**Can you refactor the JavaScript so that when I have already clicked**

**farm it does not give sound from farm.wav**

**cave it does not give sound from cave.wav**

**house it does not give sound from house.wav**

**Until reset**

**can you implement sound in my app using Javascript. The type of sound that I desire is as follows:**

**from django.http import JsonResponse**

**def get\_gold\_value(request):**

**gold = request.session.get('gold', 0)**

**return JsonResponse({'gold': gold})**

Neon

Railway

**Supabase**

**How to Connect a Free Cloud PostgreSQL Database to Django**

Sign up for one of the platforms above

Create a PostgreSQL database

**Get the database connection URL (usually looks like this):**

postgres://username:password@host:port/database\_name

**Run migrations**

python manage.py migrate

**supabase-amber-house**

Organization: RevHam1

Project name: Ninja gold

Database Password: 52454Rev@7TyLo

Copy the **PostgreSQL connection string** (starts with postgres://

**Direct connection**: postgresql://postgres:[YOUR-PASSWORD]@db.snbtbdkexdengvubityq.supabase.co:5432/postgres

**Direct connection**: postgresql://postgres: **52454Rev@7TyLo**@db.snbtbdkexdengvubityq.supabase.co:5432/**postgres**

**Transaction pooler**: postgresql://postgres.snbtbdkexdengvubityq:[YOUR-PASSWORD]@aws-0-us-east-1.pooler.supabase.com:6543/postgres

**Session pooler**: postgresql://postgres.snbtbdkexdengvubityq:[YOUR-PASSWORD]@aws-0-us-east-1.pooler.supabase.com:5432/postgres

**How to properly set Environment Variables with supabase and vercel.**

Vercel

Heroku

PythonAnywhere

Render

**Railway**

AWS Free Tier

OpenShift

Glitch

Qovery

What is 500 hours of run time in railway

What is $5 of resource usage per month on railway

Which works best with github railway or render

Which is easier to deploy a django app render or railway

**ep-sparkling-hill-a49kty24.us-east-1.pg.koyeb.app**

Host: (ep-xxxxxx.us-east-1.pg.koyeb.app)

Database Name: koyebdb

Username: koyeb-adm

Password: npg\_lLoIRy9Y7qVQ

Port (5432 by default)

pg\_dump -h ep-xxxxxx.us-east-1.pg.koyeb.app -U your\_username -d your\_database -F p -f koyeb\_backup.sql

pg\_dump -h **ep-sparkling-hill-a49kty24.us-east-1.pg.koyeb.app** -U koyeb-adm -d **koyeb-adm** -F p -f koyeb\_backup.sql

**Project URL**: https://snbtbdkexdengvubityq.supabase.co

**API Key**: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InNuYnRiZGtleGRlbmd2dWJpdHlxIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI1OTcyNDQsImV4cCI6MjA1ODE3MzI0NH0.ERrG5i4cBTIdKLIVsyCmL9z6kA15eGYVv6mLKaHzEJA

**eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTk0MjIsImV4cCI6MjA1ODE5NTQyMn0.H46LYVE9t3JfuqgzP2USIkdQlZLgHsTq6\_Xvn7acUJ**Y

I am trying to switch form Koyeb data base to **supabase** Using vercel Market place do I need swich the following to use supabase information

**POSTGRES\_USER="postgres"**

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTk0MjIsImV4cCI6MjA1ODE5NTQyMn0.H46LYVE9t3JfuqgzP2USIkdQlZLgHsTq6\_Xvn7acUJY"

**POSTGRES\_PASSWORD="YMC85vrZrYyG03gs"**

**POSTGRES\_DATABASE="postgres"**

SUPABASE\_SERVICE\_ROLE\_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc0MjYxOTQyMiwiZXhwIjoyMDU4MTk1NDIyfQ.eL3ic1SJHP70clSZEgjDn\_DVNHHNrDGfoR6dxhKUwbs"

**POSTGRES\_HOST="db.ovxhorlyrpsxlbpokqfy.supabase.co"**

**These supbase environment variable are also included with keyob environment variables in vercel.com do I need remove keyobs variables?**

