

RampAI Website Deployment Guide

Table of Contents

1. [Overview](#)
 2. [Pre-Deployment Checklist](#)
 3. [Domain Setup for ramp-ai.co.in](#)
 4. [Deployment Options](#)
 5. [Environment Configuration](#)
 6. [Post-Deployment Steps](#)
 7. [Troubleshooting](#)
-

Overview

This guide provides comprehensive instructions for deploying the RampAI website to production, specifically targeting the domain `ramp-ai.co.in`. The website is built with Next.js 14 and can be deployed on multiple platforms.

Website Features:

- Next.js 14 with App Router
 - Responsive design with Tailwind CSS
 - Motion animations with Framer Motion
 - Optimized images with Next.js Image component
 - Professional project showcase
 - Training programs catalog
 - Contact forms and consultation booking
-

Pre-Deployment Checklist

Code Requirements

- ☐ All images are displaying correctly
- ☐ All BrewAI references changed to RampAI
- ☐ Timeline section updated with correct milestones
- ☐ Build process completes without errors
- ☐ All pages load correctly in production mode

Domain Requirements

- ☐ Domain `ramp-ai.co.in` is registered and active
- ☐ Access to domain registrar control panel
- ☐ DNS management capabilities confirmed

Technical Requirements

- ☐ Node.js 18+ installed

- [] Project builds successfully (`npm run build`)
- [] All dependencies properly installed

Domain Setup for ramp-ai.co.in

Step 1: Verify Domain Ownership

1. Confirm you have administrative access to `ramp-ai.co.in`
2. Log into your domain registrar (where you purchased the domain)
3. Navigate to DNS management section

Step 2: Nameserver Configuration

Choose one of the following options based on your hosting provider:

Option A: Using Vercel Nameservers

```
ns1.vercel-dns.com
ns2.vercel-dns.com
```

Option B: Using Netlify Nameservers

```
dns1.p01.nsone.net
dns2.p01.nsone.net
dns3.p01.nsone.net
dns4.p01.nsone.net
```

Option C: Using Cloudflare Nameservers

```
nameserver1.cloudflare.com
nameserver2.cloudflare.com
```

Step 3: DNS Records Setup

If you're not using your hosting provider's nameservers, configure these DNS records:

A Records

```
Type: A
Name: @
Value: [Your hosting provider's IP address]
TTL: 300

Type: A
Name: www
Value: [Your hosting provider's IP address]
TTL: 300
```

CNAME Records (Alternative)

```
Type: CNAME
Name: www
Value: [Your hosting provider's domain]
TTL: 300
```

Deployment Options

Option 1: Vercel Deployment (Recommended)

Step 1: Create Vercel Account

1. Go to vercel.com (<https://vercel.com>)
2. Sign up with GitHub account
3. Install Vercel CLI: `npm i -g vercel`

Step 2: Deploy from Local Machine

```
# Navigate to project directory
cd /home/ubuntu/brewai-website/app

# Login to Vercel
vercel login

# Deploy
vercel

# Follow prompts:
# - Set up and deploy? Y
# - Which scope? [Your username]
# - Link to existing project? N
# - Project name: ramp-ai-website
# - Directory: ./
# - Override settings? N
```

Step 3: Configure Custom Domain

1. Go to Vercel dashboard
2. Select your project
3. Go to Settings → Domains
4. Add domain: `ramp-ai.co.in`
5. Add domain: `www.ramp-ai.co.in`
6. Configure DNS as prompted

Step 4: Environment Variables

```
# In Vercel dashboard, go to Settings → Environment Variables
# Add any required environment variables
```

Option 2: Netlify Deployment

Step 1: Create Netlify Account

1. Go to netlify.com (<https://netlify.com>)
2. Sign up with GitHub account
3. Install Netlify CLI: `npm install -g netlify-cli`

Step 2: Build and Deploy

```
# Navigate to project directory
cd /home/ubuntu/brewai-website/app

# Build the project
npm run build

# Login to Netlify
netlify login

# Deploy
netlify deploy

# For production deployment
netlify deploy --prod
```

Step 3: Configure Custom Domain

1. Go to Netlify dashboard
2. Select your site
3. Go to Domain settings
4. Add custom domain: `ramp-ai.co.in`
5. Configure DNS as prompted

Option 3: Traditional VPS/Server Deployment

Step 1: Server Setup

```
# Connect to your server
ssh user@your-server-ip

# Install Node.js 18+
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs

# Install PM2 for process management
npm install -g pm2

# Install Nginx
sudo apt update
sudo apt install nginx
```

