ShotStack Integration - Complete Index

Project: Basketball Shooting Analysis
Date: December 13, 2025
Status: ✅ COMPLETE

📁 File Structure

```
/home/ubuntu/basketball_app/
│
├── Configuration
│   └── .env.shotstack                    (839 bytes)   API credentials
│
├── Python Scripts
│   ├── shotstack_helpers.py              (19 KB)       Main integration library
│   ├── shotstack_test.py                 (9.1 KB)      Test suite
│   └── shotstack_example.py              (8.9 KB)      Usage examples
│
└── Documentation
    ├── SHOTSTACK_SETUP.md                (14 KB)       Setup guide
    ├── SHOTSTACK_INTEGRATION_GUIDE.md    (15 KB)       Developer guide
    ├── SHOTSTACK_INTEGRATION_GUIDE.pdf   (124 KB)      PDF version
    ├── SHOTSTACK_COMPLETE_SETUP.md       (12 KB)       Summary
    ├── SHOTSTACK_FINAL_REPORT.md         (17 KB)       Final report
    ├── SHOTSTACK_FINAL_REPORT.pdf        (176 KB)      PDF version
    ├── SHOTSTACK_QUICK_REFERENCE.md      (3.4 KB)      Quick reference
    ├── SHOTSTACK_QUICK_REFERENCE.pdf     (88 KB)       PDF version
    └── SHOTSTACK_INDEX.md                (This file)   Index
```

Total: 12 files, ~500 KB

📖 Documentation Guide

For Quick Start

Read: `SHOTSTACK_QUICK_REFERENCE.md`
Purpose: Get started in 5 minutes
Contains: API keys, basic usage, common tasks

For Setup

Read: `SHOTSTACK_SETUP.md`
Purpose: Complete setup guide
Contains: API credentials, capabilities, templates, examples

For Development

Read: `SHOTSTACK_INTEGRATION_GUIDE.md`
Purpose: Developer integration guide
Contains: Usage patterns, examples, best practices, troubleshooting

For Overview

Read: `SHOTSTACK_COMPLETE_SETUP.md`
Purpose: High-level summary
Contains: Deliverables, capabilities, integration points

For Final Report

Read: `SHOTSTACK_FINAL_REPORT.md`
Purpose: Complete project report
Contains: Everything - completion status, metrics, next steps

---

🔧 Script Guide

Main Library

File: `shotstack_helpers.py`
Classes:
- `ShotStackClient` - Low-level API client
- `BasketballVideoEditor` - High-level basketball editor

Functions:
- `create_basketball_analysis_video()` - Main integration function

Usage:

```
from shotstack_helpers import ShotStackClient, BasketballVideoEditor

client = ShotStackClient(environment='sandbox')
editor = BasketballVideoEditor(client)
```

Test Suite

File: `shotstack_test.py`
Tests:
1. API connection test
2. Simple text overlay test
3. Basketball video editor test
4. Split-screen comparison test

Usage:

```
python shotstack_test.py
```

Examples

File: `shotstack_example.py`
Examples:
1. Simple text annotation
2. Angle measurements
3. Complete shooting form analysis

4. Split-screen comparison
5. Raw JSON template

Usage:

```
python shotstack_example.py
```

🔑 API Credentials

Location

File: `.env.shotstack`

Sandbox

```
API Key: 5I9pXTQbDLmcF6tvgjOzgYtDN5jyK2FnurBSU5oy
Endpoint: https://api.shotstack.io/edit/stage
```

Production

```
API Key: HQNZcbuBHc1zVapRhzAdHQFqNkXzQG1YrqYhBhwZ
Endpoint: https://api.shotstack.io/edit/v1
```

🎯 Use Cases

Use Case 1: Shooting Form Analysis

Script: `shotstack_helpers.py`
Function: `create_shooting_form_analysis()`
Features:
- Original video
- Skeleton overlay
- Angle measurements
- Text annotations

Example:

```
response = editor.create_shooting_form_analysis(
    video_url="shot.mp4",
    annotations=[...],
    angles=[...],
    duration=5.0
)
```

Use Case 2: Before/After Comparison

Script: `shotstack_helpers.py`
Function: `create_split_screen_comparison()`

Features:
- Two videos side-by-side
- Titles
- Divider line

Example:

```
response = editor.create_split_screen_comparison(
    video1_url="before.mp4",
    video2_url="after.mp4",
    title1="Before",
    title2="After"
)
```

Use Case 3: Custom Analysis

Script: `shotstack_helpers.py`
Function: `create_basketball_analysis_video()`
Features:
- Full pipeline integration
- RoboFlow data input
- Analysis results input
- Complete video output

Example:

```
output_url = create_basketball_analysis_video(
    video_path=video_url,
    skeleton_data=pose_data,
    analysis_results=analysis,
    output_path="analysis.mp4"
)
```

🏀 Basketball Features

Feature 1: Skeleton Overlay

Implementation: ImageAsset with opacity
Documentation: `SHOTSTACK_SETUP.md` - Section “Basketball Analysis Features”
Code: `shotstack_helpers.py` - Line ~150

Feature 2: Angle Measurements

Implementation: ShapeAsset (circles) + TextAsset
Documentation: `SHOTSTACK_SETUP.md` - Section “Basketball Analysis Features”
Code: `shotstack_helpers.py` - Line ~400

