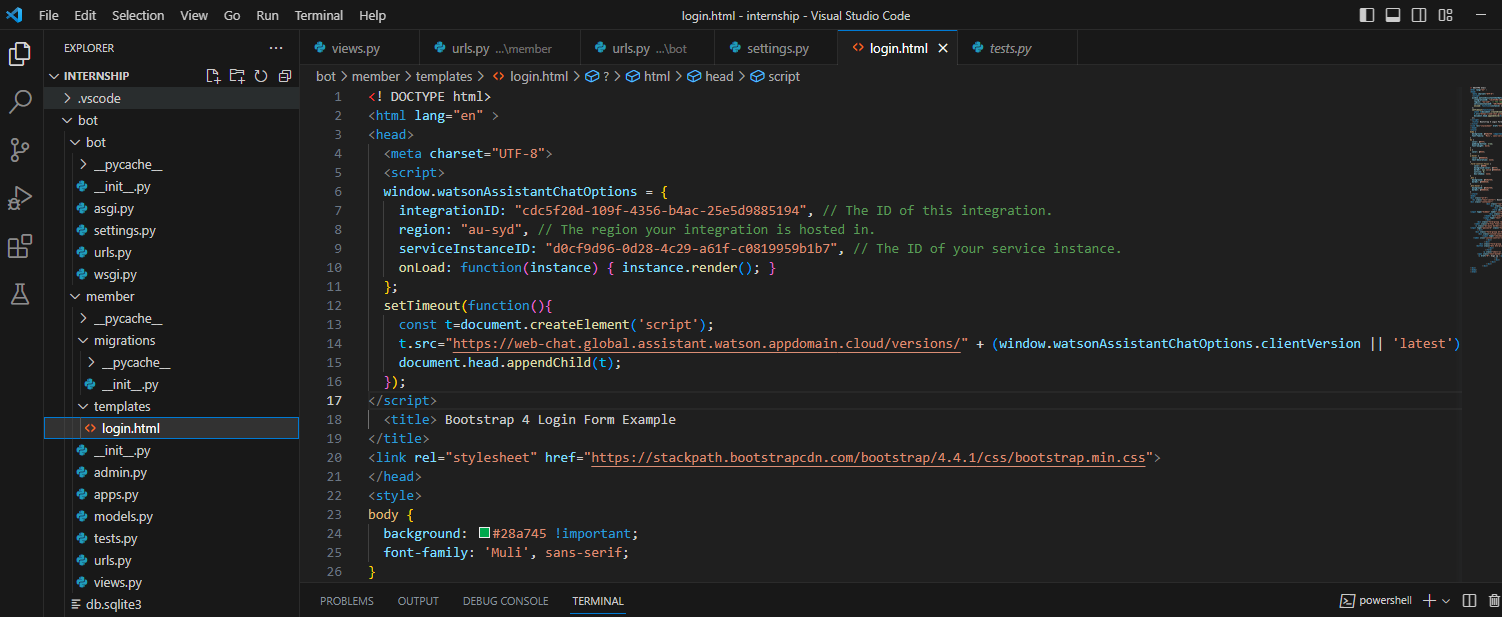


**INTEGRATING THE CHATBOT**

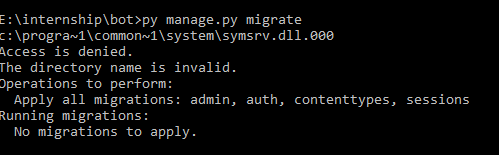
1.Copy the code from embed



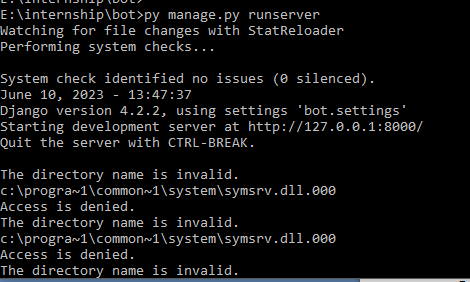
2.Paste it in your home page code



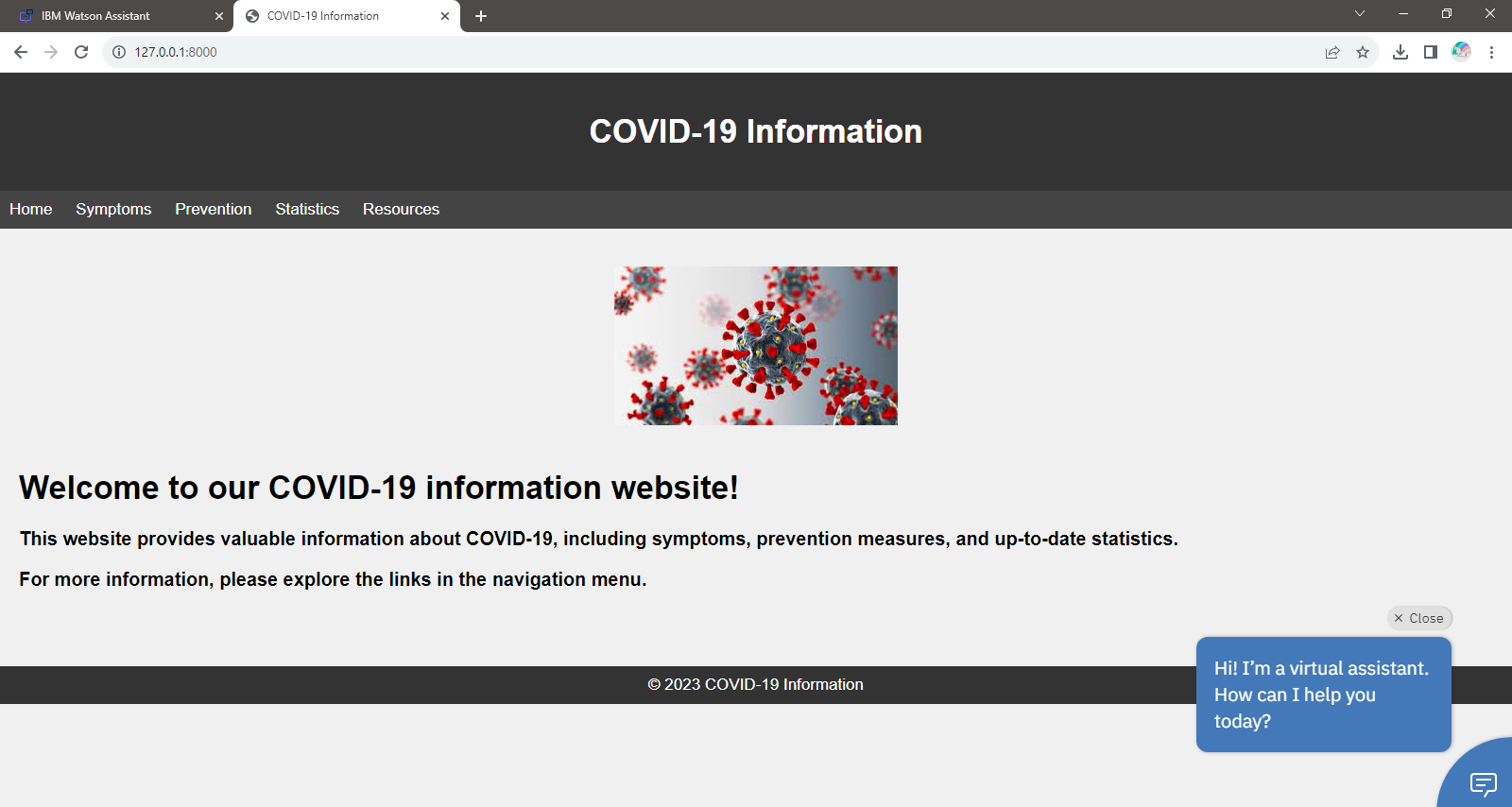
3..Migrate it



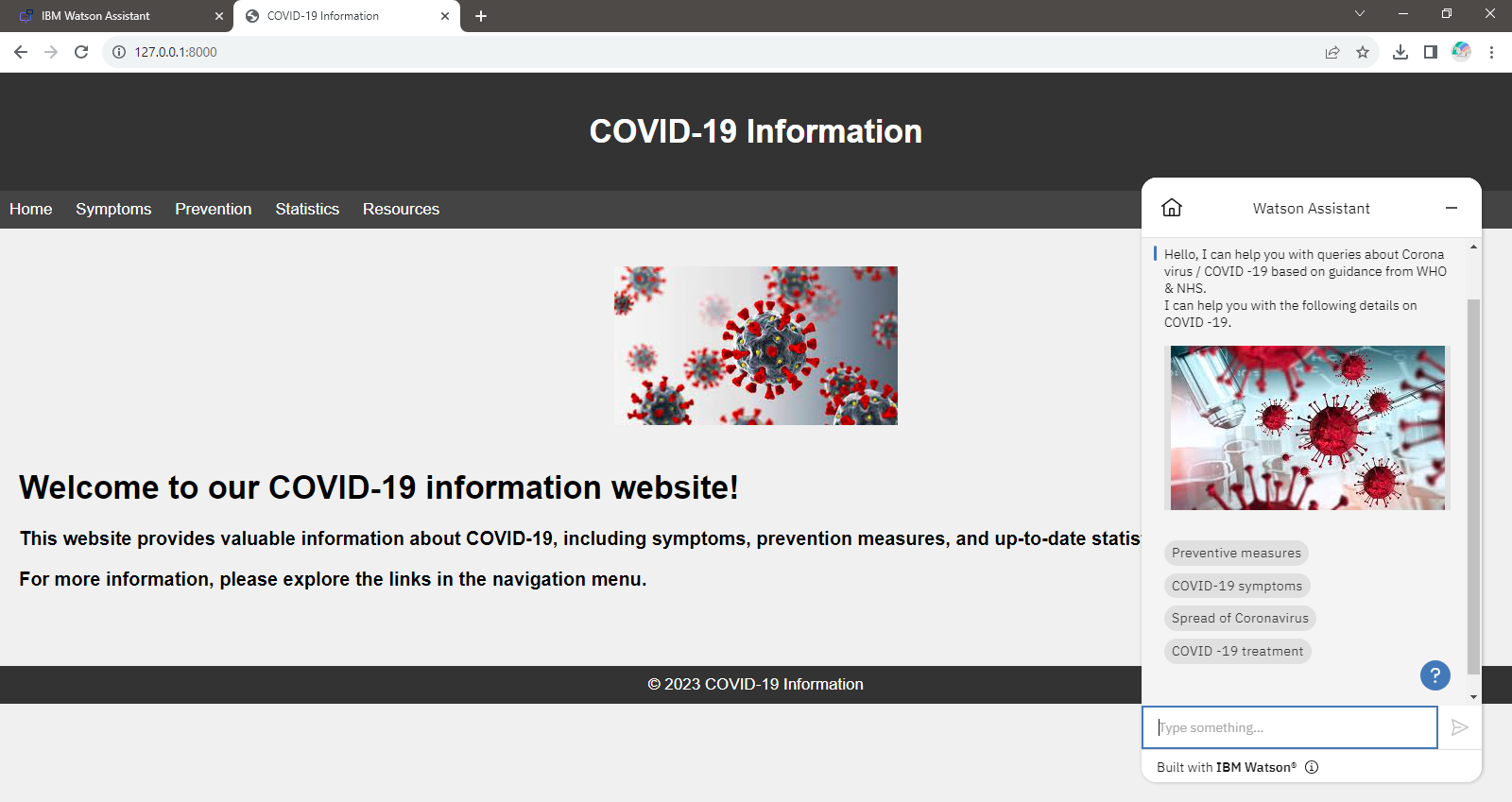
4..Runserver



5..Copy the http code and run it in the browser



**FINAL RESULT**

****

**APPENDIX**:

**SOURCE CODE:**

**urls.py**

"""

URL configuration for bot project.

The `urlpatterns` list routes URLs to views. For more information please see:

    https://docs.djangoproject.com/en/4.2/topics/http/urls/

Examples:

Function views

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

Class-based views

    1. Add an import:  from other\_app.views import Home

    2. Add a URL to urlpatterns:  path('', Home.as\_view(), name='home')

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import include,path

urlpatterns = [

    path('',include('member.urls')),

    path('admin/',admin.site.urls),

]

**settings.py**

"""

Django settings for bot project.

Generated by 'django-admin startproject' using Django 4.2.2.

For more information on this file, see

https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/4.2/ref/settings/

"""

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-!aga#hcx4$&f3i-k4ltxnp&2=oshp85@dg9zj2i7)r3al(44dw'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'member',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'bot.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'bot.wsgi.application'