1. **POSTGRES\_UR**L="***postgres://postgres.ovxhorlyrpsxlbpokqfy:YMC85vrZrYyG03gs@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa=base-pooler.x***"
2. **POSTGRES\_PRISMA\_URL**="postgres://postgres.ovxhorlyrpsxlbpokqfy:YMC85vrZrYyG03gs@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa=base-pooler.x"
3. **SUPABASE\_URL**="https://ovxhorlyrpsxlbpokqfy.supabase.co"
4. **NEXT\_PUBLIC\_SUPABASE\_UR**L="https://ovxhorlyrpsxlbpokqfy.supabase.co"
5. **POSTGRES\_URL\_NON\_POOLING**="postgres://postgres.ovxhorlyrpsxlbpokqfy:YMC85vrZrYyG03gs@aws-0-us-east-1.pooler.supabase.com:5432/postgres?sslmode=require"
6. **SUPABASE\_JWT\_SECRET**="V5OLohyyGWCAv7rSVwL4YeJyeyoPoIPVmjaD22l7+BUD6zqS65o203dicGr1jgQqtjKnZi2rqh7E0XFusaKj+w=="
7. **POSTGRES\_USER**="***postgres***"
8. **NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY**="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTk0MjIsImV4cCI6MjA1ODE5NTQyMn0.H46LYVE9t3JfuqgzP2USIkdQlZLgHsTq6\_Xvn7acUJY"
9. **POSTGRES\_PASSWORD**="***YMC85vrZrYyG03gs***"
10. **POSTGRES\_DATABASE**="***postgres***"
11. **SUPABASE\_SERVICE\_ROLE\_KEY**="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc0MjYxOTQyMiwiZXhwIjoyMDU4MTk1NDIyfQ.eL3ic1SJHP70clSZEgjDn\_DVNHHNrDGfoR6dxhKUwbs"
12. **POSTGRES\_HOST**="***db.ovxhorlyrpsxlbpokqfy.supabase.co***"
13. **NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY**="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTk0MjIsImV4cCI6MjA1ODE5NTQyMn0.H46LYVE9t3JfuqgzP2USIkdQlZLgHsTq6\_Xvn7acUJY"

**NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY**="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTk0MjIsImV4cCI6MjA1ODE5NTQyMn0.H46LYVE9t3JfuqgzP2USIkdQlZLgHsTq6\_Xvn7acUJY"

**connection to server at "ep-sparkling-hill-a49kty24.us-east-1.pg.koyeb.app" (34.194.100.28)**

https://ovxhorlyrpsxlbpokqfy.supabase.co

Where can I find connection server for SUPABASE in vercel.com

<https://swiycxvcswhqdnkyrfcs.supabase.co>

**API**

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InN3aXljeHZjc3docWRua3lyZmNzIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTM4MzIsImV4cCI6MjA1ODE4OTgzMn0.rSJzYvuxWj6vza4BMA5bKVyTJjwhAH7veyKejgtnRvg

I have

**Build a requirements.txt-compatible wheel locally:**

pip download psycopg2-binary --no-binary=:none: --dest ./precompiled\_packages

**How do I Add the ./precompiled\_packages folder to my codebase, and reference it in my vercel.json:**

**I have**

C:\Users\Tyrone\Desktop\Django Dojo\A Projects\4 ninja\_gold\staticfiles

https://vercel.com/tyrones-projects-e4281f99/ninja-gold-n1na/stores/integration/store\_mUqbDNSZXLnMwgGl/guides

https://vercel.com/tyrones-projects-e4281f99/ninja-gold-n1na/settings/environment-variables

