Vercel Deployment Setup Guide for RishiPrajnya

✅ Repository Status

Your GitHub repository has been successfully restructured for Vercel deployment!

GitHub Repository: https://github.com/Victordtesla24/rishi-prajnya
Branch: `main`
Structure: ✅ Next.js app at root level (Vercel-ready)

🚀 Deploying to Vercel

Step 1: Access Vercel Dashboard

1. Go to https://vercel.com/dashboard (https://vercel.com/dashboard)
2. Sign in with your account

Step 2: Import Your GitHub Repository

1. Click **"Add New Project"**
2. Select **"Import Git Repository"**
3. Find and select: **`Victordtesla24/rishi-prajnya`**
4. Click **"Import"**

Step 3: Configure Build Settings

On the configuration page, use these exact settings:

Project Settings:

- **Project Name**: `rishi-prajnya` (or your preferred name)
- **Framework Preset**: `Next.js` (should auto-detect)
- **Root Directory**: `./` (leave as root - DO NOT change)

Build Settings:

- **Build Command**: `yarn build` (auto-detected)
- **Output Directory**: `.next` (auto-detected)
- **Install Command**: `yarn install` (auto-detected)

Node.js Version:

- **Node.js Version**: `18.x` or higher (recommended: 20.x)

Step 4: Add Environment Variables

CRITICAL: Before deploying, add your environment variables:

1. Scroll down to **"Environment Variables"** section
2. Add the following variables:

```
Name: DATABASE_URL
Value: postgresql://role_139b68a72a:GKCmPkTG1l1BVZlJjwFCLcR1RlvAt3LN@db-139b68a72a.db002.hosteddb.reai.io:5432/139b68a72a?connect_timeout=15
Environments: Production, Preview, Development (select all)
```

```
Name: NODE_ENV
Value: production
Environments: Production only
```

1. Click **"Add"** after each variable

Step 5: Deploy

1. Review all settings
2. Click **"Deploy"**
3. Wait for the build to complete (usually 2-5 minutes)

🔍 Troubleshooting Common Issues

Issue 1: Build Fails with "Module not found"

Solution:
- Check that all dependencies are in `package.json`
- Vercel installs dependencies from scratch, so local symlinks don't matter
- Ensure `yarn.lock` is committed to repository

Issue 2: 404 Error on Deployment

Solution:
- Verify **Root Directory** is set to `./` (root)
- Do NOT point to `nextjs_space` subdirectory
- Check that `app` directory is at repository root

Issue 3: Environment Variables Not Working

Solution:
- Ensure variables are added in Vercel Dashboard → Settings → Environment Variables
- Redeploy after adding variables (existing deployments don't auto-update)
- For client-side access, prefix with `NEXT_PUBLIC_`

Issue 4: Build Timeout

Solution:
- Default timeout is 10 minutes on Free tier
- Optimize build by removing unused dependencies
- Consider upgrading to Pro tier for longer timeouts

📋 Vercel Deployment Checklist

Before deploying, verify:

- [x] GitHub repository structure is correct (Next.js files at root)
- [x] `package.json` exists at root
- [x] `next.config.js` exists at root
- [x] `vercel.json` exists at root
- [x] `app` directory exists at root
- [x] All code is pushed to GitHub `main` branch
- [ ] Environment variables are configured in Vercel
- [ ] Root Directory is set to `./` (default)
- [ ] Build command is `yarn build`
- [ ] Deploy button is clicked

🎯 Expected Deployment Flow

1. **Connect Repository**: Vercel links to your GitHub repo
2. **Install Dependencies**: Vercel runs `yarn install`
3. **Build Application**: Vercel runs `yarn build`
4. **Deploy**: Application is deployed to Vercel CDN
5. **Success**: You get a live URL (e.g., `rishi-prajnya.vercel.app`)

Build Output Should Show:

```
Installing dependencies...
✓ Dependencies installed

Building application...
✓ Compiled successfully
✓ Linting and checking validity of types
✓ Collecting page data
✓ Generating static pages
✓ Finalizing page optimization

Route (app)                              Size     First Load JS
┌ ƒ /                                    134 kB          221 kB
├ ƒ /_not-found                          877 B          88.3 kB
└ ƒ /api/contact                         0 B                0 B

Deployment successful!
```

🔄 Automatic Deployments

Once configured, Vercel will automatically deploy:

- **Production**: Every push to `main` branch
- **Preview**: Every push to other branches

- **Pull Requests**: Automatic preview deployments with unique URLs

---

🌐 Custom Domain (Optional)

To add a custom domain:

1. Go to Project Settings → Domains
2. Enter your domain name
3. Follow DNS configuration instructions
4. Wait for SSL certificate to be issued (automatic)

📊 Post-Deployment Verification

After deployment succeeds:

1. **Visit Deployment URL**: Click the provided Vercel URL
2. **Check Homepage**: Verify hero section and navigation load correctly
3. **Test Navigation**: Click through all sections
4. **Test Contact Form**: Submit a test message
5. **Check Mobile**: Test responsive design on mobile devices
6. **Browser Console**: Ensure no errors in developer console

---

🆘 Getting Help

If deployment fails:

1. **Check Build Logs**:
   - Go to Vercel Dashboard → Deployments
   - Click on failed deployment
   - Review build logs for errors

2. **Common Fixes**:
   - Clear Vercel build cache and retry
   - Verify environment variables are set
   - Check Node.js version compatibility
   - Ensure all imports use correct paths (case-sensitive)

3. **Vercel Support**:
   - Visit: https://vercel.com/help
   - Check: https://vercel.com/docs/errors
   - Community: https://github.com/vercel/vercel/discussions

✨ Next Steps After Successful Deployment

1. **Test thoroughly**: Verify all features work in production
2. **Setup Analytics**: Enable Vercel Analytics for insights
3. **Configure Monitoring**: Setup error tracking
4. **Setup CI/CD**: Automatic deployments are already configured
5. **Share URL**: Send deployment URL to stakeholders

📝 Important Notes

- Vercel installs dependencies from scratch (symlinks won’t affect deployment)
- Build happens on Vercel’s infrastructure (local build issues don’t matter)
- Environment variables must be configured in Vercel Dashboard
- SSL certificates are automatic and free
- Deployments are cached for faster subsequent builds

Repository is ready for Vercel deployment! Follow the steps above to deploy. 🚀