

## Deployment of IndoWordnet Django project on Ubuntu based Apache Server with mod\_wsgi:

- **Prerequisites:**

This section describes pre-installation requirements for IndoWordnet Deployment.

- Python==3.9.2
- pip==20.2.3
- virtualenv==20.0.31
- MySQL Server 8.0==8.0.23
- MySQL 8.0 Command Line Client==8.0.23
- git==2.23.0

- **Installing Apache2 and mod\_wsgi:**

Run the following commands in command terminal.

```
sudo apt-get update
```

```
sudo apt-get install apache2 libapache2-mod-wsgi-py3
```

The above command will install **Apache2** the web server and **mod\_wsgi** for communicating and interfacing with our Django app.

**Note:** If **Apache2** is already installed on the machine, then skip installing **Apache2**. Run the following command instead to install **mod\_wsgi** only.

```
sudo apt-get install libapache2-mod-wsgi-py3
```

- **Cloning the repository:**

Run following command at any desired location(where the project needs to be stored) to clone the github repository to your machine.

```
git clone <github-repository-https-url>
```

This shall create a directory named as “**IndoWordNet**” in the current directory.

**Note:** Ensure that all the files have been cloned properly.

**Important Note:** From here and so on, in all the further instructions, we are assuming that “**IndoWordNet**” directory is located in “**/home/username**” directory. Here, username can be any random name based on the user.

- **Installing Django in Python virtual environment**

Go to the directory “**/home/username/IndoWordNet**” using **cd** command as following:

```
cd /home/username/IndoWordNet
```

Create a virtual environment using following command.

```
virtualenv -p python3 venv
```

Activate the virtual environment using following command.

```
source venv/bin/activate/
```

You shall see the name of your virtual environment enclosed in parentheses prior to command prompt. It indicates that virtual environment is successfully activated.

Now run following command to install Django from **requirements.txt** file(See **Appendix A**).

```
pip install -r requirements.txt
```

To check whether Django is successfully installed or not, run the following command.

```
python -m django --version
```

You shall see the latest version of Django on your machine.

Once, Django is installed successfully on your machine, add directory name “**venv**” in “**.gitignore**” file in the “**IndoWordNet**” directory(See **Appendix A**) to avoid any redundant git commits in future.

- **Hosting the databases on machine:**

Our project consists of three **MySQL** databases(See **Appendix B**). Here are the steps to host a MySQL database to your machine.

1. Open MySQL command line client. Enter the mysql password(if required). You shall see mysql command prompt.
2. Create a database by using following command

*create database <database-name>;*

**Note:** Ensure that database is created successfully by using following command:

*show databases;*

You shall see the recently created database name in the output.

3. Download the .sql files for all three databases and place them inside the /home/username/downloads folder(See **Appendix B**).
4. Open Ubuntu command terminal. Go to the **bin** folder of MySQL Server. And then run the following command to import the data from SQL text file to MySQL database.

*mysql -u <mysql-username> -p <database-name> < <path-to-sql-file>*

For ex.

*mysql -u root -p "iwn" < "/home/username/downloads/iwn.sql"*

**Note:** Ensure that SQL file path(See **Appendix B**) is valid, if any error is generated while running the above command.

5. Now run the following command in MySQL command line client in order to ensure that all the tables in the database are imported successfully.

*show tables;*

You shall see the list of all tables in the database. Hence, database is imported successfully.

Repeat above steps for all three databases, and make sure that all the databases are hosted properly.

- **Connect databases to Django project:**

Here, are the following steps to connect our database to our Django project.

1. Open the **settings.py** file located in **iwn** folder (See **Appendix A**).
2. You shall see the python dictionary named as “**DATABASES**” in the file. It contains the data of databases in key-value pairs where,

key : database variable name (used to address the particular database)

For ex. *'default'* , *'region'* , *'iwn\_utilities'*

**Important Note:** Never change the names of this keys.

value : dictionary containing data of database setting in key-value pairs.

