Production Deployment Checklist

Use this checklist to ensure your YourStyle backend is properly configured for production deployment.

Pre-Deployment Security

Environment Variables

- [ ] Changed `DB_PASSWORD` from default value
- [ ] Changed `JWT_SECRET` to random 32+ character string
- [ ] Changed `PGADMIN_PASSWORD` from default value
- [ ] Set `NODE_ENV=production`
- [ ] Verified `CORS_ORIGIN` includes only your production domains
- [ ] Removed any test/debug values from `.env`

Database Security

- [ ] PostgreSQL uses strong password
- [ ] PostgreSQL port (5432) not exposed to public internet
- [ ] Database backups configured (daily recommended)
- [ ] pgAdmin not accessible from public internet (or password protected)

Application Security

- [ ] All default credentials changed
- [ ] `.env` file not committed to git
- [ ] `.gitignore` properly configured
- [ ] No sensitive data in code or logs
- [ ] CORS properly restricted to your domains only

Infrastructure Setup

Server Requirements

- [ ] Windows Server or Linux VPS with adequate resources
- [ ] Minimum: 1 CPU, 2GB RAM, 20GB storage
- [ ] Recommended: 2 CPU, 4GB RAM, 50GB storage
- [ ] Static IP address or domain configured
- [ ] Firewall configured (allow ports: 80, 443, deny 3000, 5432, 5050 from public)
- [ ] SSH/RDP access secured

Docker Setup

- [ ] Docker Desktop (Windows) or Docker Engine (Linux) installed
- [ ] Docker Compose installed
- [ ] Docker containers set to auto-restart
- [ ] Docker volumes backed up regularly

Node.js Setup

- [ ] Node.js LTS version installed
- [ ] npm or yarn installed
- [ ] Dependencies installed (`npm install`)

Database Configuration

Initial Setup

- [ ] PostgreSQL container running
- [ ] Database migrations executed successfully (`npm run migrate`)
- [ ] Products seeded successfully (`npm run seed`)
- [ ] Database connection tested from backend
- [ ] pgAdmin accessible (for management)

Backup Strategy

- [ ] Automated daily backups configured
- [ ] Backup location secured (off-server)
- [ ] Backup restoration tested
- [ ] Backup retention policy defined (e.g., keep 30 days)

Backend Server Configuration

Application Setup

- [ ] Backend code deployed to server
- [ ] `.env` file configured with production values
- [ ] Server starts without errors (`npm start`)
- [ ] Health endpoint responds: `http://localhost:3000/health`

Process Management (PM2 Recommended)

- [ ] PM2 installed globally: `npm install -g pm2`
- [ ] Backend started with PM2: `pm2 start server.js --name yourstyle-backend`
- [ ] PM2 configured to start on system boot: `pm2 startup` and `pm2 save`
- [ ] PM2 log rotation configured

Logging

- [ ] Application logs being written
- [ ] Log rotation configured (prevent disk fill)
- [ ] Error logs monitored
- [ ] Access logs reviewed regularly

Apache/Web Server Configuration

Apache Setup (Windows/XAMPP)

- [ ] Apache installed and running
- [ ] Proxy modules enabled (mod_proxy, mod_proxy_http, mod_headers)
- [ ] `apache-reverse-proxy.conf` copied to Apache config directory
- [ ] Configuration included in `httpd.conf`
- [ ] Apache restarted successfully
- [ ] Proxy working: `http://yourdomain.com/api/health` responds

Static Frontend

- [ ] Frontend files deployed to DocumentRoot
- [ ] Frontend can access backend via `/api` endpoints
- [ ] CORS not blocking requests

SSL/TLS Configuration

Certificate Setup

- [ ] SSL certificate obtained (Let’s Encrypt recommended)
- [ ] Certificate files copied to Apache ssl directories:
- [ ] Certificate: `ssl.crt/yourstyle.crt`
- [ ] Private key: `ssl.key/yourstyle.key`
- [ ] Chain: `ssl.crt/yourstyle-chain.crt`
- [ ] SSL modules enabled in Apache
- [ ] HTTPS VirtualHost configured (port 443)
- [ ] HTTP redirects to HTTPS
- [ ] Apache restarted

Certificate Renewal

- [ ] Certbot auto-renewal configured
- [ ] Renewal tested: `certbot renew --dry-run`
- [ ] Renewal scheduled (every 60 days)
- [ ] Post-renewal hook configured (copy certs, restart Apache)

Testing & Validation

API Endpoints Testing

- [ ] Health check: `GET /health` returns 200 OK
- [ ] Register user: `POST /api/auth/register` works
- [ ] Login: `POST /api/auth/login` returns JWT token
- [ ] Products: `GET /api/products` returns products
- [ ] Search: `GET /api/search?q=test` works
- [ ] Cart (with auth): `GET /api/cart` requires token

