MY PORTFOLIO

Project Documentation

A Modern 3D Web Portfolio with React & Three.js

Author: Mayank Chouhan

Email: mayankchouhan@example.com | GitHub: https://github.com/MAYANK2264

Date: August 2025 | Version 1.0.0

Table of Contents

1. Introduction & Architecture

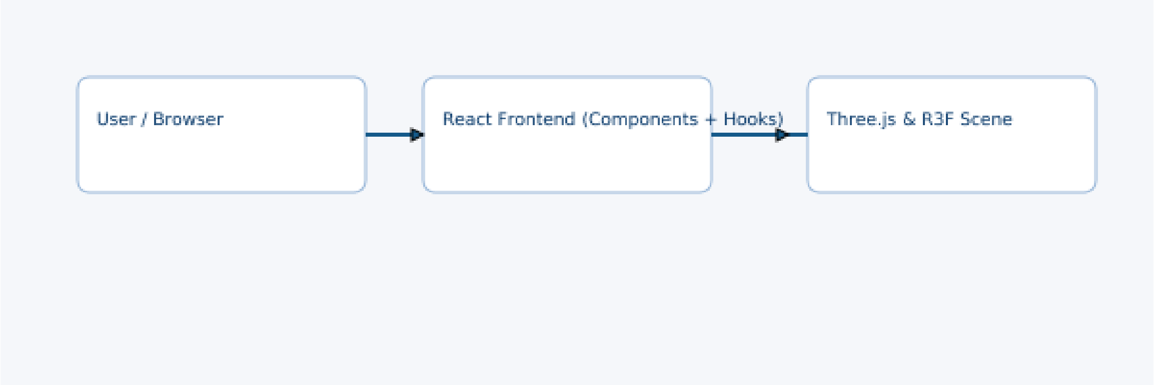
5ba16293-7378-4788-ada2-78084e4bcfdc.md

12. Troubleshooting & Common Issues

12.1 Three.js Errors
Problem: `SphereBufferGeometry is not part of the THREE namespace`
Solution: Use `sphereGeometry` instead of `sphereBufferGeometry` (deprecated)

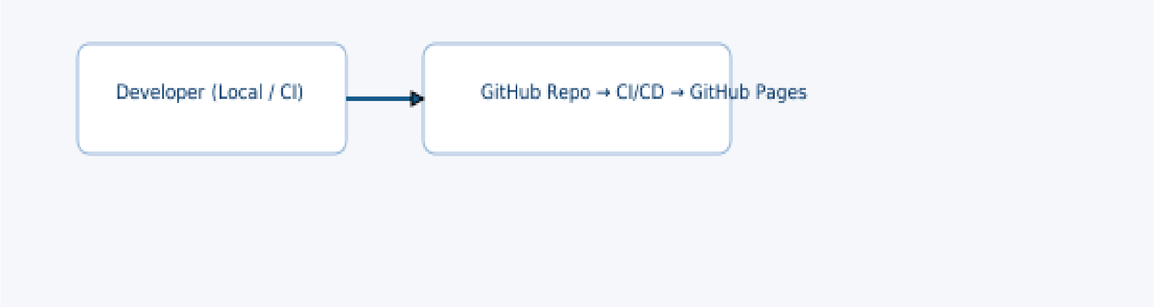
Problem: Performance issues with particle systems
Solution: Reduce particle count, implement LOD, use object pooling

Architecture Diagram



2. Core Features & 3D Implementation

Deployment Pipeline (visual)



