Cross-Platform npm Migration Validation Guide

Project: QuantumLeap AI Website
Purpose: Validate npm migration works on Windows, Mac, and Linux
Date: November 10, 2025

🎯 Test Objectives

Verify that the Yarn → npm migration works correctly across all major operating systems with fresh repository clones.

✅ Pre-Test Requirements

Required Software

- **Git:** Latest version
- **Node.js:** v18.x or v20.x (LTS recommended)
- **npm:** v10.x (comes with Node.js)
- **Terminal/Command Line:**
- Windows: PowerShell or CMD
- Mac: Terminal.app
- Linux: Bash/Zsh

Repository Access

- GitHub URL: `https://github.com/AICyberSecurity911/WBsite.git`
- Branch: `main` or latest migration branch
- Ensure you have read access to the repository

📋 Test Procedure (All Platforms)

Step 1: Fresh Clone

```
# Navigate to your workspace
cd ~/workspace  # Mac/Linux
cd C:\workspace  # Windows

# Clone repository
git clone https://github.com/AICyberSecurity911/WBsite.git
cd WBsite/nextjs_space

# Verify no Yarn files exist
ls -la | grep yarn  # Mac/Linux
dir | findstr yarn  # Windows

# Expected: No yarn.lock or .yarnrc files (only package-lock.json)
```

Step 2: Verify Node/npm Versions

```
node --version  # Should be v18.x or v20.x
npm --version   # Should be v10.x

# Document versions for your test report
```

Step 3: Install Dependencies

```
# CRITICAL: Must use --legacy-peer-deps flag
npm install --legacy-peer-deps

# Monitor for:
# ✅ No Yarn-related errors
# ✅ No "command not found: yarn" messages
# ✅ package-lock.json created/updated
# ✅ 1,333+ packages installed
# ❌ Any "ERESOLVE" errors (should not occur with flag)
```

Expected Output:

```
added 1333 packages, and audited 1334 packages in 2-4m
211 packages are looking for funding
5 vulnerabilities (2 low, 3 moderate)
```

Step 4: Development Server Test

```
npm run dev

# Wait for server to start (30-60 seconds)
# Expected output:
# ▲ Next.js 14.2.28
# - Local: http://localhost:3000
# ✓ Ready in Xs
```

Browser Verification:
1. Open `http://localhost:3000`
2. Verify homepage loads with hero section
3. Check console for errors (F12)
4. Navigate to 2-3 pages (About Us, SMB, Services)
5. Verify no hydration errors or Yarn warnings

Stop server: `Ctrl+C`

Step 5: Production Build Test

```
npm run build

# Monitor for:
# ✓ Compiled successfully
# ✓ Generating static pages (41/41)
# ❌ Any TypeScript errors
# ❌ Any Yarn-related build failures
```

Expected Output:

```
▲ Next.js 14.2.28
- Environments: .env
Creating an optimized production build ...
✓ Compiled successfully
✓ Linting and checking validity of types
✓ Collecting page data
✓ Generating static pages (41/41)
✓ Collecting build traces
✓ Finalizing page optimization

Route (app)                              Size     First Load JS
┌ ○ /                                    5.2 kB         100 kB
├ ○ /about-us                            8.1 kB         103 kB
├ ○ /smb                                 7.5 kB         102 kB
...
```

Step 6: Production Server Test

```
npm start

# Expected output:
# ▲ Next.js 14.2.28
# - Local: http://localhost:3000
# - Production mode
```

Browser Verification:
1. Open `http://localhost:3000`
2. Verify production build loads
3. Check page load speed (should be faster than dev)
4. Verify all assets load correctly

Stop server: `Ctrl+C`

🖥️ Platform-Specific Considerations

Windows

Common Issues:

- Long file paths may cause errors → Use shorter clone paths (C:\work)
- Line ending conversion → Ensure `.gitattributes` handles CRLF
- npm cache issues → Run `npm cache clean --force` if install fails

PowerShell Commands:

```
# Check for Yarn files
Get-ChildItem -Filter "*yarn*" -Recurse -File | Select-Object FullName

# Clean install
Remove-Item node_modules -Recurse -Force
Remove-Item package-lock.json -Force
npm install --legacy-peer-deps
```

macOS

Common Issues:

- Permission errors → Never use `sudo` with npm
- Xcode Command Line Tools required → `xcode-select --install`
- M1/M2 ARM compatibility → Node v18+ handles this automatically

Bash/Zsh Commands:

```
# Check for Yarn files
find . -name "*yarn*" -not -path "./node_modules/*"

# Clean install
rm -rf node_modules package-lock.json
npm install --legacy-peer-deps
```

Linux (Ubuntu/Debian)

Common Issues:

- Node version from apt may be outdated → Use nvm or NodeSource
- Build tools missing → `sudo apt install build-essential`
- Port 3000 already in use → Change port with `PORT=3001 npm run dev`

