## Ticketing System Deployment Document

**Version:** 1.0  
**Date:** 2025-09-17  
**Prepared by:** Tharun and Suffiyan

### 1. Introduction

**Purpose:** This document explains the deployment process and configuration details of the Ticketing System.

**Scope:** Covers pre-deployment preparation, production deployment, verification, and security considerations.

### 2. Pre-Deployment Preparation

#### 2.1 Environment Configuration

* Checked .env files for client (React) and server (Node.js).
* Updated production variables: API keys, ports, and database credentials.
* Verified CORS configuration on backend to allow frontend requests.

#### 2.2 Database Setup (Hostinger MySQL)

* Created MySQL database using Hostinger phpMyAdmin.
* Created database user with appropriate privileges.
* Imported existing SQL schema and data files.
* Added DB credentials to .env file:

DB\_HOST=<Hostinger-hostname>  
DB\_USER=<username>  
DB\_PASS=<password>  
DB\_NAME=<database\_name>

* Tested connection locally to ensure backend could connect.

#### 2.3 Scripts and Build Preparation

* Updated package.json scripts for production:
  + React: npm run build → static files
  + Node: npm start → backend server
* Ensured PM2 compatibility for Ubuntu deployment.

#### 2.4 Build & Test

* Built React frontend using npm run build.
* Tested build locally for API connectivity, CORS, and functionality.
* Verified Node.js backend runs correctly with production .env settings.

#### 2.5 Version Control

* Committed final changes to Git.
* Pushed production-ready code to repository.
* Tagged release or marked branch for deployment.

### 3. Production Deployment

#### 3.1 Server Environment

* Ubuntu server with Nginx & PM2 installed.
* Verified Node.js runtime and packages are up-to-date.

#### 3.2 Application Deployment

* Cloned repository to server directory.
* Installed dependencies using npm install.
* Built React frontend (npm run build) for production.
* Configured .env with Hostinger MySQL credentials and production variables.

#### 3.3 Nginx Configuration

* Configured server block in /etc/nginx/sites-available/.
* Set reverse proxy to Node.js backend.
* Installed SSL certificate via Let’s Encrypt.
* Verified configuration with nginx -t and restarted Nginx.

#### 3.4 PM2 Process Management

* Started Node.js app using:

pm start --name "ticketing-app"

* Configured PM2 startup scripts for auto-restart on reboot.
* Monitored logs using pm2 monit.

#### 3.5 Deployment Verification & Testing

* Verified application accessibility via domain over HTTPS.
* Tested workflows: login, ticket creation/update, database connectivity.
* Confirmed PM2 and Nginx logs show stable operation.

### 4. Security & Reliability

* SSL enabled for encrypted communication.
* PM2 ensures high availability and auto-recovery.
* Nginx handles traffic efficiently and securely.
* Environment variables and credentials secured on server.

### 5. Final Outcome

* Ticketing System successfully deployed and fully functional.
* Nginx serves as secure reverse proxy with HTTPS.
* PM2 manages Node.js backend with auto-restart and monitoring.
* Hostinger MySQL database created, imported, and connected via .env.
* Pre-deployment checks ensured smooth and error-free deployment.