AWS Credentials Issue - RESOLVED ✅

Problem Summary

Sam's mySafePlay Core Safety Loop system was experiencing AWS credential errors despite having valid credentials configured in the `.env` file. The errors included:
- "invalid security token"
- "resolved credential object is not valid"

Root Cause Analysis

The issue was **environment variable conflicts**. The system had AWS environment variables set glob-
ally (from the Abacus.AI hosting environment) that were overriding Sam's credentials from the `.env` file:

Conflicting System Variables:

```
AWS_ACCESS_KEY_ID=ASIAWRCFZRKU3CG7Q2TP (temporary session token)
AWS_SECRET_ACCESS_KEY=ogNmRxcI... (temporary session token)
AWS_SESSION_TOKEN=FwoGZXIv... (temporary session token)
```

Sam's Correct Credentials (from .env):

```
AWS_ACCESS_KEY_ID=AKIASTDCWLU6VZIRMM5B (permanent access key)
AWS_SECRET_ACCESS_KEY=xXaCwZTQ... (permanent secret key)
AWS_REGION=us-east-2
```

The system was trying to use the temporary/invalid session tokens instead of Sam's valid permanent credentials.

Solution Implemented

1. Created AWS Configuration Utility

- **File**: `lib/utils/aws-config.ts`
- **Purpose**: Forces the system to read credentials directly from the `.env` file, ignoring system envir-
onment variables
- **Key Functions**:
- `getAWSCredentialsFromEnv()`: Reads `.env` file directly
- `createAWSConfig()`: Creates AWS SDK configuration with correct credentials

2. Updated Core Safety Loop Services

- **Updated**: `lib/services/real-time-face-recognition-service.ts`
- **Updated**: `lib/services/core-safety-loop-integration-service.ts`
- **Change**: Modified to use `createAWSConfig()` instead of relying on environment variables

3. Fixed Test Scripts

- **Updated**: `test-core-safety-loop.js`
- **Change**: Updated AWS connection test to use the same credential loading logic

Verification Results

Before Fix:

```
❌ Error: The security token included in the request is invalid
❌ AWS Configuration: Issues detected
❌ Rekognition: Permissions needed
```

After Fix:

```
✅ STS Test Passed: Account: 178448129341
✅ Rekognition Test Passed: Collections found: 0
✅ AWS Configuration: Valid
✅ Rekognition: Connected
```

Current Status

- ✅ **AWS Credentials**: Working perfectly
- ✅ **Core Safety Loop**: All 9/9 files present and functional
- ✅ **API Endpoints**: All 6 endpoints implemented
- ✅ **Rekognition Access**: Connected and ready

Next Steps for Sam

1. **Set up face collections**: Run `node scripts/setup-face-collections.js`
2. **Enroll children faces**: Use the UI to add face data
3. **Test real-time recognition**: Access `/venue-admin/core-safety-loop`
4. **Monitor system**: Use `/api/system/aws-status` for health checks

Technical Details

- **AWS Account**: 178448129341
- **IAM User**: SafePlay-app
- **User ARN**: arn:aws:iam::178448129341:user/SafePlay-app
- **Region**: us-east-2
- **S3 Bucket**: safeplay-faces

Files Modified

1. `lib/utils/aws-config.ts` (new)
2. `lib/services/real-time-face-recognition-service.ts` (updated)
3. `lib/services/core-safety-loop-integration-service.ts` (updated)
4. `test-core-safety-loop.js` (updated)

The Core Safety Loop system is now fully operational and ready for face collection initialization! 🚀