Commands:

```
# Check for Yarn files
find . -name "*yarn*" -not -path "./node_modules/*"

# Clean install
rm -rf node_modules package-lock.json
npm install --legacy-peer-deps
```

📊 Test Report Template

Use this template to document your results:

```
## Cross-Platform Test Results

**Tester:** [Your Name]
**Date:** [Test Date]
**Repository Commit:** [Git SHA]

### Platform 1: Windows
- **OS Version:** Windows 11 Pro 23H2
- **Node Version:** v20.11.0
- **npm Version:** v10.9.2
- **Test Results:**
  - [ ] Clone successful
  - [ ] No Yarn files present
  - [ ] npm install completed (Time: ___ minutes)
  - [ ] npm run dev works
  - [ ] npm run build succeeds
  - [ ] npm start functions
  - [ ] No Yarn-related errors
- **Issues Encountered:** None / [Describe]
- **Screenshots:** [Attach if issues found]

### Platform 2: macOS
- **OS Version:** macOS Sonoma 14.5
- **Node Version:** v20.11.0
- **npm Version:** v10.9.2
- **Test Results:**
  - [ ] Clone successful
  - [ ] No Yarn files present
  - [ ] npm install completed (Time: ___ minutes)
  - [ ] npm run dev works
  - [ ] npm run build succeeds
  - [ ] npm start functions
  - [ ] No Yarn-related errors
- **Issues Encountered:** None / [Describe]
- **Screenshots:** [Attach if issues found]

### Platform 3: Linux (Ubuntu 22.04)
- **OS Version:** Ubuntu 22.04 LTS
- **Node Version:** v20.11.0
- **npm Version:** v10.9.2
- **Test Results:**
  - [ ] Clone successful
  - [ ] No Yarn files present
  - [ ] npm install completed (Time: ___ minutes)
  - [ ] npm run dev works
  - [ ] npm run build succeeds
  - [ ] npm start functions
  - [ ] No Yarn-related errors
- **Issues Encountered:** None / [Describe]
- **Screenshots:** [Attach if issues found]

### Overall Assessment
- **Migration Status:** ✅ Success / ❌ Issues Found
- **Recommendation:** Ready for Production / Requires Fixes
- **Next Steps:** [List any follow-up actions]
```

🚨 Failure Scenarios & Solutions

Scenario 1: “Cannot find module ‘xyz’“

Cause: Incomplete installation
Solution:

```
rm -rf node_modules package-lock.json
npm install --legacy-peer-deps
```

Scenario 2: “peer dep missing: react-hook-form@^7.55.0”

Cause: Forgot –legacy-peer-deps flag
Solution: Always use `npm install --legacy-peer-deps`

Scenario 3: TypeScript compilation errors

Cause: Type definitions missing or outdated
Solution:

```
npm install --save-dev @types/node @types/react @types/react-dom --legacy-peer-deps
```

Scenario 4: Port 3000 already in use

Cause: Another process using port
Solution:

```
# Find and kill process (Linux/Mac)
lsof -ti:3000 | xargs kill -9

# Or use different port
PORT=3001 npm run dev
```

Scenario 5: Build fails with “Dynamic server usage” errors

Cause: Expected behavior for API routes
Solution: These are warnings, not errors. Build continues successfully.

✅ Success Criteria

The migration is successful if ALL of the following are true:

1. **No Yarn Files:** Only `package-lock.json` exists (no `yarn.lock`)
2. **Clean Install:** `npm install --legacy-peer-deps` completes without errors
3. **Dev Server:** `npm run dev` starts and site loads in browser
4. **Production Build:** `npm run build` completes with 41/41 pages
5. **No Yarn Errors:** Zero Yarn-related error messages in any step
6. **Cross-Platform:** Works identically on Windows, Mac, and Linux
7. **Documentation:** README and DEVELOPER_HANDOVER reflect npm usage

---

📞 Support & Escalation

If you encounter issues during testing:

1. **Check Documentation:** Review `YARN_TO_NPM_MIGRATION.md`
2. **Clean Reinstall:** Delete `node_modules` and retry
3. **Verify Node/npm Versions:** Ensure using LTS versions
4. **Consult Team:** Share test report with development team
5. **GitHub Issues:** Open issue with detailed error logs

📝 Notes for Testers

- **Test Duration:** Allow 30-45 minutes per platform
- **Network:** Stable internet required for package downloads
- **Disk Space:** Ensure 2GB+ free space for node_modules
- **Screenshots:** Capture any errors for documentation
- **Document Everything:** Even successful tests should be logged

---

Created By: DeepAgent (Abacus.AI)
Last Updated: November 10, 2025
Migration Document: YARN_TO_NPM_MIGRATION.md