Feature 3: Coaching Annotations

Implementation: TextAsset with timing
Documentation: `SHOTSTACK_SETUP.md` - Section “Basketball Analysis Features”
Code: `shotstack_helpers.py` - Line ~450

Feature 4: Split-Screen

Implementation: Multiple video tracks
Documentation: `SHOTSTACK_SETUP.md` - Section “Basketball Analysis Features”
Code: `shotstack_helpers.py` - Line ~250

🔗 Integration Points

With RoboFlow

Documentation: `SHOTSTACK_INTEGRATION_GUIDE.md` - Section “Integration with RoboFlow”
Workflow:
1. Get video from user
2. Run RoboFlow pose estimation
3. Calculate angles
4. Generate feedback
5. Create ShotStack video
6. Return to user

With Basketball App

Documentation: `SHOTSTACK_FINAL_REPORT.md` - Section “Integration Architecture”
Components:
- Frontend: Video upload, results display
- Backend: Processing pipeline
- Database: Store results
- ShotStack: Video enhancement

📊 Quick Reference

API Endpoints

```
Edit:   https://api.shotstack.io/edit/{version}
Serve:  https://api.shotstack.io/serve/{version}
Ingest: https://api.shotstack.io/ingest/{version}
```

Common Commands

```
# Test connection
python -c "from shotstack_helpers import ShotStackClient; print('✓ OK')"

# Run tests
python shotstack_test.py

# Run examples
python shotstack_example.py
```

Position System

```
# Named positions
position = "top" | "center" | "bottom" | "left" | "right"

# Offset coordinates
offset = {'x': 0.2, 'y': -0.1}  # -1 to 1
```

Colors

```
'#00ff00'  # Green - Good
'#ffff00'  # Yellow - Warning
'#ff0000'  # Red - Needs work
```

🧪 Testing

Quick Test

```
cd /home/ubuntu/basketball_app
python -c "from shotstack_helpers import ShotStackClient; c = ShotStackCli-
ent('sandbox'); print('✓ Connected')"
```

Full Test Suite

```
python shotstack_test.py
```

Example Demonstrations

```
python shotstack_example.py
```

📚 Learning Path

Beginner

1. Read `SHOTSTACK_QUICK_REFERENCE.md`
2. Run `shotstack_example.py`
3. Try basic annotation example

Intermediate

1. Read `SHOTSTACK_SETUP.md`
2. Understand JSON templates
3. Create custom annotations

Advanced

1. Read `SHOTSTACK_INTEGRATION_GUIDE.md`
2. Integrate with RoboFlow

3. Build complete pipeline

🎓 Key Concepts

Multi-Track Timeline

Videos are composed of multiple tracks (layers) that stack on top of each other.

Documentation: `SHOTSTACK_SETUP.md` - Section “Template Structure”

Asset Types

Different types of content: video, image, text, shape, audio.

Documentation: `SHOTSTACK_SETUP.md` - Section “Available Assets”

Timing System

Control when clips start and how long they play.

Documentation: `SHOTSTACK_INTEGRATION_GUIDE.md` - Section “Customization Options”

Position & Offset

Place elements precisely on the video frame.

Documentation: `SHOTSTACK_INTEGRATION_GUIDE.md` - Section “Customization Options”

🔍 Troubleshooting

Issue: API Connection Failed

Solution: Check `.env.shotstack` file exists and has correct keys
Documentation: `SHOTSTACK_INTEGRATION_GUIDE.md` - Section “Debugging”

Issue: Render Failed

Solution: Validate JSON structure, check video URLs
Documentation: `SHOTSTACK_INTEGRATION_GUIDE.md` - Section “Debugging”

Issue: Video Not Loading

Solution: Ensure video URL is publicly accessible
Documentation: `SHOTSTACK_INTEGRATION_GUIDE.md` - Section “Debugging”

📞 Support

Documentation

- **Setup:** `SHOTSTACK_SETUP.md`
- **Integration:** `SHOTSTACK_INTEGRATION_GUIDE.md`
- **Quick Ref:** `SHOTSTACK_QUICK_REFERENCE.md`

External Resources

- **Dashboard:** https://dashboard.shotstack.io
- **API Docs:** https://shotstack.io/docs/
- **Support:** https://shotstack.io/support/

✅ Completion Status

Phase 1: Setup ✅

- [x] API credentials obtained
- [x] Capabilities explored
- [x] Documentation reviewed

Phase 2: Development ✅

- [x] Integration scripts created
- [x] Test suite built
- [x] Examples provided

Phase 3: Documentation ✅

- [x] Setup guide written
- [x] Integration guide written
- [x] Quick reference created
- [x] Final report completed

Phase 4: Testing ✅

- [x] API connection verified
- [x] Test suite created
- [x] Examples validated

Phase 5: Delivery ✅

- [x] All files created
- [x] Documentation complete
- [x] Ready for integration

🎉 Summary

Status: ✅ COMPLETE
Files Created: 12
Documentation Pages: ~50
Code Lines: ~1,000
Ready for: Production Integration

🚀 Next Steps

1. ⏳ Test with real basketball video
2. ⏳ Integrate with RoboFlow
3. ⏳ Build upload system
4. ⏳ Deploy to production

Last Updated: December 13, 2025
Version: 1.0
Status: FINAL

Complete Index - All ShotStack Integration Files