**✅ Goal:**

You will deploy your **React + Django project together** as **one Azure App Service** using **VS Code only**, with no GitHub, no Docker, and no separate deployments.

**⚙️ Prerequisites (Install Only Once)**

1. Install [**Azure CLI**](https://learn.microsoft.com/en-us/cli/azure/install-azure-cli)
2. Install **VS Code**
3. Install VS Code extensions:
   * ✅ **Azure App Service**
   * ✅ **Python**
4. Create an Azure account: <https://portal.azure.com>
5. **Login to Azure in VS Code**:
   * Open VS Code → Press Ctrl+Shift+P → Type Azure: Sign In → Login via browser

**🏗️ Step-by-Step Deployment**

**🔹 STEP 1: Set Up Django**

1. Open your **Django project** folder in VS Code (must contain manage.py).
2. Install these:

bash

CopyEdit

pip install gunicorn whitenoise

pip freeze > requirements.txt

1. In settings.py:
   * Add to MIDDLEWARE at the top:

python

CopyEdit

'whitenoise.middleware.WhiteNoiseMiddleware',

* + At the bottom:

python

CopyEdit

import os

STATIC\_ROOT = os.path.join(BASE\_DIR, 'staticfiles')

ALLOWED\_HOSTS = ['\*'] # You can update this after deployment

1. Run:

bash

CopyEdit

python manage.py collectstatic

1. Create a file named startup.txt in root of your project with:

bash

CopyEdit

gunicorn your\_project\_name.wsgi --bind=0.0.0.0 --timeout 600

Replace your\_project\_name with your actual Django project folder name.

**🔹 STEP 2: Build React and Add to Django**

1. Go to your React folder:

bash

CopyEdit

cd react-frontend

npm install

npm run build

1. After build, go back to Django project and do this:
   * Copy the contents of react-frontend/build to a folder like:

arduino

CopyEdit

your\_django\_project/static/react/

1. In urls.py, serve the React app:

python

CopyEdit

from django.views.generic import TemplateView

urlpatterns = [

path('', TemplateView.as\_view(template\_name="index.html")),

]

1. Copy index.html from React build to Django templates folder.

**🔹 STEP 3: Deploy to Azure from VS Code**

1. In VS Code → Click the **Azure icon** on the sidebar.
2. Click **"App Service"** → Click the **"+" icon** to create a new web app:
   * Runtime: **Python 3.10**
   * OS: **Linux**
   * Name: e.g., ticketingtool2025
   * Region: Choose nearest
3. After it's created, it appears in the App Service list.
4. Right-click on your created App → Select **"Deploy to Web App"**
   * Select your Django folder (the one with manage.py and startup.txt)
   * VS Code will zip and upload everything

**🔹 STEP 4: Final Azure Portal Configuration**

1. Go to [Azure Portal](https://portal.azure.com)
2. Open your App Service → **Configuration**
3. Under “Application settings”, add:
   * DJANGO\_SECRET\_KEY with your actual key
   * DEBUG = False
4. Go to **General Settings**:
   * Startup command:

bash

CopyEdit

gunicorn your\_project\_name.wsgi --bind=0.0.0.0 --timeout 600

**🎉 You’re Done**

Open your live app:

cpp

CopyEdit

https://<your-app-name>.azurewebsites.net

React + Django is now fully deployed and served from **one Azure App Service** via **VS Code**, no Git, no confusion.

**✅ You said:**

Your main folder is:

sql

CopyEdit

Ticketing\_tool/

│

├── Ticketing\_tool/ ← (Contains settings.py, urls.py, wsgi.py)

│

├── manage.py

You also have a **React app**, likely in a folder like:

CopyEdit

react-frontend/

**✅ Step-by-Step: Move React Build to Django Correctly**

**🔹 Step 1: Build React App**

From terminal:

bash

CopyEdit

cd react-frontend

npm install

npm run build

This will generate a folder:

bash

CopyEdit

react-frontend/build/

Inside build/ you’ll see:

cpp

CopyEdit

index.html

static/

favicon.ico

etc.

**🔹 Step 2: Move React Build Into Django**

Now, follow this exact path:

1. **Go to your Django project folder:**

CopyEdit

Ticketing\_tool/

1. **Create these folders** if they don’t exist:
   * static/
   * templates/

You’ll now have:

cpp

CopyEdit

Ticketing\_tool/

├── static/

└── templates/

1. **Copy files from React build folder like this:**

| **From** | **To** |
| --- | --- |
| react-frontend/build/static/ | Ticketing\_tool/static/ |
| react-frontend/build/index.html | Ticketing\_tool/templates/index.html |

📌 DO NOT copy the entire build/ folder — just the static/ contents and index.html.

**🔹 Step 3: Tell Django to Use Them**

**In settings.py (inside inner Ticketing\_tool/):**

Add:

python

CopyEdit

import os

# For templates

TEMPLATES[0]['DIRS'] = [os.path.join(BASE\_DIR, 'templates')]

# For static

STATICFILES\_DIRS = [os.path.join(BASE\_DIR, 'static')]

STATIC\_ROOT = os.path.join(BASE\_DIR, 'staticfiles')

**🔹 Step 4: Set Route to Serve React App**

In urls.py (inside inner Ticketing\_tool/):

python

CopyEdit

from django.urls import path

from django.views.generic import TemplateView

urlpatterns = [

path('', TemplateView.as\_view(template\_name="index.html")),

]

**✅ Final Folder Structure Now Looks Like:**

csharp

CopyEdit

Ticketing\_tool/

├── manage.py

├── static/ ← contains React static files

├── staticfiles/ ← created by collectstatic

├── templates/

│ └── index.html ← from React build

├── Ticketing\_tool/

│ ├── \_\_init\_\_.py

│ ├── settings.py

│ ├── urls.py

│ ├── wsgi.py

After this, do:

bash

CopyEdit

python manage.py collectstatic

Then you’re ready to deploy to Azure from VS Code as explained in the earlier steps.