How to check if supabase is running in vercel

How to check if TCP/IP connections in vercel

**How to find Supabase database URL in vercel.com**

https://ovxhorlyrpsxlbpokqfy.supabase.co

https://<your-supabase-url>.supabase.co/rest/v1/<your-table-name>" \

-H "apikey: <your-supabase-key>

What is <your-table-name>" in

https://db.**ovxhorlyrpsxlbpokqfy.supabase.co**.supabase.co/rest/v1/ " \

-H "apikey: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InNuYnRiZGtleGRlbmd2dWJpdHlxIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI1OTcyNDQsImV4cCI6MjA1ODE3MzI0NH0.ERrG5i4cBTIdKLIVsyCmL9z6kA15eGYVv6mLKaHzEJA

https://**ovxhorlyrpsxlbpokqfy**.supabase.co/rest/v1/<your-table-name>" \

-H "apikey: <your-supabase-key>

**API Key**:

apikey

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InNuYnRiZGtleGRlbmd2dWJpdHlxIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI1OTcyNDQsImV4cCI6MjA1ODE3MzI0NH0.ERrG5i4cBTIdKLIVsyCmL9z6kA15eGYVv6mLKaHzEJA

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Im92eGhvcmx5cnBzeGxicG9rcWZ5Iiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI2MTk0MjIsImV4cCI6MjA1ODE5NTQyMn0.H46LYVE9t3JfuqgzP2USIkdQlZLgHsTq6\_Xvn7acUJY

**https://ninja\_gold.vercel.app/test-supabase/**

**This is my supabase\_test function in views,py**

**The issue with your deployed app showing nothing on the page**

I see nothing when on the page

 Checked for errors in my browser console and Vercel logs. The following are the messages

Used Buildings: (3) ['farm', 'cave', 'house']

(index):168 Game Won Condition: true

(index):169 Game Lost Condition: false

(index):179 Playing applause sound for win

application was successfully cloned, built, and deployed on Vercel.

 Tested my app locally and ensure it renders correctly.

Yes, my app in development renders correct

 Confirmed that all static files, routes, and configurations are working as expected.

Yes, my in development static files, routes, and configurations are working as expected

When I click on custom domains **ninja-gold-n1na.vercel.app** I am routed **to https://vercel.com/tyrones-projects-e4281f99/ninja-gold-n1na/EDReVpkKU9vaA8CHvyRotE74QMPp**

**When I navigate directly to https://ninja-gold-ttuj.vercel.app/**

**I get 404**

**1. Verify Your Entry Point**

**My views index fuunction**

def index(request):

if 'gold' not in request.session:

request.session['gold'] = 0

print(f"Current gold in session: {request.session.get('gold')}")

if 'activites' not in request.session:

request.session['activites'] = []

if 'used\_buildings' not in request.session:

request.session['used\_buildings'] = []

return render(request, 'index.html')

**My app url.py**

from django.urls import path

from . import views

urlpatterns = [

path('', views.index),

path('process\_money', views.process\_money),

path('reset', views.reset),

path('ninja\_gold/', views.ninja\_gold\_game, name='ninja\_gold\_game'),

path('reset/', views.reset, name='reset'),

path('get\_gold\_value/', views.get\_gold\_value, name='get\_gold\_value'),

]

**Ny project url.py**

from django.urls import path, include

urlpatterns = [

path('', include('ninja\_gold\_app.urls'))

# path('admin/', admin.site.urls),

]

2. My vercel.json Configuration

{

"version": 2,

"routes": [

{

"src": "/(.\*)",

"dest": "ninja\_gold/wsgi.py"

}

],

"installCommand": "pip install --no-index --find-links=./precompiled\_packages -r requirements.txt"

}

**3. Verify Deployment Settings**

* Go to your Vercel dashboard and check the **Build & Development Settings**.
* Ensure the correct framework is selected (e.g., Next.js, React, or Other).
* Confirm that the **Root Directory** is set correctly. If your app is in a subdirectory (e.g., /ninja\_gold), update the root directory in the settings.

