🚀 BestieQuest Migration Playbook

Overview

This guide will help you deploy BestieQuest on any platform of your choice. Your app is fully portable and works on all major hosting providers.

📦 What You Have

Backend (Node.js/NestJS)

- Location: `/nodejs_space/`
- Framework: NestJS (TypeScript)
- Database: PostgreSQL with Prisma ORM
- API Documentation: Swagger at `/api-docs`
- Port: 3000

Frontend (React Native PWA)

- Location: `/react_native_space/`
- Framework: Expo (React Native Web)
- Build output: `/react_native_space/web-build/`
- Already compiled and ready to serve as static files

Architecture

- **Unified hosting**: Backend serves frontend as static files
- **Single domain**: Both frontend and backend on same URL
- **Progressive Web App**: Installable on iOS/Android

📥 Step 1: Download Your Code

Option A: Download via DeepAgent Files Browser

1. Click **Files** button (top-right)
2. Navigate to `kids_story_app` folder
3. Download the entire folder as ZIP

Option B: Download Specific Parts

Backend Code:
- `nodejs_space/` - Complete backend
- `nodejs_space/src/` - Source code
- `nodejs_space/prisma/` - Database schema
- `nodejs_space/package.json` - Dependencies
- `nodejs_space/.env` - Environment variables (create new on new platform)

Frontend Code:
- `react_native_space/` - Source code
- `react_native_space/web-build/` - Pre-built PWA (ready to deploy!)
- `react_native_space/app.json` - Expo configuration

Assets:
- `react_native_space/assets/` - Icons, images
- All icons with cute characters already generated

Option C: Git Repository (Recommended)

If you want version control:

```
cd kids_story_app
git init
git add .
git commit -m "Initial commit - BestieQuest app"
# Push to GitHub, GitLab, or Bitbucket
```

🌐 Step 2: Choose Your Hosting Platform

Best Options for Your App:

Option 1: Vercel (Recommended for simplicity)

- ✅ Free tier available
- ✅ Automatic deployments
- ✅ Custom domain support
- ✅ SSL included
- ✅ Serverless functions for backend
- 💰 Cost: Free for hobby, $20/mo Pro

Option 2: Railway

- ✅ Full Node.js support
- ✅ PostgreSQL database included
- ✅ Simple deployment
- ✅ Custom domains
- 💰 Cost: $5/month minimum

Option 3: Render

- ✅ Full Node.js + PostgreSQL
- ✅ Free tier available
- ✅ Auto-deploy from Git
- ✅ Custom domains
- 💰 Cost: Free tier, $7/mo for paid

Option 4: Heroku

- ✅ Mature platform
- ✅ PostgreSQL addon
- ✅ Custom domains

- 💰 Cost: ~$7-13/month

Option 5: DigitalOcean App Platform

- ✅ Full control
- ✅ Managed database
- ✅ Simple deployment
- 💰 Cost: $5-12/month

Option 6: Self-hosted (VPS)

- ✅ Full control
- ✅ Cheapest long-term
- ❌ More technical setup
- 💰 Cost: $4-6/month (Hetzner, DigitalOcean Droplet)

🔧 Step 3: Platform-Specific Deployment

Deploy to Vercel

Requirements:
- Node.js backend needs to be adapted to serverless functions OR
- Deploy backend elsewhere (Railway/Render) and frontend on Vercel

Steps:

```
# Install Vercel CLI
npm i -g vercel

# Deploy backend (needs adaptation)
cd nodejs_space
vercel --prod

# Deploy frontend
cd react_native_space/web-build
vercel --prod
```

Note: Vercel is best for static frontend. Use Railway/Render for backend.

