OUC ACCESS SYSTEM - STAGING SERVER MIGRATION INSTRUCTIONS

==========================================================

STEP 1: SERVER PREPARATION

--------------------------

Update the server:

sudo apt update && sudo apt upgrade -y

Install Node.js (use same version as development):

curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash -

sudo apt-get install -y nodejs

Install PM2 for process management:

sudo npm install -g pm2

Install Git:

sudo apt install git -y

STEP 2: CLONE YOUR REPOSITORY

------------------------------

Clone from GitHub:

git clone https://github.com/your-username/your-repo-name.git

cd your-repo-name

Install dependencies:

npm install

STEP 3: ENVIRONMENT CONFIGURATION

----------------------------------

Create environment file:

nano .env.local

Add these variables (update with your staging values):

MYSQL\_HOST=your-staging-db-host

MYSQL\_USER=your-staging-db-user

MYSQL\_PASSWORD=your-staging-db-password

MYSQL\_DATABASE=your-staging-db-name

NEXTAUTH\_URL=https://your-staging-domain.com

NEXTAUTH\_SECRET=your-staging-secret

GOOGLE\_CLIENT\_ID=your-staging-google-client-id

GOOGLE\_CLIENT\_SECRET=your-staging-google-client-secret

SMTP\_HOST=smtp.gmail.com

SMTP\_PORT=587

SMTP\_USER=your-smtp-user

SMTP\_PASS=your-smtp-password

STEP 4: DATABASE SETUP

----------------------

Install MySQL:

sudo apt install mysql-server -y

Secure MySQL:

sudo mysql\_secure\_installation

Create staging database:

mysql -u root -p

Run these SQL commands:

CREATE DATABASE your\_staging\_database;

CREATE USER 'staging\_user'@'localhost' IDENTIFIED BY 'strong\_password';

GRANT ALL PRIVILEGES ON your\_staging\_database.\* TO 'staging\_user'@'localhost';

FLUSH PRIVILEGES;

EXIT;

Import development database:

mysqldump -u dev\_user -p dev\_database > staging\_dump.sql

mysql -u staging\_user -p your\_staging\_database < staging\_dump.sql

STEP 5: BUILD THE APPLICATION

-----------------------------

Build for production:

npm run build

Test the build:

npm start

STEP 6: WEB SERVER SETUP (NGINX)

---------------------------------

Install Nginx:

sudo apt install nginx -y

Create Nginx configuration:

sudo nano /etc/nginx/sites-available/ouc-staging

Add this content to the file:

server {

listen 80;

server\_name your-staging-domain.com;

location / {

proxy\_pass http://localhost:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_cache\_bypass $http\_upgrade;

}

}

Enable the site:

sudo ln -s /etc/nginx/sites-available/ouc-staging /etc/nginx/sites-enabled/

sudo nginx -t

sudo systemctl restart nginx

STEP 7: SSL CERTIFICATE (LET'S ENCRYPT)

----------------------------------------

Install Certbot:

sudo apt install snapd

sudo snap install core; sudo snap refresh core

sudo snap install --classic certbot

Create certificate:

sudo certbot --nginx -d your-staging-domain.com

STEP 8: PROCESS MANAGEMENT WITH PM2

------------------------------------

Create PM2 ecosystem file:

nano ecosystem.config.js

Add this content:

module.exports = {

apps: [{

name: 'ouc-staging',

script: 'npm',

args: 'start',

instances: 1,

autorestart: true,

watch: false,

max\_memory\_restart: '1G',

env: {

NODE\_ENV: 'production',

PORT: 3000

}

}]

};

Start with PM2:

pm2 start ecosystem.config.js

pm2 save

pm2 startup

STEP 9: DEPLOYMENT SCRIPT FOR UPDATES

--------------------------------------

Create deployment script:

nano deploy-staging.sh

Add this content:

#!/bin/bash

echo "Deploying to staging..."

git pull origin main

npm install

npm run build

pm2 restart ouc-staging

echo "Staging deployment complete!"

Make script executable:

chmod +x deploy-staging.sh

STEP 10: VERIFICATION CHECKLIST

--------------------------------

Check these items:

- Application loads at staging URL

- Database connections work

- Authentication (Google & email) works

- Email notifications work

- All forms and features function

- Images and assets load correctly

- SSL certificate is active

STEP 11: MONITORING & LOGS

--------------------------

View PM2 logs:

pm2 logs ouc-staging

View Nginx logs:

sudo tail -f /var/log/nginx/access.log

sudo tail -f /var/log/nginx/error.log

Monitor PM2 processes:

pm2 monit

IMPORTANT NOTES:

================

1. Replace "your-staging-domain.com" with your actual staging domain

2. Replace "your-username/your-repo-name" with your actual GitHub repo

3. Update all database credentials with your staging values

4. Create separate Google OAuth credentials for staging environment

5. Test each step before proceeding to the next

6. Keep your .env.local file secure and never commit it to Git

TROUBLESHOOTING:

================

If application won't start:

- Check PM2 logs: pm2 logs ouc-staging

- Verify environment variables: cat .env.local

- Check database connection: mysql -u staging\_user -p

If Nginx shows errors:

- Test config: sudo nginx -t

- Check logs: sudo tail -f /var/log/nginx/error.log

- Restart service: sudo systemctl restart nginx

If SSL certificate fails:

- Ensure domain points to server IP

- Check firewall: sudo ufw status

- Try manual certificate: sudo certbot certonly --nginx

END OF INSTRUCTIONS