

# Complete Deployment Guide for MERN MySQL App

This guide will walk you through deploying your MERN MySQL application to a Hostinger VPS with automatic GitHub integration.

## Table of Contents

1. [Local Development Setup](#)
  2. [VPS Initial Setup](#)
  3. [GitHub Repository Setup](#)
  4. [VPS Application Deployment](#)
  5. [Automatic Deployment with GitHub Actions](#)
  6. [Testing and Verification](#)
  7. [Troubleshooting](#)
- 

## Local Development Setup

### Step 1: Install Prerequisites

- Node.js (v18+): <https://nodejs.org/>
- MySQL (v8+): <https://dev.mysql.com/downloads/>
- Git: <https://git-scm.com/>

### Step 2: Create Local MySQL Database

```
sql  
-- Open MySQL Command Line or MySQL Workbench  
CREATE DATABASE mern_app_db;
```

### Step 3: Setup Backend Locally

```
bash
```

```
cd backend  
npm install  
  
# Create .env file from example  
cp .env.example .env  
  
# Edit .env with your local MySQL credentials  
# DB_HOST=localhost  
# DB_USER=root  
# DB_PASSWORD=your_local_password  
# DB_NAME=mern_app_db  
  
# Start backend  
npm run dev
```

Backend runs on: <http://localhost:5000>

#### Step 4: Setup Frontend Locally

```
bash  
  
cd frontend  
npm install  
  
# Create .env file from example  
cp .env.example .env  
  
# Start frontend  
npm run dev
```

Frontend runs on: <http://localhost:5173>

#### Step 5: Test Locally

1. Visit <http://localhost:5173>
2. Register a new user
3. Login with credentials
4. View users list

---

## VPS Initial Setup

#### Step 1: Connect to Your VPS

```
bash
```

```
ssh root@your_vps_ip
```

## Step 2: Run Initial Setup Script

```
bash

# Upload setup-vps.sh to your VPS
chmod +x setup-vps.sh
./setup-vps.sh
```

Or manually install:

```
bash

# Update system
apt update && apt upgrade -y

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

# Install MySQL
apt install mysql-server -y

# Secure MySQL
mysql_secure_installation

# Install Nginx
apt install nginx -y

# Install PM2
npm install -g pm2

# Install Git
apt install git -y
```

## Step 3: Configure MySQL on VPS

```
bash

mysql -u root -p
```

```
sql
```

```
CREATE DATABASE mern_app_db;
CREATE USER 'mern_user'@'localhost' IDENTIFIED BY 'strong_secure_password';
GRANT ALL PRIVILEGES ON mern_app_db.* TO 'mern_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

## Step 4: Configure Firewall

```
bash

ufw allow OpenSSH
ufw allow 'Nginx Full'
ufw enable
ufw status
```

## 📦 GitHub Repository Setup

### Step 1: Create GitHub Repository

1. Go to <https://github.com/new>
2. Create a new repository (e.g., "mern-mysql-app")
3. Don't initialize with README (we already have one)

### Step 2: Push Code to GitHub

```
bash

# On your local machine, in project root
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/yourusername/mern-mysql-app.git
git push -u origin main
```

### Step 3: Generate SSH Key for GitHub Actions

On your local machine:

```
bash

ssh-keygen -t rsa -b 4096 -C "github-actions-deploy" -f ~/.ssh/vps_deploy_key
```

This creates two files:

- `~/.ssh/vps_deploy_key` (private key)
- `~/.ssh/vps_deploy_key.pub` (public key)

## Step 4: Add Public Key to VPS

bash

```
# Copy the public key
cat ~/.ssh/vps_deploy_key.pub

# On VPS, add it to authorized_keys
echo "paste_your_public_key_here" >> ~/.ssh/authorized_keys
```

## Step 5: Add Secrets to GitHub

1. Go to your GitHub repository
2. Click Settings → Secrets and variables → Actions
3. Click "New repository secret"
4. Add these secrets:

### VPS\_HOST

```
your_vps_ip_address
```

### VPS\_USERNAME

```
root
```

### VPS\_SSH\_KEY

```
# Copy and paste the entire private key
cat ~/.ssh/vps_deploy_key
# Copy everything including -----BEGIN and -----END lines
```

## 🚀 VPS Application Deployment

### Step 1: Clone Repository on VPS

bash

```
cd /var/www  
git clone https://github.com/yourusername/mern-mysql-app.git  
cd mern-mysql-app
```

## Step 2: Setup Backend

```
bash  
  
cd backend  
npm install --production  
  
# Create .env file  
nano .env
```

Add production environment variables:

```
env  
  
NODE_ENV=production  
PORT=5000  
DB_HOST=localhost  
DB_USER=mern_user  
DB_PASSWORD=strong_secure_password  
DB_NAME=mern_app_db  
DB_PORT=3306  
JWT_SECRET=super_secret_production_key_change_this  
JWT_EXPIRE=7d  
CORS_ORIGIN=https://yourdomain.com  
API_VERSION=v1
```

Save with **(Ctrl+X)**, then **(Y)**, then **(Enter)**

## Step 3: Start Backend with PM2