Deploy to Railway (Easiest Full-Stack)

Steps:

1. **Create Railway Account**
   - Go to: https://railway.app/
   - Sign up with GitHub

2. **Create New Project**
   - Click “New Project”
   - Select “Deploy from GitHub repo” OR “Empty Project”

3. **Add PostgreSQL Database**
   - Click “+ New”

   - Select “Database” → “PostgreSQL”
   - Railway provides DATABASE_URL automatically

4. **Deploy Backend**
   - Upload your `nodejs_space` folder
   - Railway auto-detects Node.js
   - Set environment variables:
   ```
   DATABASE_URL=<auto-provided by Railway>
       ABACUSAI_API_KEY=<your key>
       NODE_ENV=production
       PORT=3000
   ```
   - Railway will run `npm install` and `npm run build` automatically
   - Start command: `node dist/main.js`

5. **Configure Domain**
   - Railway provides default domain: `your-app.railway.app`
   - Add custom domain: `bestiequest.co.za`
   - Update DNS: CNAME to Railway domain

6. **Database Migration**
   ```
   bash
     npx prisma migrate deploy
   ```

Railway `railway.json` config:

```
{
  "$schema": "https://railway.app/railway.schema.json",
  "build": {
    "builder": "NIXPACKS",
    "buildCommand": "npm install && npm run build"
  },
  "deploy": {
    "startCommand": "node dist/main.js",
    "restartPolicyType": "ON_FAILURE",
    "restartPolicyMaxRetries": 10
  }
}
```

Deploy to Render

Steps:

1. **Create Render Account**
   - Go to: https://render.com/
   - Sign up

2. **Create PostgreSQL Database**
   - New → PostgreSQL
   - Note the Internal Database URL

3. **Create Web Service**
   - New → Web Service

- Connect your Git repo or upload files
- Settings:

  - **Name:** bestiequest
  - **Environment:** Node
  - **Build Command:** `cd nodejs_space && npm install && npm run build`
  - **Start Command:** `cd nodejs_space && node dist/main.js`
  - **Instance Type:** Free or Starter ($7/mo)

4. **Environment Variables**

```
DATABASE_URL=<from Render PostgreSQL>
   ABACUSAI_API_KEY=<your key>
   NODE_ENV=production
   PORT=3000
```

5. **Custom Domain**
   - Settings → Custom Domain
   - Add `bestiequest.co.za`
   - Update DNS CNAME

6. **Deploy**
   - Render auto-deploys on Git push
   - Or manual deploy from dashboard

Deploy to Heroku

Steps:

1. **Install Heroku CLI**

```
bash
   npm install -g heroku
   heroku login
```

2. **Create App**

```
bash
   cd kids_story_app/nodejs_space
   heroku create bestiequest
```

3. **Add PostgreSQL**

```
bash
   heroku addons:create heroku-postgresql:mini
```

4. **Set Environment Variables**

```
bash
   heroku config:set ABACUSAI_API_KEY=your_key
   heroku config:set NODE_ENV=production
```

5. **Create Procfile**

```
web: node dist/main.js
```

6. **Deploy**

```
bash
```

```
   git init
   git add .
   git commit -m "Deploy to Heroku"
   heroku git:remote -a bestiequest
   git push heroku main
```

7. **Run Migrations**
   `bash`
   ```
      heroku run npx prisma migrate deploy
   ```

8. **Add Custom Domain**
   `bash`
   ```
      heroku domains:add bestiequest.co.za
   ```

Self-Hosted (VPS - Ubuntu/Debian)

Requirements:
- VPS with Ubuntu 22.04+ (Hetzner, DigitalOcean, Vultr, etc.)
- SSH access
- Domain pointing to server IP

Steps:

1. **Connect to VPS**
   `bash`
   ```
      ssh root@your-server-ip
   ```

2. **Install Dependencies**
   ```bash
   # Update system
   apt update && apt upgrade -y

# Install Node.js 20
curl -fsSL https://deb.nodesource.com/setup_20.x | bash -
apt install -y nodejs

# Install PostgreSQL
apt install -y postgresql postgresql-contrib

# Install Nginx
apt install -y nginx

# Install PM2 (process manager)
npm install -g pm2
```

1. **Setup PostgreSQL**
   `bash`
   ```
      sudo -u postgres psql
      CREATE DATABASE bestiequest;
      CREATE USER bestiequest WITH PASSWORD 'your_secure_password';
   ```

```
   GRANT ALL PRIVILEGES ON DATABASE bestiequest TO bestiequest;
   \q
```

2. **Upload Your Code**

```bash
# On your computer:
scp -r kids_story_app root@your-server-ip:/var/www/

# On server:
cd /var/www/kids_story_app/nodejs_space
npm install
npm run build
```

