1.

```
pip install django
django-admin startproject myproject
cd myproject
python manage.py startapp chatapp
INSTALLED_APPS = [
    ...
    'chatapp',
]
python manage.py migrate
python manage.py runserver
from django.db import models

class Conversation(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    message = models.TextField()
    timestamp = models.DateTimeField(auto_now_add=True)
    summary = models.TextField(blank=True, null=True)  # New field

    def _str_(self):
        return f"Conversation {self.id} with {self.user.username}"
python manage.py makemigrations
python manage.py migrate
pip install openai
import openai

openai.api_key = 'your-openai-api-key'
```

```
def generate_summary(conversation_text):
    response = openai.Completion.create(
        engine="text-davinci-004",
        prompt=f"Summarize the following conversation: {conversation_text}",
        max_tokens=150
    )
    summary = response.choices[0].text.strip()
    return summary

def save_conversation(request):
    if request.method == 'POST':
        message = request.POST['message']
        conversation = Conversation(user=request.user, message=message)
        conversation.summary = generate_summary(message)
        conversation.save()
        return redirect('conversation_detail', conversation_id=conversation.id)
from django.contrib import admin
from .models import Conversation

class ConversationAdmin(admin.ModelAdmin):
    list_display = ('user', 'message', 'timestamp', 'summary')

admin.site.register(Conversation, ConversationAdmin)
```

2.

```
pip install psycopg2-binary
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'mydatabase',
        'USER': 'mydatabaseuser',
        'PASSWORD': 'mypassword',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
sudo -u postgres psql
CREATE DATABASE mydatabase;
CREATE USER mydatabaseuser WITH PASSWORD 'mypassword';
ALTER ROLE mydatabaseuser SET client_encoding TO 'utf8';
ALTER ROLE mydatabaseuser SET default_transaction_isolation TO 'read committed';
ALTER ROLE mydatabaseuser SET timezone TO 'UTC';
GRANT ALL PRIVILEGES ON DATABASE mydatabase TO mydatabaseuser;
\q
python manage.py dumpdata > db.json
python manage.py migrate
python manage.py loaddata db.json
import datetime
from django.core.management.base import BaseCommand
from chatapp.models import Conversation

class Command(BaseCommand):
    help = 'Delete conversations older than a specified number of days'
```

```
    def add_arguments(self, parser):
        parser.add_argument('days', type=int, help='Number of days to retain conversations')

    def handle(self, *args, **kwargs):
        days = kwargs['days']
        cutoff_date = datetime.datetime.now() - datetime.timedelta(days=days)
        old_conversations = Conversation.objects.filter(timestamp__lt=cutoff_date)
        count = old_conversations.count()
        old_conversations.delete()
        self.stdout.write(f'Deleted {count} conversations older than {days} days')
pip install django-crontab
INSTALLED_APPS = [
    ...
    'django_crontab',
]
CRONJOBS = [
    ('0 0 * * 0', 'django.core.management.call_command', ['cleanup_conversations', '30'])
]
python manage.py crontab add
```