{

  "version": 2,

  "routes": [

    {

      "src": "/(.\*)",

      "dest": "ninja\_gold/wsgi.py"

    }

  ],

  "installCommand": "pip install --no-index --find-links=./precompiled\_packages -r requirements.txt"

}

{

  "version": 2,

  "builds": [

    {

      "src": "ninja\_gold/wsgi.py",

      "use": "@vercel/python",

      "config": {

        "runtime": "python@3.10"

      }

    }

  ],

  "routes": [

    {

      "src": "/(.\*)",

      "dest": "ninja\_gold/wsgi.py"

    }

  ],

  "installCommand": "pip install --upgrade setuptools pip wheel && pip install --disable-pip-version-check --upgrade -r requirements.txt"

}

**vercel.com/ninja-gold-n1na.vercel.app**

How many apps on the free plan can you have with vercel.com and supabase

Would it be worth removing the app from vercel.com and starting over also implementing supabase? Do I first delete and recreate the app in vercel,com then utilize supabase. Even though I tried to use superbase with the app I am going to delete and recreate

cd \Users\Tyrone\Desktop\Django Dojo\A Projects\4 ninja\_gold\ninja\_gold

cd /Users/Tyrone/Desktop/Django Dojo/A Projects/4 ninja\_gold/ninja\_gold

give me step by step instructions to remove an app from vercel.com and starting over also implementing supabase as a backend eventhough I really don't need tables. I need it to run y django python app, migrations and make migrations. Do I first delete and recreate the app in vercel,com then utilize supabase. Even though I tried to use superbase with the app I am going to delete and recreate

python manage.py runserver

deploying a python django app on vercel with supabase

supabase

The 500 Error: Trying to get my Django-Python app to display live with vercel hosted with supabase. I tried multiple approaches. I tried so many things I cannot remember. But one of the last things I did was to have my app display a simple line of JSON that says HELLO. It works in development but deployment, I get the sane 500 Error. Even reducing the requirements.txt to a minimal produces the same thing. I am stuck. How do I handle this error in my case.

This Serverless Function has crashed.

Your connection is working correctly.

Vercel is working correctly.

500: INTERNAL\_SERVER\_ERROR

Code: FUNCTION\_INVOCATION\_FAILED

ID: cle1::6tmk6-1743047530274-c1dd48758875

These are some of the many things I have done.

**Check Build Logs**

seems like you've successfully deployed your project on Vercel! From the logs, it shows that the build and deployment processes were completed without any errors.

**Verify Environment Variables**

Checked Environment variable in the app and in vercel

**Check vercel.json Configuration**

builds and routes sections are set up to serve my app correctly.

**Logs and Errors**

500 Internal Server: this is the problem

**Static Files and Media**

Ran collect sta*tic*

**Domain Configuration**

Not using custom domin

**Test Locally**

locally to ensure it works as expected

**Simplify Function**

To show simple JSON

**Simplify requirements**

To only a few bare minimum lines

https://github.com/RevHam1/NinjaGold

**https://vercel.community/t/the-dreaded-500-error-again/7673/2**

**These are Supabase Evironment variables https://vercel.com/tyrones-projects-e4281f99/ninja-gold/stores/integration/store\_bTzr51IvAgVX7FBK/guides**

POSTGRES\_URL="postgres://postgres.cqqhkbnuqbqsvifjhqhs:l5Uf9L4A7Cx3m7cc@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa=base-pooler.x"

POSTGRES\_PRISMA\_URL="postgres://postgres.cqqhkbnuqbqsvifjhqhs:l5Uf9L4A7Cx3m7cc@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa=base-pooler.x"

SUPABASE\_URL="https://cqqhkbnuqbqsvifjhqhs.supabase.co"

NEXT\_PUBLIC\_SUPABASE\_URL="https://cqqhkbnuqbqsvifjhqhs.supabase.co"

POSTGRES\_URL\_NON\_POOLING="postgres://postgres.cqqhkbnuqbqsvifjhqhs:l5Uf9L4A7Cx3m7cc@aws-0-us-east-1.pooler.supabase.com:5432/postgres?sslmode=require"

SUPABASE\_JWT\_SECRET="cg8/G8E+dnGTHWk60JaGUjsEkfBHCorX1d0pOAgePNwKgewo+dg8xLjYSLC0o++u3SOMbkg8Fu79fMYqnJ2Rmw=="

POSTGRES\_USER="postgres"

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImNxcWhrYm51cWJxc3ZpZmpocWhzIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI4OTIwMzMsImV4cCI6MjA1ODQ2ODAzM30.49qez5yJPPBmjPkp6yPQljM-csUSHvIUkbAoClDXmWQ"