1. **Setup Environment Variables**

```
bash
   cd /var/www/kids_story_app/nodejs_space
   nano .env
```

Add:

```
DATABASE_URL="postgresql://bestiequest:your_secure_password@localhost:5432/bestiequest"
   ABACUSAI_API_KEY=your_key
   NODE_ENV=production
   PORT=3000
```

1. **Run Migrations**

```
bash
   npx prisma migrate deploy
```

2. **Start with PM2**

```
bash
   pm2 start dist/main.js --name bestiequest
   pm2 startup
   pm2 save
```

3. **Configure Nginx**

```
bash
   nano /etc/nginx/sites-available/bestiequest
```

Add:

```nginx
server {
listen 80;
server_name bestiequest.co.za www.bestiequest.co.za;
```

```
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
```

~~~
}
```
~~~

Enable:

```
bash
  ln -s /etc/nginx/sites-available/bestiequest /etc/nginx/sites-enabled/
  nginx -t
  systemctl reload nginx
```

1. **Setup SSL with Let's Encrypt**

   ```
   bash
     apt install -y certbot python3-certbot-nginx
     certbot --nginx -d bestiequest.co.za -d www.bestiequest.co.za
   ```

2. **Setup Auto-Start**
   PM2 already configured to start on boot!

🗄️ Database Migration

Export Data from Current Database

```
# Get current DATABASE_URL from Abacus deployment
# Run on your computer:
cd kids_story_app/nodejs_space
npx prisma db pull
npx prisma db push --preview-feature
```

Import to New Database

```
# On new platform:
cd nodejs_space
npx prisma migrate deploy
```

Manual Data Export (if needed)

```
# Export stories from current database:
pg_dump -h current_host -U current_user -d current_db -t stories > stories.sql

# Import to new database:
psql -h new_host -U new_user -d new_db < stories.sql
```

🔐 Environment Variables Setup

Required Variables

Create `.env` file in `nodejs_space/`:

```
# Database
DATABASE_URL="postgresql://user:password@host:5432/database"

# LLM API (for story generation)
ABACUSAI_API_KEY="your_abacus_api_key"

# Application
NODE_ENV="production"
PORT="3000"

# Optional: If backend and frontend are on different domains
APP_ORIGIN="https://bestiequest.co.za/"
```

Get Your Abacus API Key

Your story generation uses Abacus AI's RouteLLM API (Grok 4.1).

To keep using it:
1. Go to: https://abacus.ai/app/route-llm-apis
2. Get your API key
3. Add to environment variables

Alternative: Use OpenAI Instead

Edit `nodejs_space/src/stories/stories.service.ts`:

```
// Change from:
const response = await fetch('https://default.abacus.ai/v1/chat/completions', {
  headers: {
    'Authorization': `Bearer ${process.env.ABACUSAI_API_KEY}`,
  },
  // ...
});

// To:
const response = await fetch('https://api.openai.com/v1/chat/completions', {
  headers: {
    'Authorization': `Bearer ${process.env.OPENAI_API_KEY}`,
  },
  body: JSON.stringify({
    model: 'gpt-4',  // or 'gpt-3.5-turbo' for cheaper
    // ... rest stays same
  })
});
```

Add to `.env`:

```
OPENAI_API_KEY="sk-..."
```

🌐 DNS Configuration

Point Your Domain to New Platform

For Railway/Render/Heroku:

At your domain registrar (where you bought bestiequest.co.za):

1. Go to DNS settings
2. Delete old records
3. Add new CNAME:
   `Type: CNAME`
   `   Name: @  (or leave blank for root)`
   `   Value: your-app.railway.app  (or render.com URL, etc.)`
   `   TTL: 3600`
4. Add www subdomain:
   `Type: CNAME`
   `   Name: www`
   `   Value: your-app.railway.app`
   `   TTL: 3600`

For Self-Hosted VPS:

Add A records:

```
Type: A
Name: @
Value: your.server.ip.address
TTL: 3600

Type: A
Name: www
Value: your.server.ip.address
TTL: 3600
```

Wait: DNS propagation takes 5 minutes to 48 hours

Test:

