```
#!/bin/bash

# Check for GPG installation
if ! command -v gpg &> /dev/null; then
    echo "GPG is not installed. Install it first (e.g., 'sudo apt install gnupg')."
    exit 1
fi

# Prompt for user input
echo "Setting up GPG for the Axium project..."
read -p "Enter your name (e.g., Axium Project): " NAME
read -p "Enter your email (e.g., your-email@example.com): " EMAIL

# Generate GPG key
echo "Generating GPG key..."
gpg --batch --gen-key <<EOF
%no-protection
Key-Type: RSA
Key-Length: 4096
Subkey-Type: RSA
Subkey-Length: 4096
Name-Real: $NAME
Name-Email: $EMAIL
Expire-Date: 1y
EOF

# Fetch the key ID
KEY_ID=$(gpg --list-keys --with-colons | grep -m 1 pub | cut -d: -f5)

# Export public key
PUBLIC_KEY_FILE="axium_public_key.asc"
gpg --export --armor $KEY_ID > $PUBLIC_KEY_FILE
echo "Public key exported to $PUBLIC_KEY_FILE."

# Sign a sample JavaScript file
SAMPLE_FILE="axium.js"
echo "// Axium project JavaScript file" > $SAMPLE_FILE
gpg --detach-sign --armor $SAMPLE_FILE
echo "Signed $SAMPLE_FILE. Signature file: $SAMPLE_FILE.asc"

# Optional: Configure Git to use this GPG key for signing
read -p "Would you like to configure Git to sign commits with this GPG key? (y/n): "
CONFIGURE_GIT
if [[ $CONFIGURE_GIT =~ ^[Yy]$ ]]; then
```

```
    git config --global user.signingkey $KEY_ID
    git config --global commit.gpgsign true
    echo "Git is now configured to sign commits with your GPG key."
fi

echo "GPG setup for the Axium project is complete!"
chmod +x setup_gpg_axium.sh
./setup_gpg_axium.sh
```