**DEPLOYMENT MANUAL**

A cartoon of a person measuring a length

Description automatically generated

**Prerequisites:**

1. Ensure that you have the necessary credentials and access to a server.

2. Install Git on your local machine to clone the repository.

**Step 1: Clone the Repository**

```bash

git clone < >

cd <your\_project\_directory>

```



**Next step-A screenshot of a computer

Description automatically generated**

**Next step-**

A screenshot of a computer

Description automatically generated

**Next step-**

**Copy and paste the link in the open fromVCS and then click clone to clone the code from the github to your desired folder**

A screenshot of a computer

Description automatically generated

**Step 2**: Install Dependencies

```bash

pip install -r requirements.txt

```

A screenshot of a computer

Description automatically generated

**Open terminal and then install all dependencies**

A screenshot of a computer

Description automatically generated

**Step 3:** Database Migration

```bash

python manage.py migrate

```

**Step 4**: Static Files Collection

```bash

python manage.py collectstatic

```

**Step 5:** Configure Django Settings

Update your Django settings for production. Update the `DEBUG` setting to `False` and set the `ALLOWED\_HOSTS` to the domain or IP address of your server.

```python

# settings.py

DEBUG = False

ALLOWED\_HOSTS = ['your\_domain\_or\_ip']

```

**Step 6:** Set Up a Web Server

Choose a web server to serve your Django application. Common choices include Nginx or Apache. Configure the server to serve your Django app using a WSGI server such as Gunicorn.

**Step 7:** Configure Database Connection

Update your database settings in the Django settings file to point to your production database. You might use PostgreSQL, MySQL, or another database supported by Django.

**Step 8:** Set Up ChatterBot

Configure ChatterBot by creating a ChatterBot instance and training it if necessary. Update your Django settings to use the configured ChatterBot instance.

```python

# settings.py

CHATTERBOT = {

'name': 'YourChatterBotInstance',

'trainer': 'chatterbot.trainers.ChatterBotCorpusTrainer',

'training\_data': [

'chatterbot.corpus.english',

# Add additional training data if needed

],

}

```

**Step 9:** Set Up Keras and TensorFlow

Ensure that Keras and TensorFlow are configured correctly. If using GPU, make sure GPU drivers are installed on your server.

**Step 10:** Configure Crispy Forms and Bootstrap

Update your Django settings to use crispy forms with Bootstrap.

```python

# settings.py

CRISPY\_TEMPLATE\_PACK = 'bootstrap5'

```

**Step 11:** Secure Your Application

Implement security measures such as HTTPS, and ensure that your server and application are properly configured.

**Step 12:** Test the Deployment

Run your web server and test your application in a production environment. Check for any errors or issues and resolve them.

**Step 13:** Monitoring and Maintenance

Set up monitoring tools to track the performance of your application. Implement a maintenance plan for regular updates and backups.

**Step- 14: to runserver**

**Use this command line to run server- python manage.py runserver**

A screenshot of a computer

Description automatically generated

Then application will be started in unique sever-id click on it to to redirect from the default browser

A screenshot of a computer

Description automatically generated

Finally application will be running on default browser

A screenshot of a computer

Description automatically generated