```
nslookup bestiequest.co.za
```

📱 Update Frontend API URL

If Backend on Different Domain

Edit `react_native_space/utils/constants.ts`:

```
export const API_URL = 'https://bestiequest-api.railway.app';
// Or wherever your backend is deployed
```

Rebuild frontend:

```
cd react_native_space
npx expo export --platform web --output-dir web-build
```

Copy to backend:

```
cp -r web-build/* ../nodejs_space/public/
```

✅ Post-Deployment Checklist

Test Everything:

- [ ] Visit `https://bestiequest.co.za`
- [ ] Homepage loads with hero image
- [ ] Click “Create New Story”
- [ ] Select 3-4 characters
- [ ] Choose theme
- [ ] Pick story length
- [ ] Select gender
- [ ] Generate story (should work with LLM API)

- [ ] Story displays properly
- [ ] Save story to library
- [ ] Favorite a story
- [ ] Delete a story
- [ ] Install PWA on phone
- [ ] Check icon shows cute characters
- [ ] Test API docs at `/api-docs`

Performance:

- [ ] Check response times (should be <1 second)
- [ ] Monitor database connections
- [ ] Check error logs
- [ ] Monitor memory usage

🔄 Keep-Warm Strategy (Optional)

If Your New Platform Has Cold Starts:

Many platforms (Render free tier, Railway free tier) also sleep containers.

Solutions:

Option 1: UptimeRobot (Free)

1. Go to: https://uptimerobot.com/
2. Create monitor:
   - Type: HTTP(s)
   - URL: `https://bestiequest.co.za/health`
   - Interval: 5 minutes
3. UptimeRobot pings your app every 5 minutes
4. Keeps container warm!

Option 2: Cron-Job.org (Free)

1. Go to: https://cron-job.org/
2. Create job:
   - URL: `https://bestiequest.co.za/health`
   - Interval: Every 15 minutes

Option 3: GitHub Actions (Free)

Create `.github/workflows/keep-warm.yml`:

```
name: Keep Warm
on:
  schedule:
    - cron: '*/15 * * * *'  # Every 15 minutes
workflow_dispatch:

jobs:
  ping:
    runs-on: ubuntu-latest
    steps:
      - name: Ping endpoint
        run: curl https://bestiequest.co.za/health
```

💰 Cost Comparison

Monthly Costs:

Platform	Free Tier	Paid Tier	Database	SSL	Custom Domain
Railway	$5 credit	$5/mo min	Included	Free	Free
Render	Yes (sleeps)	$7/mo	$7/mo extra	Free	Free
Heroku	No	$7-13/mo	$5/mo	Free	Free
DigitalOcean	No	$12/mo	Included	Free	Free
VPS (Hetzner)	No	$4-6/mo	Included	Free	Free
Vercel	Yes	$20/mo	Separate	Free	Free

Recommendation: Railway or Render for simplicity, VPS for cheapest long-term.

🆘 Troubleshooting

"Cannot connect to database"

- Check DATABASE_URL format
- Verify database is running
- Check firewall rules
- Test connection: `psql $DATABASE_URL`

"Story generation fails"

- Check ABACUSAI_API_KEY is set
- Verify API key is valid

- Check API endpoint URL
- Monitor API rate limits

“PWA not installing”

- Check manifest.json is accessible
- Verify icons exist
- Check HTTPS is working
- Clear browser cache

“Slow response times”

- Check server resources
- Monitor database query times
- Add indexes to database
- Consider CDN for static assets

📞 Support

Your App Stack:

- **Backend:** NestJS + TypeScript
- **Frontend:** React Native + Expo
- **Database:** PostgreSQL + Prisma
- **LLM:** Abacus AI RouteLLM (Grok 4.1)

Community Resources:

- NestJS Docs: https://docs.nestjs.com/
- Expo Docs: https://docs.expo.dev/
- Prisma Docs: https://www.prisma.io/docs/
- Railway Docs: https://docs.railway.app/
- Render Docs: https://render.com/docs/

🎉 You’re Ready!

Your BestieQuest app is completely portable and can run anywhere Node.js is supported!

Choose your platform, follow the steps, and you’ll be live in 30-60 minutes.

Good luck! 🚀💜