mySafePlay Rollback Instructions

This document provides step-by-step instructions for rolling back to previous stable versions of the mySafePlay application.

🎯 Available Stable Versions

Version 0.5 (Current Stable)

- **Tag:** v0.5
- **Date:** July 6, 2025
- **Status:** Production Ready
- **Features:** Complete biometric application with all core functionality
- **Backup:** `safeplay-v0.5-YYYYMMDD.tar.gz`

🔄 Rollback Methods

Method 1: Git Checkout (Recommended)

To View Available Versions

```
cd /home/ubuntu/safeplay-staging
git tag --list -n5
```

To Rollback to Version 0.5

```
cd /home/ubuntu/safeplay-staging
git checkout v0.5
```

To Create New Branch from Version 0.5

```
cd /home/ubuntu/safeplay-staging
git checkout -b rollback-to-v0.5 v0.5
```

Method 2: Reset Main Branch (Destructive)

⚠️ **WARNING:** This method permanently removes commits after the specified version.

```
cd /home/ubuntu/safeplay-staging
git reset --hard v0.5
git push --force-with-lease origin main
```

Method 3: Backup Restoration

Extract from Backup

```
cd /home/ubuntu
tar -xzf safeplay-v0.5-YYYYMMDD.tar.gz
mv safeplay-staging safeplay-staging-current
mv safeplay-staging-backup safeplay-staging
```

🚀 Post-Rollback Deployment

1. Verify Environment

```
cd /home/ubuntu/safeplay-staging
npm install
npm run build
```

2. Check Database Connection

```
npx prisma generate
npx prisma db push
```

3. Test Application Locally

```
npm run dev
```

4. Deploy to Production

```
# If using Vercel CLI
vercel --prod

# Or push to trigger automatic deployment
git push origin main
```

🔍 Verification Steps

After rollback, verify these components:

✅ Application Status

- [ ] Application loads at production URL
- [ ] Authentication system working
- [ ] Demo accounts accessible
- [ ] Database connectivity confirmed
- [ ] API endpoints responding

✅ Demo Account Testing

Test these accounts after rollback:
- [ ] parent1@example.com / password123

- [ ] admin@safeplay.com / admin123
- [ ] New user registration working

✅ Core Features

- [ ] User login/logout
- [ ] Child profile creation
- [ ] Biometric data entry
- [ ] Analytics dashboard
- [ ] Alert configurations

🆘 Emergency Procedures

If Rollback Fails

1. **Check Git Status**

```
bash
   git status
   git log --oneline -10
```

2. **Force Clean State**

```
bash
   git clean -fd
   git reset --hard HEAD
```

3. **Restore from Backup**

```
bash
   cd /home/ubuntu
   rm -rf safeplay-staging
   tar -xzf safeplay-v0.5-YYYYMMDD.tar.gz
```

If Deployment Fails

1. **Check Environment Variables**
   - Verify DATABASE_URL
   - Confirm NEXTAUTH_SECRET
   - Check all required env vars

2. **Database Issues**

```
bash
   npx prisma reset
   npx prisma db push
   npx prisma generate
```

3. **Dependency Issues**

```
bash
   rm -rf node_modules package-lock.json
   npm install
```

📞 Support Information

Version Information

- **Current Stable:** v0.5

- **Production URL:** https://safeplay-staging-drsammd-my-safe-play.vercel.app
- **Repository:** https://github.com/drsammd/safeplay-staging

Backup Locations

- **Git Tags:** All versions tagged in repository
- **File Backups:** `/home/ubuntu/safeplay-v*.tar.gz`
- **Documentation:** `/home/ubuntu/safeplay-staging/RELEASES/`

Contact

- **Project Stakeholder:** Sam
- **Development Team:** AI Development Team
- **Repository:** GitHub - drsammd/safeplay-staging

📋 Rollback Checklist

Before performing rollback:
- [ ] Backup current state if needed
- [ ] Identify target version
- [ ] Verify backup availability
- [ ] Plan deployment strategy
- [ ] Prepare verification tests

After rollback:
- [ ] Verify application functionality
- [ ] Test demo accounts
- [ ] Confirm database connectivity
- [ ] Check production deployment
- [ ] Document rollback completion

Remember: Always test rollback procedures in a development environment before applying to production.