- [ ] Add to cart: `POST /api/cart` works with valid token
- [ ] Create order: `POST /api/orders` works with items in cart
- [ ] Order history: `GET /api/orders` returns user orders

Security Testing

- [ ] Endpoints without token return 401 Unauthorized
- [ ] Invalid credentials return 401
- [ ] SQL injection attempts blocked (parameterized queries)
- [ ] XSS attempts sanitized
- [ ] CORS blocks unauthorized domains
- [ ] Passwords properly hashed in database
- [ ] JWT tokens expire correctly

Performance Testing

- [ ] Load test with expected traffic volume
- [ ] Database queries optimized (use indexes)
- [ ] Response times acceptable (< 500ms for API calls)
- [ ] Concurrent user handling tested
- [ ] Memory leaks checked (monitor over 24 hours)

Monitoring & Maintenance

Monitoring Setup

- [ ] Server resource monitoring (CPU, RAM, disk)
- [ ] Application uptime monitoring
- [ ] Database health monitoring
- [ ] Error rate monitoring
- [ ] Response time monitoring
- [ ] Alert system configured (email/SMS for critical issues)

Regular Maintenance Tasks

- [ ] Weekly: Review logs for errors
- [ ] Weekly: Check disk space
- [ ] Weekly: Verify backups are working
- [ ] Monthly: Update dependencies (test first)
- [ ] Monthly: Review security updates
- [ ] Quarterly: Review and rotate logs
- [ ] Quarterly: Performance audit

Backup Procedures

- [ ] Database backup script created
- [ ] Backup schedule configured (daily recommended)
- [ ] Backup storage location secured
- [ ] Backup retention policy implemented
- [ ] Restore procedure documented and tested

Documentation

Internal Documentation

- [ ] Server access credentials documented (secured)
- [ ] Environment variables documented
- [ ] Deployment process documented
- [ ] Rollback procedure documented
- [ ] Emergency contacts documented

Team Knowledge

- [ ] Setup guide shared with team (SETUP.md)
- [ ] API documentation shared (README.md)
- [ ] Troubleshooting guide accessible
- [ ] On-call procedures defined

Performance Optimization

Database Optimization

- [ ] Indexes created for frequently queried columns
- [ ] Query performance analyzed
- [ ] Connection pooling configured
- [ ] Slow query logging enabled

Application Optimization

- [ ] Compression enabled (gzip)
- [ ] Static assets cached
- [ ] Rate limiting implemented (recommended)
- [ ] Request/response size minimized

Disaster Recovery

Backup Strategy

- [ ] Database backup automated
- [ ] Application code backed up (git repository)
- [ ] Configuration files backed up
- [ ] SSL certificates backed up
- [ ] Backup restoration tested

Rollback Plan

- [ ] Previous version tagged in git
- [ ] Rollback procedure documented
- [ ] Database rollback strategy defined

- [ ] Downtime communication plan

Compliance & Legal

Data Protection

- [ ] User data encrypted at rest (database encryption)
- [ ] User data encrypted in transit (HTTPS/SSL)
- [ ] Password policy enforced (min 6 characters)
- [ ] Data retention policy defined
- [ ] GDPR compliance reviewed (if applicable)

Privacy

- [ ] Privacy policy created
- [ ] Terms of service created
- [ ] Cookie policy defined (if using cookies)
- [ ] User consent mechanisms in place

Go-Live Checklist

Final Verifications

- [ ] All above items checked and verified
- [ ] Staging environment tested successfully
- [ ] Production database seeded with initial data
- [ ] DNS records pointing to production server
- [ ] SSL certificate valid and working
- [ ] All team members notified of go-live
- [ ] Support channels ready (email, phone, etc.)
- [ ] Monitoring dashboards active

Post-Launch

- [ ] Monitor error logs for first 24 hours
- [ ] Verify user registrations working
- [ ] Verify order processing working
- [ ] Check payment gateway (if integrated)
- [ ] Monitor server resources
- [ ] Collect initial user feedback

Emergency Contacts

System Administrator:
- Name: __

- Phone: __
- Email: ____

Database Administrator:
- Name: __
- Phone: __
- Email: ____

Developer On-Call:
- Name: __
- Phone: __
- Email: ____

Hosting Provider Support:
- Phone: __
- Email: __
- Portal: ____

Notes

Deployment Date: ____

Deployed By: ____

Server Details:
- IP Address: __
- Domain: __
- OS: ____

Additional Notes:

Sign-Off

By signing below, you confirm that all critical items have been reviewed and the system is ready for production deployment.

Technical Lead: ____ Date: ______

Project Manager: ____ Date: ______

Security Officer: ____ Date: ______

Document Version: 1.0.0

Last Updated: October 2025