📘 Supabase Integration with Django

Complete Guide: Account Creation → Integration → File Upload (Models & Views)

---

1. Introduction

In this project, we integrate **Supabase** with a **Django application** to handle: - User data storage - File (profile picture) uploads - Public URL generation for uploaded files

Technology roles: - **Django** → Backend logic, views, models - **Supabase** → Database + File Storage

---

2. Creating a Supabase Account

1. Go to **https://supabase.com**
2. Sign up using **GitHub / Email**
3. Click **New Project**
4. Enter:
   - Project name
   - Database password
   - Region
5. Click **Create Project**

---

3. Getting Supabase Credentials

Navigate to:

```
Settings → API
```

Copy and store securely: - **Project URL** - **Anon Public Key** - **Service Role Key** (server-side only)

---

4. Installing Supabase in Django

```
pip install supabase
```

---

5. Environment Variables Setup

Create a `.env` file:

```
SUPABASE_URL=https://xxxx.supabase.co
SUPABASE_SERVICE_KEY=your_service_role_key
```

Add `.env` to `.gitignore`.

---

6. Supabase Client Configuration

Create file:

```
app_name/supabase_client.py
```

```python
from supabase import create_client
import os

SUPABASE_URL = os.getenv("SUPABASE_URL")
SUPABASE_KEY = os.getenv("SUPABASE_SERVICE_KEY")

supabase = create_client(SUPABASE_URL, SUPABASE_KEY)
```

---

7. Creating Supabase Storage Bucket

1. Go to **Storage** in Supabase Dashboard
2. Create a bucket named:

```
profile_pics
```

3. Set bucket to **Public**

---

8. Creating Database Table in Supabase

Run in SQL Editor:

```sql
create table user_profiles (
  id bigint generated by default as identity primary key,
  name text,
  email text,
  age int,
  profile_pic_url text,
  created_at timestamp default now()
);
```

---

9. Django Model (`models.py`)

```
from django.db import models

class UserProfile(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    age = models.PositiveIntegerField()
    profile_pic_url = models.URLField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name
```

Run migrations:

```
python manage.py makemigrations
python manage.py migrate
```

---

10. File Upload Naming Logic

Original File Path

```
c/doc/new/sample.jpg
```

Generated Unique File

```
14521.jpg
```

Final Storage Path

```
users/14521.jpg
```

---

11. File Upload Flow (Step-by-Step)

Step 1: Get File from Form-Data

```
profile_pic = request.FILES.get("profile_pic")
```

Step 2: Extract File Extension

```
extension = profile_pic.name.split('.')[-1]
```

Step 3: Generate Unique Name

```
import uuid
unique_name = str(uuid.uuid4())
```

Step 4: Merge Name and Extension

```
file_path = f"users/{unique_name}.{extension}"
```

Step 5: Upload File to Supabase Storage

```
supabase.storage.from_("profile_pics").upload(
    file_path,
    profile_pic.read(),
    {"content-type": profile_pic.content_type}
)
```

Step 6: Generate Public URL

```
public_url = supabase.storage.from_("profile_pics").get_public_url(file_path)
```

Step 7: Store URL in Django Database

```
UserProfile.objects.create(
    name=name,
    email=email,
    age=age,
    profile_pic_url=public_url
)
```

---

12. Django View (`views.py`)

```
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from .models import UserProfile
from .supabase_client import supabase
import uuid

@csrf_exempt
def create_user_profile(request):
    if request.method != "POST":
        return JsonResponse({"error": "POST required"}, status=405)
    name = request.POST.get("name")
    email = request.POST.get("email")
    age = request.POST.get("age")
    profile_pic = request.FILES.get("profile_pic")

    ext = profile_pic.name.split('.')[-1]
    file_name = f"{uuid.uuid4()}.{ext}"
    file_path = f"users/{file_name}"

    supabase.storage.from_("profile_pics").upload(
        file_path,
        profile_pic.read(),
        {"content-type": profile_pic.content_type}
    )

    public_url =
supabase.storage.from_("profile_pics").get_public_url(file_path)

    user = UserProfile.objects.create(
        name=name,
        email=email,
        age=int(age),
        profile_pic_url=public_url
    )

    return JsonResponse({
        "status": "success",
        "user_id": user.id,
        "profile_pic_url": public_url
    })
```

13. URLs Configuration

```
from django.urls import path
from .views import create_user_profile

urlpatterns = [
    path("create-profile/", create_user_profile),
]
```

14. Testing Using Postman

- Method: **POST**
- Body Type: **form-data**

Key	Type

Key	Type
name	Text
email	Text
age	Text
profile_pic	File

---

15. Important Points for Students & Interviews

- Never store images directly in database
- Use cloud storage for files
- Store only file URLs in DB
- Use `form-data` for text + files
- Generate unique filenames to avoid conflicts

✅ Final Summary

Django handles logic, Supabase handles storage, and the database stores only file URLs.