POSTGRES\_PASSWORD="***l5Uf9L4A7Cx3m7cc***"

POSTGRES\_DATABASE="postgres"

SUPABASE\_SERVICE\_ROLE\_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImNxcWhrYm51cWJxc3ZpZmpocWhzIiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc0Mjg5MjAzMywiZXhwIjoyMDU4NDY4MDMzfQ.M9beoTqWLxSgFtxp2J2skeU12AoC1xfMWa9a4ym805s"

POSTGRES\_HOST="db.cqqhkbnuqbqsvifjhqhs.supabase.co"

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImNxcWhrYm51cWJxc3ZpZmpocWhzIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDI4OTIwMzMsImV4cCI6MjA1ODQ2ODAzM30.49qez5yJPPBmjPkp6yPQljM-csUSHvIUkbAoClDXmWQ"

**https://vercel.com/tyrones-projects-e4281f99/ninja-gold/settings/environment-variables**

Gold\_POSTGRES\_URL\_NON\_POOLING •••••••••••••••

Gold\_SUPABASE\_JWT\_SECRET •••••••••••••••

Gold\_POSTGRES\_USER •••••••••••••••

Gold\_NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY •••••••••••••••

Gold\_POSTGRES\_PASSWORD •••••••••••••••

Gold\_POSTGRES\_DATABASE •••••••••••••••

Gold\_SUPABASE\_SERVICE\_ROLE\_KEY •••••••••••••••

Gold\_POSTGRES\_HOST •••••••••••••••

Gold\_SUPABASE\_ANON\_KEY •••••••••••••••

PORT •••••••••••••••

PORT •••••••••••••••

POSTGRES\_URL •••••••••••••••

POSTGRES\_PRISMA\_URL •••••••••••••••

SUPABASE\_URL •••••••••••••••

NEXT\_PUBLIC\_SUPABASE\_URL •••••••••••••••

POSTGRES\_URL\_NON\_POOLING •••••••••••••••

SUPABASE\_JWT\_SECRET •••••••••••••••

POSTGRES\_USER •••••••••••••••

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY •••••••••••••••

POSTGRES\_PASSWORD •••••••••••••••

POSTGRES\_DATABASE •••••••••••••••

SUPABASE\_SERVICE\_ROLE\_KEY •••••••••••••••

POSTGRES\_HOST •••••••••••••••

SUPABASE\_ANON\_KEY •••••••••••••••

What’s the difference between the environment variables found at guides

https://vercel.com/tyrones-projects-e4281f99/ninja-gold/stores/integration/store\_bTzr51IvAgVX7FBK/**guides**

And the environment variables found at environment-variables

https://vercel.com/tyrones-projects-e4281f99/ninja-gold/settings/**environment-variables**

So when you talk about Verifing Environment Variables are you talking about verifying these in "Settings" https://vercel.com/tyrones-projects-e4281f99/ninja-gold/**settings**/environment-variables

If so with what? Or these in "Guides"

https://vercel.com/tyrones-projects-e4281f99/ninja-gold/stores/integration/store\_bTzr51IvAgVX7FBK/**guides**

These variables are categorized by environment (e.g., ***Production***, ***Preview***, ***Development***)

***POSTGRES\_URL*** correctly formatted

Supabase: Ensure all required environment variables are set in your Vercel project. These typically include:

**When verifying these in my settings>environment variables which are needed.**

https://vercel.com/tyrones-projects-e4281f99/ninja-gold/settings/environment-variables

