Deployment Instructions:

We have deployed our application to the Heroku server for this project. The application can also be deployed on other cloud platforms such as AWS and GCP. The guide below mentions deployment steps on both Heroku and AWS EC2.

Backend:

A. Heroku:

1. Register on the Heroku platform.
2. Install Heroku CLI on the local machine.
3. The Procfile in the project root directory has the instructions to start the Django process on the Heroku dyno.
4. Create an app (*api-backend*) on the Heroku platform.
5. Switch to the main branch.
6. In the project root directory, run the following command to

```
heroku git:remote -a api-backend
```

7. Push the backend repository to the Heroku main branch.

```
git push heroku main
```

8. Open the application using the Heroku app link.

B. EC2:

The instructions are specifically for the Ubuntu 20 operating system.

1. SSH into the machine.
2. Update all the operating system using the below command.

```
apt -y update
```

3. Install all the dependencies required to run the application.

```
apt -y install apt-utils net-tools python3.8 python3-pip mysql-client libmysqlclient-dev openssh-client
```

4. Upgrade all the packages:

```
apt -y upgrade
```

5. We are using **pipenv** to manage python packages.

```
pip3 install pipenv
```

6. Clone the repository.
7. Install all the python packages using the below commands:

```
pipenv install
```

8. Enter the credentials for the database in the highlighted points below

```
export SECRET_KEY=<SECRET_KEY>
export DB_NAME=<DB_NAME>
export DB_USER=<DB_USER>
export DB_HOST=<DB_HOST>
export DB_PASSWORD=<DB_PASSWORD>
export DJANGO_SETTINGS_MODULE=dms.settings.prod
```

9. Run the below commands to set up Django for the production.

```
sudo apt install -y mysql-client-core-8.0
sudo apt install -y libssl-dev libmysqlclient-dev
python3 -m pipenv run python manage.py migrate
python3 -m pipenv run python manage.py collectstatic
sudo apt install -y nginx
sudo systemctl stop nginx
```

10. Edit nginx default configuration to include the following content

```
location / {
        proxy_pass http://0.0.0.0:9000;

}



location /static/ {

        alias /backend/static/;
 }
```

11. Start nginx and gunicorn

```
sudo systemctl start nginx
```

```
python3 -m pipenv run gunicorn greencloud.wsgi:application --bind 0.0.0.0:9000
```

Frontend:

A. Heroku:

1. Register on the Heroku platform.
2. Install Heroku CLI on the local machine.
3. Login using Heroku CLI.
4. Create an app on Heroku using below command:

   ```
   heroku create dms-fe --buildpack mars/create-react-app
   ```

5. Add Heroku remote repository

   ```
   heroku git:remote -a dms-fe
   ```

6. Push the commit to the Heroku repository

   ```
   git push heroku master
   ```

7. Open the application using Heroku app link.

B. EC2:

The instructions are specifically for the Ubuntu 20 operating system.

1. SSH into the machine.
2. Update all the operating system using the below command.

   ```
   apt -y update
   ```

3. Clone the repository.
4. Navigate to the dms-ui directory.
5. Install nodejs

   ```
   curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
   sudo apt install -y nodejs
   ```

6. Install all the dependencies.

```
npm i
```

7. Build the project

```
npm run build
```

8. Install nginx and copy build to the /var/www/html directory

```
sudo apt-get install nginx -y
sudo cp -r build/* /var/www/html
```

References:

[1] "How to Deploy Django App on Heroku Using Gitlab CI/CD Step by Step", *Medium*, 2022. [Online]. Available: https://medium.com/@ganiilhamirsyadi/how-to-deploy-django-app-on-heroku-using-gitlab-ci-cd-step-by-step-45c0d9d31798. [Accessed: 02-Mar-2022]

[2] J.Pottinger, "Deploy React app to Heroku using Gitlab CI/CD Pipeline - test with WebdriverIO", *Juliapottinger.com*, 2022. [Online]. Available: https://juliapottinger.com/deploy-react-app-to-heroku-with-gitlab-ci-test-with-webdriverio/. [Accessed: 12-Mar-2022]