# Database

# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

# Password validation

# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/4.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC\_URL = 'static/'

# Default primary key field type

# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

**App creation:**

**urls.py**

from django.urls import path

from . import views

from django.contrib import admin

urlpatterns=[

    path('',views.index),

]

**views.py**

from unittest import loader

from django.shortcuts import render

from django.http import HttpResponse

from django.template import loader

# Create your views here.

def index(request):

    template= loader.get\_template("login.html")

    return HttpResponse (template.render())

**login page code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>COVID-19 Information</title>

    <style>

        body {

            background-color: #f0f0f0; /\* Set your desired background color \*/

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

        }

        header {

            background-color: #333; /\* Header background color \*/

            color: #fff; /\* Header text color \*/

            text-align: center;

            padding: 20px;

        }

        nav {

            background-color: #444; /\* Navigation background color \*/

            padding: 10px;

        }

        nav a {

            color: #fff; /\* Navigation link color \*/

            text-decoration: none;

            margin-right: 20px;

        }

        main {

            padding: 20px;

        }

        footer {

            background-color: #333; /\* Footer background color \*/

            color: #fff; /\* Footer text color \*/

            text-align: center;

            padding: 10px;

        }

    </style>

</head>

<body>

    <header>

        <h1>COVID-19 Information</h1>

    </header>

    <nav>

        <a href="https://www.who.int/health-topics/coronavirus">Home</a>

        <a href="https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html">Symptoms</a>

        <a href="https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html">Prevention</a>

        <a href="https://www.worldometers.info/coronavirus/">Statistics</a>

        <a href="https://en.wikipedia.org/wiki/Coronavirus">Resources</a>

    </nav><br><br>

    <div class="img">

                <center><img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRkMdJQO2w8aY2FhtKv953GjKg6Tn1VV8ZgcSvpSxOsHDGpDplYXqhzBsmWHNrIkdCSih0&usqp=CAU" class="w-100" alt="">

            </center></div>

    <main>

        <h1>Welcome to our COVID-19 information website!</h1>

        <p><h3>This website provides valuable information about COVID-19, including symptoms, prevention measures, and up-to-date statistics.</h3></p>

        <p><h3>For more information, please explore the links in the navigation menu.</h3></p>

    </main>

    <br><br>

    <footer>

        &copy; 2023 COVID-19 Information

    </footer>

</body>

</html>