Gold\_POSTGRES\_URL\_NON\_POOLING •••••••••••••••

Gold\_SUPABASE\_JWT\_SECRET •••••••••••••••

Gold\_POSTGRES\_USER •••••••••••••••

Gold\_NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY •••••••••••••••

Gold\_POSTGRES\_PASSWORD •••••••••••••••

Gold\_POSTGRES\_DATABASE •••••••••••••••

Gold\_SUPABASE\_SERVICE\_ROLE\_KEY •••••••••••••••

Gold\_POSTGRES\_HOST •••••••••••••••

Gold\_SUPABASE\_ANON\_KEY •••••••••••••••

PORT •••••••••••••••

PORT •••••••••••••••

POSTGRES\_URL •••••••••••••••

~~POSTGRES\_PRISMA\_URL •••••••••••••••~~

SUPABASE\_URL •••••••••••••••

NEXT\_PUBLIC\_SUPABASE\_URL •••••••••••••••

POSTGRES\_URL\_NON\_POOLING •••••••••••••••

SUPABASE\_JWT\_SECRET •••••••••••••••

POSTGRES\_USER •••••••••••••••

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY •••••••••••••••

POSTGRES\_PASSWORD •••••••••••••••

POSTGRES\_DATABASE •••••••••••••••

SUPABASE\_SERVICE\_ROLE\_KEY •••••••••••••••

POSTGRES\_HOST •••••••••••••••

SUPABASE\_ANON\_KEY •••••••••••••••

**POSTGRES\_URL**

**POSTGRES\_USER**

**POSTGRES\_PASSWORD**

**POSTGRES\_DATABASE**

**POSTGRES\_HOST**

***PORT***: Typically required for deployment environments to specify the port where your app runs

**SUPABASE\_URL / NEXT\_PUBLIC\_SUPABASE\_URL**: for interacting with the Supabase API (used for frontend services or REST API calls).

**SUPABASE\_ANON\_KEY / NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY**: access to Supabase services (e.g., client-side API calls).

**SUPABASE\_JWT\_SECRET**: Important for authentication or secure API access.

**SUPABASE\_SERVICE\_ROLE\_KEY**: Used for server-side operations that need elevated permissions.

Since I am not using an API or authentication. It wold be safe tom assdume the Environment variables I need are

POSTGRES\_URL: postgres://postgres:**l5Uf9L4A7Cx3m7cc**@**db.cqqhkbnuqbqsvifjhqhs.supabase.co**:**5432**/**postgres**

---------------------------------------------------------------------------------------------------------------------

postgres://postgres.cqqhkbnuqbqsvifjhqhs:l5Uf9L4A7Cx3m7cc@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa=base-pooler.x

---------------------------------------------------------------------------------------------------------------------

POSTGRES\_USER: ***postgres***

POSTGRES\_PASSWORD: ***l5Uf9L4A7Cx3m7cc***

POSTGRES\_DATABASE: ***postgres***

POSTGRES\_HOST: **db.cqqhkbnuqbqsvifjhqhs.supabase.co.**

PORT: ***5432***

postgres://postgres:l5Uf9L4A7Cx3m7cc@db.cqqhkbnuqbqsvifjhqhs.supabase.co:5432/postgres

**Do I need a .env file**

**Which POSTGRES\_URL should I use the one I set**

POSTGRES\_URL: postgres://postgres:**l5Uf9L4A7Cx3m7cc**@**db.cqqhkbnuqbqsvifjhqhs.supabase.co**:**5432**/**postgres**

**Or the one integrate into my setings > environment varibles**

postgres://postgres.cqqhkbnuqbqsvifjhqhs:l5Uf9L4A7Cx3m7cc@aws-0-us-east-1.pooler.supabase.com:6543/postgres?sslmode=require&supa=base-pooler.x

postgres://username:password@host:port/database\_name

POSTGRES\_URL: postgres://username:password@host:port/database\_name

username, password, host, port, and database name

POSTGRES\_USER: (likely **postgres** for Supabase)

POSTGRES\_PASSWORD: password for your database connection

POSTGRES\_DATABASE: postgres for Supabase by default

POSTGRES\_HOST: db.cqqhkbnuqbqsvifjhqhs.supabase.co.

PORT: 5432 for PostgreSQL, though this is handled by the database connection string

**If you’re using POSTGRES\_URL, it combines the values of**

POSTGRES\_USER,

POSTGRES\_PASSWORD,

POSTGRES\_DATABASE,

POSTGRES\_HOST

into a single string, which simplifies the setup. In that case, only POSTGRES\_URL and PORT may be required.

