



BestieQuest Download Checklist

What to Download Before Leaving

Essential Files

1. Complete Application Code

- [] `/home/ubuntu/kids_story_app/` (entire folder)
- Download as ZIP from Files browser
- Contains both frontend and backend

2. Backend Files (`nodejs_space`)

- [] `nodejs_space/src/` - All source code
- [] `nodejs_space/prisma/` - Database schema
- [] `nodejs_space/package.json` - Dependencies
- [] `nodejs_space/tsconfig.json` - TypeScript config
- [] `nodejs_space/nest-cli.json` - NestJS config
- [] `nodejs_space/public/` - Pre-built frontend PWA

3. Frontend Files (`react_native_space`)

- [] `react_native_space/app/` - Screen components
- [] `react_native_space/components/` - UI components
- [] `react_native_space/services/` - API integration
- [] `react_native_space/context/` - State management
- [] `react_native_space/assets/` - Icons and images
- [] `react_native_space/app.json` - Expo configuration
- [] `react_native_space/package.json` - Dependencies
- [] `react_native_space/web-build/` - Pre-built PWA (ready to deploy!)

4. Assets

- [] `app-icon-source.png` - Original character image
- [] `react_native_space/assets/icon-192.png` - PWA icon
- [] `react_native_space/assets/icon-512.png` - PWA icon
- [] `react_native_space/assets/hero-image.png` - Homepage image
- [] `bestiequest-custom-domain-qr.png` - QR code

5. Documentation

- [] `MIGRATION-PLAYBOOK.md` - Full deployment guide
 - [] `QUICK-START-RAILWAY.md` - Railway quick deploy
 - [] `DOWNLOAD-CHECKLIST.md` - This file
 - [] `nodejs_space/README.md` - Backend docs
 - [] `react_native_space/README.md` - Frontend docs
-

Download Methods

Method 1: DeepAgent Files Browser (Easiest)

1. Click “Files” button (top-right in DeepAgent)
2. Navigate to `kids_story_app` folder
3. Right-click folder → “Download” or click download icon
4. Save ZIP to your computer
5. Extract the ZIP file

Method 2: Download Individual Files

1. Click “Files” button
2. Browse to specific files
3. Click download icon next to each file
4. Organize locally in same folder structure

Method 3: Create Archive First

If download from Files browser doesn't work:

```
# Create ZIP archive
cd /home/ubuntu
zip -r bestiequest-complete.zip kids_story_app/

# Now download bestiequest-complete.zip from Files browser
```

Verify Download

Check You Have Everything:

Folder structure should look like:

```
kids_story_app/
├── nodejs_space/
│   ├── src/
│   ├── prisma/
│   ├── public/ (contains pre-built PWA)
│   ├── package.json
│   └── ...
├── react_native_space/
│   ├── app/
│   ├── components/
│   ├── assets/
│   ├── web-build/ (pre-built PWA)
│   ├── package.json
│   ├── app.json
│   └── ...
└── MIGRATION-PLAYBOOK.md
    └── QUICK-START-RAILWAY.md
    └── DOWNLOAD-CHECKLIST.md
    └── bestiequest-custom-domain-qr.png
```

File count check:

- Backend source files: ~50+ TypeScript files
 - Frontend source files: ~30+ TypeScript/TSX files
 - Assets: 10+ image files
 - Config files: ~10+ JSON/config files
-



Important Information to Save

1. API Keys

Abacus AI API Key (for story generation):

- Current key is in your deployment
- Get new one from: <https://abacus.ai/app/route-llm-apis>
- Save it securely!

Database Connection:

- Current DATABASE_URL (if you want to keep existing stories)
- Can also start fresh on new platform

2. Configuration Details

App Details:

App Name: BestieQuest
 Domain: bestiequest.co.za
 Backend Port: 3000
 Database: PostgreSQL
 LLM: Grok 4.1 Fast Non-Reasoning (via Abacus AI RouteLLM)

Features:

- 50 characters/items
- 4 main themes (Funny, Magical, Learning, Life Lessons)
- 3 story lengths (800, 1200, 2000 words)
- Gender selection (Boy/Girl)
- Story library with save/favorite/delete
- PWA (Progressive Web App)
- Unified backend + frontend hosting

3. Deployment URLs

Current:

- Production: <https://bestiequest.co.za>
- API Docs: <https://bestiequest.co.za/api-docs>

Save these for reference!

Security Notes

DO NOT Include in Downloads:

- ✗ `.env` file with live API keys (create new on new platform)
- ✗ `node_modules/` folders (reinstall with `npm install`)
- ✗ `.git/` folder (if exists)
- ✗ Build artifacts (`dist/`, `.next/`, etc.) - rebuild on new platform
- ✗ Database dump with user data (unless you specifically want to migrate)

DO Include:

- ✓ All source code (`.ts`, `.tsx`, `.js`, `.jsx`)
 - ✓ Configuration files (`package.json`, `tsconfig.json`, etc.)
 - ✓ Assets (images, icons)
 - ✓ Database schema (`prisma/schema.prisma`)
 - ✓ Documentation
 - ✓ Pre-built PWA (`web-build/` folder)
-

Backup Strategy

Cloud Backup (Recommended):

1. GitHub (Free, Private Repo)

```
bash
cd kids_story_app
git init
git add .
git commit -m "BestieQuest complete app"
gh repo create bestiequest --private --source=. --push
```

2. Google Drive / Dropbox

- Upload ZIP file
- Always accessible
- Version history

3. Multiple Locations

- Computer local drive
 - External hard drive
 - Cloud storage
 - USB drive
-

Pre-Deployment Checklist

Before deploying elsewhere:

- [] Downloaded all files
- [] Verified folder structure
- [] Saved API keys

- [] Read MIGRATION-PLAYBOOK.md
 - [] Choose hosting platform
 - [] Have domain DNS access
 - [] Tested extracted files locally (optional)
-



Quick Test Locally (Optional)

To verify your download works:

```
# Backend
cd kids_story_app/nodejs_space
npm install
npm run build
# Setup local PostgreSQL and .env with DATABASE_URL
npx prisma migrate deploy
npm start
# Visit http://localhost:3000

# Frontend (if you want to modify)
cd kids_story_app/react_native_space
npm install
npx expo export --platform web --output-dir web-build
# Copy web-build to nodejs_space/public/
```



You're Ready!

Once you've downloaded everything:

1. Extract ZIP
2. Verify files
3. Read MIGRATION-PLAYBOOK.md
4. Choose platform (Railway recommended)
5. Follow QUICK-START-RAILWAY.md
6. Deploy!
7. Update DNS
8. Test app
9. Install PWA
10. Enjoy! 🎉

Your BestieQuest app is completely portable and ready to deploy anywhere! ❤️✨