4. Troubleshooting & Recruiter Q&A;

~~~
Troubleshooting & Common Issues
### 12.1 Three.js Errors
**Problem**: `SphereBufferGeometry is not part of the THREE namespace`
**Solution**: Use `sphereGeometry` instead of `sphereBufferGeometry` (deprecated)
**Problem**: Performance issues with particle systems
**Solution**: Reduce particle count, implement LOD, use object pooling
**Problem**: WebGL context lost
**Solution**: Implement context restoration, add error boundaries
### 12.2 React Three Fiber Issues
**Problem**: Canvas not rendering
**Solution**: Check WebGL support, verify Three.js version compatibility
**Problem**: Memory leaks in 3D scenes
**Solution**: Implement proper cleanup in useEffect return functions
**Problem**: Component not updating in 3D scene
**Solution**: Use useFrame hook properly, check dependency arrays
### 12.3 Build & Deployment Issues
**Problem**: GitHub Pages 404 errors
**Solution**: Use HashRouter, verify base path in Vite config
**Problem**: Assets not loading
**Solution**: Check file paths, verify public directory structure
**Problem**: Build fails with Three.js
**Solution**: Check import statements, verify package versions
### 12.4 Performance Issues
**Problem**: Slow animations on mobile
**Solution**: Reduce particle count, implement device detection
**Problem**: Large bundle size
**Solution**: Code splitting, tree shaking, asset optimization
**Problem**: Memory usage growing
**Solution**: Implement proper cleanup, dispose of Three.js resources
### 12.5 Common Debugging Steps
```javascript
// 1. Check browser console for errors
console.log('Debug info:', { scene, camera, renderer });
// 2. Verify Three.js objects exist
if (meshRef.current) {
console.log('Mesh position:', meshRef.current.position);
}
// 3. Monitor frame rate
let frameCount = 0;
useFrame(() => {
frameCount++;
if (frameCount % 60 === 0) {
console.log('FPS:', 60 / (Date.now() - lastTime) * 1000);
lastTime = Date.now();
}
});
// 4. Check memory usage
console.log('Memory:', performan
~~~

Future Enhancements

13.1 Technical Improvements
- **TypeScript Migration**: Add type safety to the codebase
- **PWA Support**: Progressive web app capabilities
- **Service Worker**: Offline functionality and caching
- **WebGL 2.0**: Enhanced 3D graphics capabilities

13.2 Feature Additions
- **Blog Section**: Technical articles and tutorials
- **Portfolio Filters**: Advanced project categorization
- **Dark/Light Theme**: User preference toggle
- **Multi-language Support**: Internationalization

13.3 Performance Enhancements
- **Web Workers**: Background processing for physics
- **WebAssembly**: Performance-critical calculations
- **Virtual Scrolling**: Large list optimization
- **Image Optimization**: WebP format and lazy loading

13.4 User Experience
- **Voice Navigation**: Accessibility improvements
- **Gesture Controls**: Touch and mouse gesture support
- **Keyboard Shortcuts**: Power user navigation
- **Analytics Integration**: User behavior tracking

13.5 Advanced 3D Features
- **Raycasting**: Interactive 3D object selection
- **Post-processing**: Advanced visual effects
- **Physics Engine**: More realistic simulations
- **VR/AR Support**: Immersive experiences

13.6 Implementation Roadmap
```javascript
// Phase 1: Core improvements (Month 1-2)
const phase1 = [
'TypeScript migration',
'Performance optimization',
'Testing implementation',
'SEO optimization'
];

// Phase 2: Feature expansion (Month 3-4)
const phase2 = [
'Blog system',
'Advanced filtering',
'Theme system',
'Analytics integration'
];

// Phase 3: Advanced features (Month 5-6)
const phase3 = [
'PWA implementation',
'Advanced 3D features',
'Internationalization',
'Performance monitoring'
```

];
```

---

14. Recruiter Questions & Answers

14.1 Technical Questions

Q: Why did you choose React for this project?
A: React was chosen for its component-based architecture, excellent ecosystem, and modern features like hooks and concurrent rendering. It
```

## Quick Reference Commands

```
// 1. Check browser console for errors
console.log('Debug info:', { scene, camera, renderer });
// 2. Verify Three.js objects exist
if (meshRef.current) {
console.log('Mesh position:', meshRef.current.position);
}
// 3. Monitor frame rate
let frameCount = 0;
useFrame(() => {
frameCount++;
if (frameCount % 60 === 0) {
console.log('FPS:', 60 / (Date.now() - lastTime) * 1000);
lastTime = Date.now();
}
});
// 4. Check memory usage
console.log('Memory:', performance.memory);
// Phase 1: Core improvements (Month 1-2)
const phase1 = [
'TypeScript migration',
'Performance optimization',
'Testing implementation',
'SEO optimization'
];
// Phase 2: Feature expansion (Month 3-4)
const phase2 = [
'Blog system',
'Advanced filtering',
'Theme system',
'Analytics integration'
];
// Phase 3: Advanced features (Month 5-6)
const phase3 = [
'PWA implementation',
'Advanced 3D features',
'Internationalization',
'Performance monitoring'
];
# Development
npm run dev # Start development server
npm run build # Build for production
npm run preview # Preview production build
npm run lint # Run ESLint
npm run deploy # Deploy to GitHub Pages
# Git operations
git status # Check repository status
git add . # Stage all changes
git commit -m "msg" # Commit changes
git push origin main # Push to remote repository
# Dependencies
npm install # Install dependencies
npm update # Update dependencies
npm audit # Check for vulnerabilities
npm audit fix # Fix vulnerabilities
```

Skills Demonstrated

- React 18, JavaScript (ES6+), HTML5, CSS3 (Tailwind)
- Three.js, React Three Fiber, WebGL, Custom physics & particle systems
- Performance optimization: LOD, object pooling, code splitting
- Build & Deployment: Vite, GitHub Pages, CI/CD