✅ YARN TO NPM MIGRATION - COMPLETE

Date: November 10, 2025
Status: ✅ PRODUCTION READY
Migration Type: Full Yarn → npm transition

🎯 MIGRATION SUMMARY

The QuantumLeap AI project has been **fully migrated from Yarn to npm**. All Yarn files, references, and workflows have been removed. The project is now **npm-first** and production-ready.

✅ VALIDATION RESULTS

1️⃣ File Audit

- ✅ **No yarn.lock files** in project root or nextjs_space
- ✅ **No .yarnrc files** found
- ✅ **package-lock.json** present and functional (682KB)
- ✅ **node_modules** contains only npm-managed dependencies

Note: Some dependencies in node_modules contain their own yarn.lock files. This is normal and does not affect our npm workflow.

2️⃣ Package CRUD Test

- ✅ **Install test:** `npm install lodash --legacy-peer-deps` → SUCCESS
- ✅ **Verification:** Package added to package.json and package-lock.json
- ✅ **Uninstall test:** `npm uninstall lodash --legacy-peer-deps` → SUCCESS
- ✅ **Lock file update:** package-lock.json correctly regenerated

3️⃣ Build Validation

```
npm run build
```

Result: ✅ SUCCESS
- All pages compiled successfully
- No Yarn-related errors
- Production build ready for deployment

4️⃣ IDE/Tooling Check

- ✅ No VSCode/WebStorm Yarn configurations found
- ✅ No workspace files with Yarn references
- ✅ All scripts use npm commands

5️⃣ Documentation Audit

- ✅ README.md updated with npm commands
- ✅ DEVELOPER_HANDOVER.md reflects npm workflow
- ✅ .gitignore includes Yarn blocking rules
- ✅ Pre-commit hook blocks new Yarn files

🚀 TEAM ONBOARDING

⚠️ IMPORTANT: YARN IS NOW DEPRECATED

Effective immediately, this project uses **npm only**. Yarn commands will not work and should not be used.

🔧 Setup Instructions for Team Members

1. Pull Latest Changes

```
git pull origin semi-final
```

2. Remove Old Dependencies

```
cd nextjs_space
rm -rf node_modules
rm -f yarn.lock  # If it exists
```

3. Install with npm

```
npm install --legacy-peer-deps
```

Why `--legacy-peer-deps`?
Some dependencies have peer dependency conflicts. This flag resolves them safely.

4. Update IDE Settings

VSCode:
1. Open Settings (Cmd/Ctrl + ,)
2. Search for "package manager"
3. Set default to `npm`
4. Reload window

WebStorm/IntelliJ:
1. Settings → Languages & Frameworks → Node.js
2. Package Manager → npm
3. Apply and restart

5. Verify Setup

```
npm run dev    # Should start dev server on localhost:3000
npm run build  # Should compile without errors
```

📝 DAILY WORKFLOW

✅ DO THIS:

```
npm install <package>             # Add dependency
npm install <package> --save-dev  # Add dev dependency
npm uninstall <package>           # Remove dependency
npm run dev                       # Start dev server
npm run build                     # Production build
npm run lint                      # Run linter
```

❌ DON’T DO THIS:

```
yarn install       # ❌ Will not work
yarn add <package> # ❌ Will not work
yarn remove        # ❌ Will not work
```

🛡️ SAFETY MECHANISMS

Pre-commit Hook

A Git pre-commit hook blocks any attempts to commit:
- `yarn.lock`
- `.yarnrc` files
- Yarn commands in scripts

Location: `.git/hooks/pre-commit`

.gitignore Protection

The following Yarn files are blocked from commits:
- `**/yarn.lock`
- `**/.yarnrc*`
- `**/.yarn/` directories

🔍 TROUBLESHOOTING

Issue: “Cannot find module”

Solution:

```
rm -rf node_modules
npm install --legacy-peer-deps
```

Issue: “Peer dependency conflict”

Solution: Always use `--legacy-peer-deps` flag

```
npm install --legacy-peer-deps
```

Issue: “Package-lock.json conflicts”

Solution:

```
git checkout package-lock.json
npm install --legacy-peer-deps
```

Issue: “Old Yarn commands in scripts”

Solution: Check package.json scripts section. All should use `npm run`.

📚 KEY DOCUMENTS

Document	Purpose
`README.md`	Quick start guide (npm commands)
`DEVELOPER_HANDOVER.md`	Complete project overview
`YARN_TO_NPM_MIGRATION.md`	Migration technical details
`CICD_NPM_INTEGRATION_GUIDE.md`	CI/CD pipeline setup
`PEER_DEPENDENCY_UPGRADE_ROADMAP.md`	Future dependency updates

🎯 NEXT STEPS

Immediate (Week 1)

- [ ] All team members pull latest changes
- [ ] All team members delete node_modules and reinstall with npm
- [ ] Update local IDE settings
- [ ] Test dev and build commands locally

Short-term (Week 2-4)

- [ ] Verify CI/CD pipelines use npm
- [ ] Check Vercel/deployment settings
- [ ] Monitor for Yarn-related errors
- [ ] Update any team documentation/wikis

Long-term (Month 2+)

- [ ] Plan peer dependency upgrades (see PEER_DEPENDENCY_UPGRADE_ROADMAP.md)
- [ ] Consider upgrading Node.js to latest LTS
- [ ] Review and update dependencies quarterly

📞 SUPPORT

Questions?

1. Check `DEVELOPER_HANDOVER.md` for workflow details
2. Review `YARN_TO_NPM_MIGRATION.md` for technical context
3. Ask in team Slack/Discord channel

Found a Yarn reference?

1. Report it immediately
2. Do not commit Yarn files
3. Update affected documentation

✅ FINAL CHECKLIST

Before starting work, verify:
- [ ] Latest code pulled from `semi-final` branch
- [ ] `node_modules` deleted and reinstalled with npm
- [ ] `package-lock.json` present (not yarn.lock)
- [ ] IDE configured for npm
- [ ] `npm run dev` works
- [ ] `npm run build` succeeds

🎉 MIGRATION COMPLETE

This project is now npm-first and production-ready.

All validation tests passed. No Yarn dependencies remain. Team can proceed with confidence.

Document Version: 1.0
Last Updated: November 10, 2025
Maintained By: Development Team