For ex.

```
DATABASES = {
    '<key>' : {
        'ENGINE' : 'django.db.backends.mysql' ,
        'NAME' : <database-name> ,
        'USER' : <mysql-server-username> ,
        'PASSWORD' : <mysql-server-password> ,
        'HOST' : <host-name> ,
        'PORT' : <port-number> ,
    }
}
```

3. Now change the following values in dictionary corresponding to a particular key/database as described.

**NAME:** Set this value to your database name.

**USER:** Set this value to your MySQL server username.

**PASSWORD:** Set this value to your MySQL server password.

**HOST and PORT:** Set this value to the hostname and port number respectively, on which the database is hosted.

Repeat this step for all three keys/databases.

**Important Note:** Make sure that mapping of database to the keys is according to the guidelines described in following table.

SQL text file name (.sql)	Contents of the database	Corresponding Key in DATABASES dictionary
iwn_web_unicode	Hindi Wordnet database, which contains the relation data for all synsets across IndoWordnet	<i>'default'</i>
regional_wn	Synset data in different regional languages	<i>'region'</i>
iwn_utilities_user_feedback	User Feedback data	<i>'iwn_utilities'</i>

Hence, our databases are now successfully connected to our Django project.

- **Gathering Static Content:**

Open “**settings.py**” in “**iwn**” folder(See **Appendix A**). Declare a variable named as “**STATIC\_ROOT**” and set its value as following:

```
STATIC_ROOT = os.path.join(BASE_DIR, "static/")
```

**Note:** Skip this step if **STATIC\_ROOT** variable is already declared as described above.

Now, run the following commands to go to “**iwn**” folder and gather all the static content in our Django project.

```
cd /home/username/IndoWordNet/iwn
```

```
python manage.py collectstatic
```

It will generate output as following:

```
<number-of-static-files-gathered> static files copied to <path-of-static-directory>
```

This shall create a **STATIC** directory in “**/home/username/IndoWordNet**” directory.

- **Migrations:**

Now, run the following commands to go to “**iwn**” folder and migrate our data to database.

```
cd /home/username/IndoWordNet/iwn
```

```
python manage.py makemigrations
```

```
python manage.py migrate
```

- **Running the project on local webserver:**

Now, run the following commands to go to “**iwn**” folder and run our project on local webserver.

```
cd /home/username/IndoWordNet/iwn
```

```
python manage.py runserver
```

If you visit <http://127.0.0.1:8000/> or <http://localhost:8000/>, you shall see the homepage of IndoWordNet website, that means our Django project is running successfully on local webserver.

Now, after confirming that our project is working successfully, quit the server by typing CTRL+C in command terminal.

Also, run the following command to temporarily get out of the virtual environment.

```
deactivate
```

- **Deploying Django project on Apache Server:**

Open the “**iwn-apache-site.conf**” located at “**/home/username/IndoWordNet**” (See **Appendix A**). This file is a virtual host file. And you shall see the script as following:

```
<VirtualHost *:80>
```

```
    ServerAdmin <email-address-of-server-admin>
```

```
    ServerName <hostname>
```

```
    ServerAlias <www.hostname.com>
```

```
    DocumentRoot <path-to-IndoWordNet-parent-dir>
```

```
    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
    Alias /static <path-to-STATIC-directory>
```

```
    <Directory <path-to-STATIC-directory> >
```

```
        Require all granted
```

```
    </Directory>
```

```
    Alias /static <path-to-media-directory>
```

```
    <Directory <path-to-media-directory> >
```

```
        Require all granted
```

```
    </Directory>
```

```
    <Directory <path-to-directory-in-which-wsgi.py-is-located> >
```

```
        <Files wsgi.py>
```

```
            Require all granted
```

```
        </Files>
```

```
    </Directory>
```

```

WSGIDaemonProcess <project-name> python-path=<path-to-IndoWordNet-dir>
python-home=<path-to-virtual-environment-venv-dir>

WSGIProcessGroup <project-name>

WSGIScriptAlias / <path-to-wsgi.py-file>

</VirtualHost>

```

**Note:** Bold and underlined phrases are placeholders, which needs to be replaced with appropriate values. If they are already replaced with valid values then skip doing it again.

