

# Execution Plan for Satellite Overpass Notification and Data Retrieval Platform

## 1. Repository Setup

### 1. Clone the Repository

2. bash
3. Copy code
4. `git clone <repository-url>`
5. `cd <repository-directory>`

### 6. Check Directory Structure

Ensure the repository has the following structure for smooth setup:

7. arduino
8. Copy code
9. |—— project\_folder/
10. | |—— settings.py
11. | |—— urls.py
12. | |—— views.py
13. | |—— tasks.py
14. | |—— models.py
15. |—— templates/
16. |—— static/
17. |—— Execution\_Plan.pdf
18. |—— requirements.txt
19. |—— README.md
20. |—— ...

## 2. Python Environment Setup

### 1. Set up a Python Virtual Environment

2. bash
3. Copy code
4. `python3 -m venv myenv`
5. `source myenv/bin/activate` # For Windows: `myenv\Scripts\activate`

### 6. Install Requirements

Install all dependencies:

7. bash
8. Copy code
9. `pip install -r requirements.txt`

## 3. Configure Celery and Redis for Background Tasks

### 1. Install Redis (if not already installed)

Follow instructions on [Redis's official website](#) to install Redis on your system.

### 2. Start Redis Server

3. bash
4. Copy code
5. `redis-server`

## 6. Configure Celery in Django

Ensure that celery.py is configured correctly with Redis as the broker:

7. python
8. Copy code
9. # In celery.py
10. app.conf.broker\_url = 'redis://localhost:6379/0'
11. app.conf.result\_backend = 'redis://localhost:6379/0'

## 12. Run Celery Worker

Open a new terminal and start the Celery worker:

13. bash
14. Copy code
15. celery -A project\_folder worker -l info

## 4. Configure Email and SMS Services

### 1. Email Setup (SendGrid or SMTP)

Configure email settings in settings.py:

2. python
3. Copy code
4. EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
5. EMAIL\_HOST = 'smtp.sendgrid.net' # or your SMTP provider
6. EMAIL\_PORT = 587
7. EMAIL\_USE\_TLS = True
8. EMAIL\_HOST\_USER = '<sendgrid\_username>'
9. EMAIL\_HOST\_PASSWORD = '<sendgrid\_password>'

### 10. SMS Setup (Twilio)

Add your Twilio account credentials to settings.py:

11. python
12. Copy code
13. TWILIO\_ACCOUNT\_SID = '<your\_account\_sid>'
14. TWILIO\_AUTH\_TOKEN = '<your\_auth\_token>'
15. TWILIO\_PHONE\_NUMBER = '<twilio\_phone\_number>'

## 5. Database Setup

### 1. Set up PostgreSQL Database

Create a PostgreSQL database and user, then add the database configurations in settings.py:

2. python
3. Copy code
4. DATABASES = {
5. 'default': {
6. 'ENGINE': 'django.db.backends.postgresql',
7. 'NAME': 'your\_db\_name',
8. 'USER': 'your\_db\_user',
9. 'PASSWORD': 'your\_db\_password',
10. 'HOST': 'localhost',

11. 'PORT': '5432',
12. }
13. }
14. **Run Migrations**
15. bash
16. Copy code
17. python manage.py makemigrations
18. python manage.py migrate

## 6. Run Django Server

1. **Start Development Server**
2. bash
3. Copy code
4. python manage.py runserver
5. **Access the Application**

Open your browser and go to <http://localhost:8000> to view the application.