Step 2: Deploy Application

```
# Clone or upload your project
git clone [your-repo] /var/www/ramp-ai
# OR upload files to /var/www/ramp-ai

# Navigate to project
cd /var/www/ramp-ai

# Install dependencies
npm install

# Build project
npm run build

# Start with PM2
pm2 start npm --name "ramp-ai" -- start
pm2 save
pm2 startup
```

Step 3: Configure Nginx

```
# Create /etc/nginx/sites-available/ramp-ai.co.in
server {
    listen 80;
    server_name ramp-ai.co.in www.ramp-ai.co.in;

    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_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;
        proxy_cache_bypass $http_upgrade;
    }
}
```

```
# Enable site
sudo ln -s /etc/nginx/sites-available/ramp-ai.co.in /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

Step 4: SSL Certificate

```
# Install Certbot
sudo apt install certbot python3-certbot-nginx

# Get SSL certificate
sudo certbot --nginx -d ramp-ai.co.in -d www.ramp-ai.co.in
```

Environment Configuration

Required Environment Variables

```
# .env.local
NEXTAUTH_URL=https://ramp-ai.co.in
NEXTAUTH_SECRET=your-secret-key-here

# For analytics (if needed)
GOOGLE_ANALYTICS_ID=your-ga-id

# For contact forms (if needed)
SMTP_HOST=your-smtp-host
SMTP_PORT=587
SMTP_USER=your-email
SMTP_PASS=your-password
```

Build Configuration

Ensure your `next.config.js` is optimized for production:

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: 'standalone', // For Docker deployments
  images: {
    remotePatterns: [
      {
        protocol: 'https',
        hostname: '**',
      },
    ],
  },
  compress: true,
  generateEtags: false,
  poweredByHeader: false,
}

module.exports = nextConfig
```

Post-Deployment Steps

1. Verify Deployment

- [] Website loads at `https://ramp-ai.co.in`
- [] Website loads at `https://www.ramp-ai.co.in`
- [] All pages are accessible
- [] Images load correctly
- [] Navigation works properly
- [] Contact forms function (if applicable)

2. Performance Optimization

```
# Test website performance
# Use tools like:
# - Google PageSpeed Insights
# - GTmetrix
# - WebPageTest
```

3. SEO Setup

- ☐ Google Search Console verification
- ☐ Submit sitemap to Google
- ☐ Set up Google Analytics (if needed)
- ☐ Configure robots.txt

4. Monitoring Setup

- ☐ Set up uptime monitoring
- ☐ Configure error tracking
- ☐ Set up performance monitoring

Troubleshooting

Common Issues and Solutions

Issue: Images not loading

Solution:

1. Check image paths in the code
2. Verify Next.js Image component configuration
3. Check Content Security Policy settings

Issue: 404 errors on page refresh

Solution:

1. Configure server to serve index.html for all routes
2. For Nginx: add `try_files $uri $uri/ /index.html;`
3. For Apache: ensure `.htaccess` is configured

Issue: Build failures

Solution:

1. Check Node.js version compatibility
2. Clear `node_modules` and `package-lock.json`
3. Run `npm install` and `npm run build`

Issue: Slow loading times

Solution:

1. Enable compression (gzip/brotli)
2. Optimize images
3. Configure CDN
4. Enable caching headers

Issue: SSL certificate problems

Solution:

1. Verify domain ownership
2. Check DNS propagation
3. Renew certificates if expired
4. Configure proper redirects

Support Contacts

For deployment support:

- **Technical Issues:** Contact your hosting provider
- **Domain Issues:** Contact your domain registrar
- **Application Issues:** Review this guide or contact development team

Quick Reference Commands

Vercel

```
vercel --prod          # Deploy to production
vercel domains         # Manage domains
vercel env             # Manage environment variables
vercel logs            # View deployment logs
```

Netlify

```
netlify deploy --prod  # Deploy to production
netlify open           # Open site in browser
netlify status         # Check deployment status
netlify logs           # View deployment logs
```

Traditional Server

```
pm2 status             # Check application status
pm2 restart ramp-ai    # Restart application
pm2 logs ramp-ai       # View application logs
sudo systemctl status nginx # Check Nginx status
```

Conclusion

This guide provides multiple deployment options for the RampAI website. Choose the option that best fits your technical requirements and infrastructure preferences. Vercel is recommended for its simplicity and Next.js optimization, while traditional server deployment offers more control and customization options.

For additional support or questions about deployment, refer to the official documentation of your chosen hosting platform.

Deployment Date: [Current Date]

Website: ramp-ai.co.in

Framework: Next.js 14

Status: Ready for Production