

AI Model (scikit-learn) Zoning Predictor (Python)

```
-----  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.model_selection import train_test_split  
import pandas as pd  
  
data = pd.read_csv("urban_zoning_data.csv")  
X = data[['traffic_density', 'land_value', 'green_space_index']]  
y = data['recommended_zone']  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)  
model = RandomForestClassifier()  
model.fit(X_train, y_train)  
print("Accuracy:", model.score(X_test, y_test))
```

3D Visualization Trigger (JavaScript + Three.js)

```
-----  
import * as THREE from 'three';  
const scene = new THREE.Scene();  
const camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.1, 1000);  
const renderer = new THREE.WebGLRenderer();  
document.body.appendChild(renderer.domElement);  
camera.position.z = 5;  
const geometry = new THREE.BoxGeometry();  
const material = new THREE.MeshBasicMaterial({ color: 0x0077ff });  
const cube = new THREE.Mesh(geometry, material);  
scene.add(cube);  
function animate() {  
    requestAnimationFrame(animate);  
    cube.rotation.y += 0.01;  
    renderer.render(scene, camera);  
}  
animate();
```

IoT Dashboard (Flask + Chart.js)

```
-----  
@app.route('/dashboard')  
def dashboard():  
    air_quality = get_sensor_data('air_quality')  
    noise = get_sensor_data('noise')  
    return render_template("dashboard.html", air_quality=air_quality, noise=noise)
```