Deployment Documentation – roster1.sigvitas.com

# 1. Domain & Hosting

• Domain: sigvitas.com (purchased earlier, not by you).

• Subdomain: roster1.sigvitas.com (used for deployment).

• VPS: Hostinger VPS purchased and configured for deployment.

• Hosting: Self-managed VPS where both frontend and backend are deployed.

# 2. SSL Certificate

• SSL Certificate purchased from Sectigo (Wildcard for \*.sigvitas.com).

• CSR (Certificate Signing Request) and Private Key were generated on the server.

• CA Bundle (Certificate Authority Bundle) contains intermediate certificates that establish trust between the SSL certificate and root authorities.

• The final certificate chain was created by combining the domain certificate and the CA bundle.

# 3. VPS Setup

• VPS Provider: Hostinger

• Operating System: Ubuntu (default setup by Hostinger).

• Accessed VPS using SSH:

Command: ssh root@srv697522  
What: Securely logs into the VPS.  
Why: Needed to execute server-level commands.  
Result: Gives a terminal session inside the VPS.

# 4. Frontend & Backend Deployment

• Frontend was uploaded using FileZilla into the VPS hosting directory.

• Backend was deployed on the VPS and exposed via Nginx reverse proxy.

• Nginx handled both static frontend and proxied backend requests.

# 5. Deployment Flow (Step by Step)

* Command: ssh root@srv697522

What: Logs into the VPS using SSH.

Why: To get administrative access to the server.

Result: Terminal session inside VPS.

* Command: cd /etc/nginx/sites-available

What: Navigate to Nginx config directory.

Why: To create/edit site-specific server block.

Result: Now inside Nginx config path.

* Command: nano /etc/nginx/sites-available/roster1.sigvitas.com

What: Open editor to create server block config.

Why: To configure domain, SSL, and reverse proxy.

Result: Configuration file edited and saved.

* Command: ln -s /etc/nginx/sites-available/roster1.sigvitas.com /etc/nginx/sites-enabled/

What: Create symbolic link for site config.

Why: Enable site in Nginx.

Result: Site is now active in Nginx.

* Command: nginx -t

What: Test Nginx configuration.

Why: Ensure no syntax errors before reloading.

Result: Displays success message if config is valid.

* Command: systemctl reload nginx

What: Reload Nginx service.

Why: Apply new changes without downtime.

Result: Nginx reloads with new configuration.

* Command: cat domain.crt ca\_bundle.crt > fullchain.pem

What: Combine domain certificate and CA bundle.

Why: Create a full chain certificate.

Result: Generates fullchain.pem file.

* Command: openssl s\_client -connect roster1.sigvitas.com:443 -servername roster1.sigvitas.com

What: Verify SSL chain and connectivity.

Why: Check if SSL is trusted and complete.

Result: Shows certificate chain, issuer, and errors (if any).

* Command: systemctl restart nginx

What: Restart Nginx service.

Why: Final restart to apply SSL settings.

Result: Website runs with HTTPS.

# 6. Final Notes

• Website is successfully accessible via https://roster1.sigvitas.com

• SSL certificate applied and valid in browsers.

• Browser clients trust the certificate even if local VPS shows verification code 21 (outdated CA store issue).

• Nginx reverse proxy ensures both frontend and backend work seamlessly.