POSTGRES\_**PASSWORD**

**52454Rev@7TyLo**@db.snbtbdkexdengvubityq.supabase.co:5432/**postgres**

4. Check External Dependencies

If your function relies on external services or APIs (e.g., Supabase), ensure they are responding correctly and in a timely manner. Test these dependencies independently to confirm they are functioning as expected.

**Vercel.json**

{

  "version": 2,

  "builds": [

    {

      "src": "ninja\_gold/wsgi.py",

      "use": "@vercel/python"

    }

  ],

  "routes": [

    {

      "src": "/static/(.\*)",

      "dest": "static/$1"

    },

    {

      "src": "/(.\*)",

      "dest": "ninja\_gold/wsgi.py"

    }

  ],

  "installCommand": "pip install --upgrade setuptools pip wheel && pip install --disable-pip-version-check --upgrade -r requirements.txt"

}

**change nothing** **13f1c68a1065d1cdf5211b9568af97f6a6901353**

test deploy d7b47477fe237ead421ff215df2b22bc374b9412

**git revert 13f1c68a1065d1cdf5211b9568af97f6a6901353**

**URLS.py**

urlpatterns = [

    path('', views.index),

    path('process\_money', views.process\_money),

    path('reset', views.reset),

    path('ninja\_gold/', views.ninja\_gold\_game, name='ninja\_gold\_game'),

    path('reset/', views.reset, name='reset'),

    path('get\_gold\_value/', views.get\_gold\_value, name='get\_gold\_value'),

    # path('supabasedata/', views.your\_view, name='supabasedata'),

]

**Views.py**

import json  # Add this to the top

import os

import random

import requests

from django.conf import settings

from django.http import JsonResponse

# import pygame  # Ensure pygame is installed or remove sound functionality if not needed

from django.shortcuts import HttpResponse, redirect, render

# Test Supabase connection

# def index(request):

#     return JsonResponse({"message": "Function is running successfully!"})

# from ninja\_gold\_app.utils.supabase\_utils import fetch\_data\_from\_table

# def supabasedata(request):

#     try:

#         data = fetch\_data\_from\_table("your\_table\_name")

#         return JsonResponse({"data": data})

#     except Exception as e:

#         return JsonResponse({"error": str(e)}, status=400)

def index(request):

    if 'gold' not in request.session:

        request.session['gold'] = 0

        print(f"Current gold in session: {request.session.get('gold')}")

    if 'activites' not in request.session:

        request.session['activites'] = []

    if 'used\_buildings' not in request.session:

        request.session['used\_buildings'] = []

    return render(request, 'index.html')

def ninja\_gold\_game(request):

    # Define building descriptions

    building\_descriptions = {

        'farm': {

            'name': 'Farm',

            'earn\_message': 'Earns 50-100 ounces of gold',

            'visited\_message': "You've searched on a Farm.",

        },

        'cave': {

            'name': 'Cave',

            'earn\_message': 'Earns 100-250 ounces of gold',

            'visited\_message': "You've searched in a Cave.",

        },

        'house': {

            'name': 'House',

            'earn\_message': 'Earns 20-50 ounces of gold',

            'visited\_message': "You stole from a House.",

        },

        'casino': {

            'name': 'Casino',

            'earn\_message': 'Earn/lose 0-50 ounces of gold',

            'visited\_message': "You've been to the Casino.",

        }

    }

    # Initialize session variables if they don't exist

    if 'gold' not in request.session:

        request.session['gold'] = 0

    if 'activities' not in request.session:

        request.session['activities'] = []

    if 'visited\_buildings' not in request.session:

        request.session['visited\_buildings'] = []

    return render(request, 'ninja\_gold.html', {

        'building\_descriptions': building\_descriptions

    })

def get\_gold\_value(request):

    gold = request.session.get('gold', 0)

    return JsonResponse({'gold': gold})

def process\_money(request):

    print("The form has been submitted!")

    print(request.POST)

    # Safely get the building value from POST request

    building = request.POST.get('building')

