

Deploy VPS NextJS

```
sudo mysql -u root -p
mysql -u drs_user -p -h 127.0.0.1 -P 3306 drs_derma
DATABASE_URL="mysql://drs_user:your_new_password@127.0.0.1:3306/drs_derma"
```

Run these commands (replace your_new_password with your desired password):

```
DROP USER IF EXISTS 'drs_user'@'localhost';
ALTER USER 'drs_user'@'localhost' IDENTIFIED WITH mysql_native_password BY
'your_new_password';
GRANT ALL PRIVILEGES ON drs_derma.* TO 'drs_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

```
SHOW DATABASES;
npx prisma generate
npx prisma db push
```

```
sudo apt remove libnode-dev nodejs -y
sudo apt clean
sudo apt autoremove -y
curl -fsSL https://deb.nodesource.com/setup\_22.x | sudo -E bash -
sudo apt install nodejs -y
sudo apt install curl -y
```

 **Fix: Use pm2 instead (recommended for production)**

Install pm2 globally:

```
bash
CopyEdit
npm install -g pm2
```

Start the app:

bash

CopyEdit

pm2 start npm --name "nextjs-app" -- start

Make sure pm2 restarts app on server reboot:

bash

CopyEdit

pm2 save

pm2 startup

Now your app will run in the background, and survive reboots or terminal closures.

✓ **Case 2: You're already using pm2**

If you started your app with pm2, you're good — you can safely:

Close the terminal

Close SSH session

Reboot the VPS — app will stay running

Check your app with:

bash

CopyEdit

pm2 list

✓ **VPS Deployment Summary for drsdermabd.com**

1. VPS Setup & SSH

Connected to Namecheap VPS via SSH (ssh root@your_ip).

Installed **Node.js 22 LTS** and **npm** by removing old versions first:

bash

CopyEdit

sudo apt remove nodejs libnode-dev

curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -

sudo apt install -y nodejs

2. Clone & Run Next.js App

Cloned your [Next.js](#) project from GitHub.

Installed dependencies:

```
bash
CopyEdit
npm install
```

Ran the app (for testing):

```
bash
CopyEdit
npm run build
npm start
```

(Optional: You may later set up PM2 or a production server like next start with reverse proxy.)

3. Static HTML Site Deployment

You uploaded your static landing page (index.html, assets/, etc.) into:

```
bash
CopyEdit
/var/www/drs-derma
```

4. Nginx Installation & Configuration

Installed and enabled **Nginx**.

Created config file:

```
bash
CopyEdit
sudo nano /etc/nginx/sites-available/drsdermabd.com
```

With content:

```
nginx
CopyEdit
server {
    listen 80;
    server_name drsdermabd.com www.drsdermabd.com;
```

```

root /var/www/drs-derma;
index index.html;
location / {
    try_files $uri $uri/ =404;
}
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
}
}

```

Enabled site and restarted Nginx:

```

bash
CopyEdit
sudo ln -s /etc/nginx/sites-available/drsdermabd.com /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

```

5. Domain Configuration

In Namecheap DNS:

Added A records pointing @ and www to your VPS IP.

Waited for DNS propagation (~5–15 minutes).

6. SSL Certificate

You installed **SSL from hosting panel** (not Let's Encrypt).

Configured Nginx to use the .crt and .key files:

```

nginx
CopyEdit
server {
    listen 80;
    server_name drsdermabd.com www.drsdermabd.com;
    return 301 https://$host$request_uri;
}

```

```

server {
    listen 443 ssl;
    server_name drsdermabd.com www.drsdermabd.com;

    ssl_certificate /etc/ssl/certs/drsdermabd.com.crt;
    ssl_certificate_key /etc/ssl/private/drsdermabd.com.key;

    root /var/www/drs-derma;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    }
}

```

Reloaded Nginx:

```

bash
CopyEdit
sudo nginx -t
sudo systemctl reload nginx

```

Final Result:

Your **HTML landing page is live at** <https://drsdermabd.com> with SSL.

VPS is properly configured to serve static content and future dynamic apps.