```
bash  
  
pm2 start server.js --name mern-backend  
pm2 save  
pm2 startup  
# Copy and run the command that PM2 shows
```

## Step 4: Build Frontend

```
bash
```

```
cd /var/www/mern-mysql-app/frontend
```

```
# Create .env file
```

```
nano .env
```

Add:

```
env
```

```
VITE_API_URL=https://yourdomain.com/api/v1
```

Build:

```
bash
```

```
npm install
```

```
npm run build
```

## Step 5: Configure Nginx

```
bash
```

```
# Remove default config
```

```
rm /etc/nginx/sites-enabled/default
```

```
# Create new config
```

```
nano /etc/nginx/sites-available/mern-app
```

Paste this configuration (replace [yourdomain.com] with your actual domain):

```
nginx
```

```

server {
  listen 80;
  server_name yourdomain.com www.yourdomain.com;

  location / {
    root /var/www/mern-mysql-app/frontend/dist;
    try_files $uri $uri/ /index.html;
  }

  location /api {
    proxy_pass http://localhost:5000;
    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;
  }
}

```

Enable and restart:

```

bash

ln -s /etc/nginx/sites-available/mern-app /etc/nginx/sites-enabled/
nginx -t
systemctl restart nginx

```

## Step 6: Setup SSL (Optional but Recommended)

```

bash

apt install certbot python3-certbot-nginx -y
certbot --nginx -d yourdomain.com -d www.yourdomain.com

```

Follow the prompts. Certbot will automatically update your Nginx config.

## Automatic Deployment with GitHub Actions

The `.github/workflows/deploy.yml` file is already configured. Every time you push to the `main` branch, GitHub Actions will:

1. Connect to your VPS via SSH
2. Pull latest code

3. Install dependencies
4. Restart backend (PM2)
5. Build frontend
6. Restart Nginx

## Test Automatic Deployment

Make a change in your local code:

```
bash

# Edit something, like frontend/src/pages/Home.jsx
git add .
git commit -m "Test automatic deployment"
git push origin main
```

Watch the deployment:

1. Go to GitHub → Your Repository → Actions
2. See the deployment running
3. Wait for green checkmark
4. Visit your site to see changes

---

## ✓ Testing and Verification

### Check Backend

```
bash

# On VPS
pm2 status
pm2 logs mern-backend

# Test API
curl http://localhost:5000/health
```

### Check Frontend

Visit your domain in browser: <https://yourdomain.com>

### Check Database Connection

```
bash
```

```
mysql -u mern_user -p mern_app_db
```

```
# Inside MySQL
SHOW TABLES;
SELECT * FROM users;
EXIT;
```

## Monitor Logs

```
bash

# Backend logs
pm2 logs mern-backend

# Nginx logs
tail -f /var/log/nginx/access.log
tail -f /var/log/nginx/error.log
```

## Troubleshooting

### Backend won't start

```
bash

# Check logs
pm2 logs mern-backend

# Check if port 5000 is available
netstat -tulpn | grep 5000

# Restart
pm2 restart mern-backend
```

### Frontend shows blank page

```
bash
```

```
# Check Nginx config
nginx -t

# Check if files exist
ls -la /var/www/mern-mysql-app/frontend/dist

# Rebuild frontend
cd /var/www/mern-mysql-app/frontend
npm run build
```

## Database connection errors

```
bash

# Test MySQL connection
mysql -u mern_user -p

# Check backend .env file
cat /var/www/mern-mysql-app/backend/.env

# Verify MySQL is running
systemctl status mysql
```

## GitHub Actions failing

1. Check secrets are correctly set in GitHub
2. Verify SSH key is added to VPS
3. Check deployment logs in GitHub Actions tab
4. Ensure project path exists: `/var/www/mern-mysql-app`

## CORS errors

Check backend `.env`:

```
env

CORS_ORIGIN=https://yourdomain.com
```

And frontend `.env`:

```
env

VITE_API_URL=https://yourdomain.com/api/v1
```

## ⌚ Quick Commands Reference

### On VPS

```
bash

# Deploy manually
cd /var/www/mern-mysql-app && git pull && cd backend && npm install && pm2 restart mern-backend && cd ../frontend

# Or use the deploy script
./deploy.sh

# Monitor
pm2 monit

# Check status
pm2 status
systemctl status nginx
systemctl status mysql

# View logs
pm2 logs mern-backend
tail -f /var/log/nginx/error.log
```

### On Local Machine

```
bash

# Development
cd backend && npm run dev
cd frontend && npm run dev

# Deploy to production
git add .
git commit -m "Your message"
git push origin main
```



Your MERN MySQL app is now:

- Running locally for development
- Deployed on VPS for production
- Automatically deploys on GitHub push

- Using the same MySQL database locally and remotely (configured via .env)
- Secured with SSL (if configured)

Any changes you make locally and push to GitHub will automatically reflect on your live site!

---

## Support

If you encounter issues:

1. Check the troubleshooting section
2. Review logs on VPS
3. Verify all environment variables
4. Ensure all services are running

Happy coding! 