    # Initialize session variables if they don't exist

    session\_defaults = {

        'used\_buildings': [],

        'casino\_visits': 0,

        'activites': [],

        'gold': 0,

        'visited\_buildings': [],

    }

    for key, default\_value in session\_defaults.items():

        if key not in request.session:

            request.session[key] = default\_value

    # Define building logic

    building\_logic = {

        'farm': {

            'min\_gold': 50,

            'max\_gold': 100,

            'message': "You Found {gold} ounces of Gold on a Farm! Yay!",

            'sound': 'static/sounds/farm.wav'

        },

        'cave': {

            'min\_gold': 100,

            'max\_gold': 250,

            'message': "You Found {gold} ounces of Gold in a Cave! Yay!",

            'sound': 'static/sounds/cave.wav'

        },

        'house': {

            'min\_gold': 20,

            'max\_gold': 50,

            'message': "You Stole {gold} ounces of Gold from a House! Yay!",

            'sound': 'static/sounds/house.wav'

        },

        'casino': {

            'min\_gold': -50,

            'max\_gold': 50,

            'message\_win': "You Won {gold} ounces of Gold at the Casino! Yay!",

            'message\_loss': "You Lost {gold} ounces of Gold at the Casino! Boooo!!",

            'message\_neutral': "You Won nothing at the Casino. Oh Well...",

            'sound\_win': 'static/sounds/win.wav',

            'sound\_loss': 'static/sounds/loss.wav',

            'sound\_neutral': 'static/sounds/ohwell.wav'

        }

    }

    # Prevent reusing a building (except Casino)

    if building in request.session['used\_buildings'] and building != 'casino':

        return redirect('/')

    # Handle Casino-specific conditions

    if building == 'casino':

        if request.session['casino\_visits'] >= 15:

            request.session['activites'].append(

                "You have reached the maximum number of Casino visits (15). Take the GOLD and Run!"

            )

            return redirect('/')

        if request.session['gold'] <= 0:

            request.session['activites'].append(

                "No Gold! No Casino! Search another area."

            )

            return redirect('/')

    # Process gold earnings/losses for the selected building

    if building in building\_logic:

        if building != 'casino':

            # Regular buildings: Farm, Cave, House

            gold = random.randint(

                building\_logic[building]['min\_gold'], building\_logic[building]['max\_gold']

            )

            message = building\_logic[building]['message'].format(gold=gold)

            sound = building\_logic[building]['sound']

            request.session['gold'] += gold

            request.session['used\_buildings'].append(building)

            request.session['activites'].append(message)

            request.session['sound\_to\_play'] = sound

        elif building == 'casino':

            # Casino logic

            request.session['casino\_visits'] += 1

            gold = random.choices(

                population=[random.randint(-50, -1), random.randint(1, 50), 0],

                weights=[85, 10, 5],  # 85% lose, 10% win, 5% neutral

                k=1

            )[0]

            request.session['gold'] += gold

            if gold > 0:

                message = building\_logic['casino']['message\_win'].format(

                    gold=gold)

                sound = building\_logic['casino']['sound\_win']

            elif gold == 0:

                message = building\_logic['casino']['message\_neutral']

                sound = building\_logic['casino']['sound\_neutral']

            else:

                message = building\_logic['casino']['message\_loss'].format(

                    gold=abs(gold))

                sound = building\_logic['casino']['sound\_loss']

            request.session['activites'].append(message)

            request.session['sound\_to\_play'] = sound

        # Check if all conditions for "loss" are met

    farm\_used = 'farm' in request.session['used\_buildings']

    cave\_used = 'cave' in request.session['used\_buildings']

    house\_used = 'house' in request.session['used\_buildings']

    negative\_gold = request.session['gold'] < 0

    max\_casino\_visits\_reached = request.session['casino\_visits'] >= 15

    request.session['all\_lost\_conditions\_met'] = (

        farm\_used and cave\_used and house\_used and negative\_gold

    )

    print("All Lost Conditions Met:",

          request.session['all\_lost\_conditions\_met'])

    request.session['win\_condition\_met'] = max\_casino\_visits\_reached

    # Save session changes

    request.session.modified = True

    return redirect('/')

def get\_gold\_value(request):

    gold = request.session.get('gold', 0)

    return JsonResponse({'gold': gold})

def reset(request):

    request.session.clear()

    return redirect('/')