RENTAHAL Token Gatekeeper Integration Guide

This guide explains how to integrate the RENTAHAL Token Gatekeeper with your existing monolithic script.js and webgui.py codebase. The gatekeeper ensures that users must connect a wallet and transfer a $9000 token before accessing the RENTAHAL interface.

Overview

The integration consists of two main components:

1. **Frontend JavaScript** - A bolt-on script that creates a gatekeeper UI overlay
2. **Backend Python** - Routes and middleware for your FastAPI application

This approach allows you to implement the gatekeeper without modifying your core application code.

Integration Steps

1. Backend Integration (webgui.py)

Add the following code to your existing webgui.py file:

```
python
```

```python
# At the top of your file, add these imports if not already present
from fastapi import APIRouter, Request, Response
from fastapi.middleware.base import BaseHTTPMiddleware
from fastapi.staticfiles import StaticFiles
import time
import random
import json
import os
from typing import Dict, Any, Optional

# Create a gatekeeper router
gatekeeper_router = APIRouter()

# Add these routes to your FastAPI app
@app.include_router(gatekeeper_router)

# Create directory for static files if it doesn't exist
os.makedirs("static", exist_ok=True)

# Add the middleware to check authentication
class GatekeeperMiddleware(BaseHTTPMiddleware):
    async def dispatch(self, request: Request, call_next):
        # Skip API endpoints and static files
        if request.url.path.startswith("/api/") or request.url.path.startswith("/ws") or reques
            return await call_next(request)

        # Check for authentication token in cookies
        auth_cookie = request.cookies.get("rentahal_authenticated")

        # If authenticated, proceed normally
        if auth_cookie == "true":
            return await call_next(request)

        # Otherwise, inject the gatekeeper
        response = await call_next(request)

        # Only modify HTML responses
        if response.headers.get("content-type", "").startswith("text/html"):
            content = b""
            async for chunk in response.body_iterator:
                content += chunk

            html_content = content.decode("utf-8")
```

```
# Inject our gatekeeper script
gatekeeper_script = """
<script src="/static/gatekeeper.
```