



Vercel Deployment Guide for AWIBI MEDTECH Frontend



Prerequisites

- GitHub account with AWIBI MEDTECH repository
- Vercel account (free tier is sufficient)
- Backend deployed on Render (get the URL first)



Step-by-Step Deployment

Step 1: Prepare Your Repository

1. **Ensure your code is pushed to GitHub** `bash git add . git commit -m "Prepare for Vercel deployment" git push origin main`

2. **Verify project structure** `awibi-medtech/ ├── frontend/awibi-medtech-frontend/ | ├── src/ | ├── public/ | ├── package.json | ├── vite.config.js | ├── .env | ├── .env.production └── backend/`

Step 2: Deploy to Vercel

1. **Go to Vercel Dashboard**
2. Visit <https://vercel.com>
3. Sign in with your GitHub account
4. **Import Project**
5. Click "New Project"
6. Select "Import Git Repository"

7. Choose your AWIBI MEDTECH repository

8. Configure Project Settings

9. **Project Name:** `awibi-medtech` (or your preferred name)

10. **Framework Preset:** Vite

11. **Root Directory:** `frontend/awibi-medtech-frontend`

12. **Build Command:** `npm run build`

13. **Output Directory:** `dist`

14. **Install Command:** `npm install`

Step 3: Set Environment Variables

In the Vercel dashboard, go to your project → Settings → Environment Variables:

```
# Production Environment Variables
VITE_API_URL=https://your-backend-url.onrender.com
VITE_GOOGLE_CLIENT_ID=your-google-oauth-client-id
VITE_APP_NAME=AWIBI MEDTECH
VITE_APP_VERSION=1.0.0
VITE_APP_ENV=production
```

Important Notes: - Replace `your-backend-url.onrender.com` with your actual Render backend URL - Replace `your-google-oauth-client-id` with your actual Google OAuth client ID - Make sure there's no trailing slash in `VITE_API_URL`

Step 4: Deploy

1. **Click "Deploy"**

2. Vercel will automatically build and deploy your application

3. Wait for the build to complete (usually 2-3 minutes)

4. **Get Your Deployment URL**

5. Example: `https://awibi-medtech.vercel.app`

6. Note this URL for backend CORS configuration

Step 5: Update Backend CORS

1. **Go to your Render backend dashboard**
2. **Update environment variables:** `env FRONTEND_URL=https://awibi-medtech.vercel.app ALLOWED_ORIGINS=https://awibi-medtech.vercel.app`
3. **Redeploy your backend service**

Step 6: Test Your Deployment

1. **Visit your Vercel URL**
2. **Test key functionality:**
 3. Homepage loads correctly
 4. Navigation works
 5. Search functionality works
 6. Login/Register forms work
 7. API calls succeed (check browser console)

Advanced Configuration

Custom Domain Setup

1. **In Vercel Dashboard**
2. Go to Project → Settings → Domains
3. Add your custom domain
4. Follow DNS configuration instructions
5. **Update Backend CORS** `env FRONTEND_URL=https://your-custom-domain.com ALLOWED_ORIGINS=https://your-custom-domain.com,https://awibi-medtech.vercel.app CUSTOM_DOMAINS=your-custom-domain.com`

Preview Deployments

Vercel automatically creates preview deployments for: - Pull requests - Non-production branches

These will have URLs like: - `https://awibi-medtech-git-feature-branch.vercel.app`

Build Optimization

1. **Add build optimization to package.json:**

```
json { "scripts": { "build": "vite build --mode production", "preview": "vite preview" } }
```
2. **Configure Vite for production:**

```
javascript // vite.config.js export default defineConfig({ plugins: [react()], build: { outDir: 'dist', sourcemap: false, minify: 'terser', rollupOptions: { output: { manualChunks: { vendor: ['react', 'react-dom'], router: ['react-router-dom'], ui: ['@radix-ui/react-dialog', '@radix-ui/react-dropdown-menu'] } } } })
```



Troubleshooting

Build Failures

1. **Check build logs in Vercel dashboard**
2. **Common issues:**
3. Missing dependencies: `npm install`
4. TypeScript errors: Fix type issues
5. Environment variables: Ensure all required vars are set

CORS Issues

1. **Check browser console for CORS errors**
2. **Verify backend CORS configuration**

3. **Test API endpoints directly:** `bash curl -H "Origin: https://your-app.vercel.app" \ https://your-backend.onrender.com/api/test-cors`

Performance Issues

1. **Enable Vercel Analytics**
2. Go to Project → Analytics
3. Monitor Core Web Vitals
4. **Optimize images:** `javascript // Use Vercel Image Optimization import Image from 'next/image' // If using Next.js // Or optimize images before deployment`

Environment Variable Issues

1. **Check variable names** (must start with `VITE_`)
2. **Verify values** (no quotes needed in Vercel dashboard)
3. **Redeploy after changes**



Monitoring and Maintenance

Vercel Analytics

1. **Enable Analytics**
2. Go to Project → Analytics
3. Monitor page views, performance, and errors
4. **Set up alerts**
5. Configure notifications for build failures
6. Monitor performance metrics

Automatic Deployments

1. **Connected to GitHub**

2. Pushes to main branch trigger deployments
3. Pull requests create preview deployments
4. **Deployment Protection**
5. Enable deployment protection for production
6. Require manual approval for sensitive changes

Performance Monitoring

1. **Core Web Vitals**
2. Monitor LCP (Largest Contentful Paint)
3. Track FID (First Input Delay)
4. Measure CLS (Cumulative Layout Shift)
5. **Bundle Analysis** `bash npm run build -- --analyze`



Security Best Practices

Environment Variables

1. **Never commit sensitive data**
2. **Use different values for development/production**
3. **Rotate secrets regularly**

HTTPS

1. **Vercel provides HTTPS by default**
2. **Ensure all API calls use HTTPS**
3. **Set secure headers**

Content Security Policy

```
// Add to index.html or configure in Vercel
<meta http-equiv="Content-Security-Policy"
      content="default-src 'self';
              script-src 'self' 'unsafe-inline';
              style-src 'self' 'unsafe-inline';
              img-src 'self' data: https; ">
```

Deployment Checklist

- ☐ Repository is up to date on GitHub
- ☐ Backend is deployed and accessible
- ☐ Environment variables are configured
- ☐ Build completes successfully
- ☐ Application loads without errors
- ☐ API calls work correctly
- ☐ Authentication functions properly
- ☐ All pages are accessible
- ☐ Mobile responsiveness is verified
- ☐ Performance is acceptable
- ☐ CORS is configured correctly
- ☐ Custom domain is set up (if applicable)
- ☐ Analytics are enabled
- ☐ Monitoring is in place

Support

If you encounter issues:

1. **Check Vercel documentation:** <https://vercel.com/docs>
2. **Review build logs** in Vercel dashboard
3. **Test locally** before deploying

4. **Check GitHub issues** for similar problems

5. **Contact Vercel support** if needed

Your AWIBI MEDTECH frontend should now be successfully deployed on Vercel! 🎉