Values to be replaced with <email-address-of-server-admin> , <hostname> , <www.hostname.com> can be decided by server admin. Other values which are completely related to Django project needs to be replaced as following:

Placeholder	Value to be replaced
<u>&lt;path-to-IndoWordNet-parent-dir&gt;</u>	/home/username/
<u>&lt;path-to-IndoWordNet-dir&gt;</u>	/home/username/IndoWordNet
<u>&lt;path-to-STATIC-directory&gt;</u>	/home/username/IndoWordNet/static
<u>&lt;path-to-media-directory&gt;</u>	/home/username/IndoWordNet/media
<u>&lt;path-to-directory-in-which-wsgi.py-is-located&gt;</u>	/home/username/IndoWordNet/iwn/iwn
<u>&lt;project-name&gt;</u>	IndoWordNet
<u>&lt;path-to-virtual-environment-venv-dir&gt;</u>	/home/username/IndoWordNet/venv
<u>&lt;path-to-wsgi.py-file&gt;</u>	/home/username/IndoWordNet/iwn/iwn/wsgi.py

Once, it is replaced, save it and close it.

Now, copy this file to “/etc/apache2/sites-available” folder using **cp** command as following:

```
cp /home/username/IndoWordNet/iwn-apache-site.conf /etc/apache2/sites-available
```

Now run following commands to enable this new virtual host file.

```
cd /etc/apache2/sites-available
```



```
sudo a2ensite iwn-apache-site.conf
```

To activate this new configuration, run following command.

```
service apache2 reload
```

Now, we will set up Local Hosts file. This is optional but recommended. To do this just add the following line by replacing placeholder with appropriate value as discussed in previous sections at the end of “**hosts**” file located at “**/etc**”

```
127.0.0.1 <hostname>
```

**Note:** We will access our Django website using this URL “**http://<hostname>/**” in future.

- **Wrapping up some permission issues:**

Run following commands to go to “**IndoWordNet**” directory and give access of all files in directory to apache2

```
cd /home/username/IndoWordNet/
```

```
chown -R www-data:www-data ./
```

Check your apache files to make sure you did not make any syntax errors using following command:

```
sudo apache2ctl configtest
```

You shall see output – “**Syntax OK**”, that means there are no errors.

- **Back to our Django project:**

Run following commands to go to “**IndoWordNet**” directory and activate virtual environment.

```
cd /home/username/IndoWordNet/
```

```
source venv/bin/activate
```

Open “**settings.py**” in “**iwn**” folder(See **Appendix A**). You shall find the python list named as “**ALLOWED\_HOSTS**”. This list contains all the ip addresses, which will have access to our Django application. In our case, it is public. So we will set it to “\*” as following:

```
ALLOWED_HOSTS = ['*']
```

Now, to reflect code changes on the browser, we need to restart the apache service by using following command:

```
sudo service apache2 restart
```

You can access the Django website at the URL – <http://<hostname>/>

- **Error Handling:**

In case of errors, you can view the errors in apache error log. To view errors in terminal, type the following command.

```
tail -f /var/log/apache2/error.log
```

## APPENDIX

### A) File and corresponding relative paths with respect to “IndoWordNet” folder

Filename	Path
settings.py	./iwn/iwn/setings.py
requirements.txt	./requirements.txt
iwn-apache-site.conf	./iwn-apache-site.conf
.gitignore	./.gitignore

### B) Databases Contents and corresponding SQL text file paths

Contents of Databases	Corresponding SQL Text File Path
Hindi Wordnet database, which contains the relation data for all synsets across IndoWordnet	/home/username/downloads/iwn_web_unicode.sql
Synset data in different regional languages	/home/username/downloads/regional_wn.sql
User Feedback data	/home/username/downloads/iwn_utilities_user_feedback.sql

## REFERENCES

- 1) Django Software Foundation, June 2008, accessed on February 2021, <<http://www.djangoproject.com>>
- 2) “How To Setup Django Applications with Apache and mod\_wsgi on Ubuntu”, Huzaif Sayyed, 16 June 2019, accessed on April 2021, <<https://studygyaan.com/django/how-to-setup-django-applications-with-apache-and-mod-wsgi-on-ubuntu>>