**Complete Guide to Shutting Down and Restarting Your AI Tech Stack**

This guide will walk you through the process of properly shutting down your entire tech stack and starting it up again. Each step is explained in detail to ensure clarity and avoid missing critical components.

**Shutting Down Your Tech Stack**

**Step 1: Stop the React Frontend**

1. Go to the PowerShell window where your React frontend is running.
2. Press Ctrl + C on your keyboard.
3. PowerShell will ask if you want to terminate the batch job. Type Y and press Enter.
4. You should see the command prompt return, indicating the React server has stopped.

**Step 2: Stop the FastAPI Backend**

1. Go to the PowerShell window where your FastAPI backend is running.
2. Press Ctrl + C on your keyboard.
3. The backend might take a few seconds to shut down completely.
4. Wait until you see the command prompt return.

**Step 3: Stop Docker Containers**

1. Open a new PowerShell window or use an existing one.
2. Navigate to your project directory:
3. cd C:\StormFolder\my\_project
4. Shut down the Docker containers:
5. docker-compose down
6. This command stops and removes all containers defined in your docker-compose.yml file.
7. Wait until you see a message indicating the containers have been stopped and removed.

**Step 4: Deactivate Virtual Environment**

1. In any PowerShell window where the virtual environment is activated (you'll see (venv) at the beginning of the prompt), type:
2. deactivate
3. The (venv) prefix should disappear from your prompt.

**Step 5: Close PowerShell Windows (Optional)**

1. You can now close all PowerShell windows by clicking the X in the top-right corner.
2. Alternatively, type exit and press Enter in each window.

**Starting Up Your Tech Stack**

**Step 1: Start Docker Desktop**

1. Search for "Docker Desktop" in the Windows start menu and launch it.
2. Wait for Docker Desktop to fully initialize. You'll see the Docker icon in your system tray become solid when it's ready.

**Step 2: Start Docker Containers**

1. Open a new PowerShell window.
2. Navigate to your project directory:
3. cd C:\StormFolder\my\_project
4. Start the Docker containers:
5. docker-compose up -d
6. The -d flag runs containers in the background (detached mode).
7. Wait until you see messages indicating all services have been created.
8. You can verify the containers are running with:
9. docker ps

You should see your pgvector database container listed.  
  
**Temporarily allow scripts (for current session only)**

Run this in your PowerShell **as Administrator**:  
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

**Step 3: Activate the Virtual Environment**

1. In the same PowerShell window or a new one, navigate to your project directory:
2. cd C:\StormFolder\my\_project
3. Activate the virtual environment:
4. .\venv\Scripts\Activate
5. You should now see (venv) at the beginning of your command prompt, indicating the virtual environment is active.
6. Set your database password as an environment variable:  
   $env:DB\_PASSWORD = "Ishinehere1"

**Step 4: Start the FastAPI Backend**

1. With the virtual environment activated, run:
2. python run\_api.py
3. Wait for the messages indicating the server has started, which should look like:
4. INFO: Started server process [XXXXX]INFO: Waiting for application startup.INFO: Application startup complete.INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
5. Keep this PowerShell window open and running.

**Step 5: Start the React Frontend**

1. Open a new PowerShell window.
2. Navigate to the frontend directory:
3. cd C:\StormFolder\my\_project\frontend
4. Set the Node.js environment variable for OpenSSL compatibility:
5. $env:NODE\_OPTIONS = "--openssl-legacy-provider"
6. Start the React development server:
7. npm start
8. Wait for the message indicating the development server has started.
9. A browser window should automatically open to http://localhost:3000 with your application.

**Verifying Everything is Working**

**Check Database Connection**

1. Upload a small test document through the frontend interface.
2. If the upload completes successfully, the database connection is working.

**Check Search Functionality**

1. Use the search feature to query for some text.
2. If results appear, the vector search is working properly.

**Check Text Analysis**

1. Enter some text in the analysis section.
2. Click "Analyze Text" and verify that sentiment, summary, and topics appear.

**Troubleshooting Common Issues**

**Docker Container Issues**

If Docker containers aren't starting:

1. Check if Docker Desktop is running.
2. Try restarting Docker Desktop.
3. Run docker-compose down followed by docker-compose up -d.

**Database Connection Errors**

If the backend can't connect to the database:

1. Verify the Docker containers are running with docker ps.
2. Check for any error messages in the FastAPI terminal.
3. Ensure the database connection parameters match your Docker setup.

**Frontend Not Loading**

If the React frontend doesn't start:

1. Make sure you've set the Node.js environment variable:
2. $env:NODE\_OPTIONS = "--openssl-legacy-provider"
3. Check if there are any error messages in the terminal.
4. Make sure all required NPM packages are installed:
5. npm install

**Backend API Errors**

If the FastAPI backend doesn't start:

1. Ensure your virtual environment is activated ((venv) prefix).
2. Check that all Python dependencies are installed:
3. pip install -r requirements.txt
4. Look for syntax errors in the terminal output.

**Important Notes**

* Always shut down components in the order listed: Frontend → Backend → Docker → Virtual Environment
* Always start components in the reverse order: Virtual Environment → Docker → Backend → Frontend
* Never close PowerShell windows directly while services are running; use Ctrl+C first
* If you restart your computer, you'll need to restart Docker Desktop before starting the containers

By following this detailed guide, you should be able to reliably shut down and restart your complete AI tech stack without issues.