**# Linux Deployment Guide for Touch-Not Application**

**## System Requirements**

- Ubuntu 20.04 LTS or higher

- Python 3.8+

- MySQL 8.0+

- Nginx

- Gunicorn

**## 1. System Preparation**

```bash

# Update system packages

sudo apt update && sudo apt upgrade -y

# Install required system dependencies

sudo apt install -y python3-pip python3-dev mysql-server nginx

```

**## 2. MySQL Setup**

```bash

# Secure MySQL installation

sudo mysql\_secure\_installation

# Create database and user

sudo mysql -u root -p

```

In MySQL prompt:

```sql

CREATE DATABASE cash\_collection CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

CREATE USER 'touch\_not\_user'@'localhost' IDENTIFIED BY 'your\_strong\_password';

GRANT ALL PRIVILEGES ON cash\_collection.\* TO 'touch\_not\_user'@'localhost';

FLUSH PRIVILEGES;

EXIT;

```

**## 3. Python Environment Setup**

```bash

# Create project directory

sudo mkdir -p /var/www/touch-not

sudo chown -R $USER:$USER /var/www/touch-not

# Create virtual environment

cd /var/www/touch-not

python3 -m venv venv

source venv/bin/activate

# Install dependencies

pip install -r requirements.txt

pip install gunicorn

```

**## 4. Application Configuration**

```bash

# Copy and configure environment file

cp .env.production.example .env.production

```

Edit `.env.production` with appropriate values:

```ini

FLASK\_ENV=production

DATABASE\_URL=mysql+pymysql://touch\_not\_user:your\_strong\_password@localhost/cash\_collection

SECRET\_KEY=your\_secure\_secret\_key

MYSQL\_HOST=localhost

MYSQL\_USER=touch\_not\_user

MYSQL\_PASSWORD=your\_strong\_password

MYSQL\_DB=cash\_collection

```

**## 5. Gunicorn Setup**

Create Gunicorn service file:

```bash

sudo nano /etc/systemd/system/touch-not.service

```

Add the following content:

```ini

[Unit]

Description=Gunicorn instance to serve Touch-Not application

After=network.target

[Service]

User=www-data

Group=www-data

WorkingDirectory=/var/www/touch-not

Environment="PATH=/var/www/touch-not/venv/bin"

ExecStart=/var/www/touch-not/venv/bin/gunicorn --workers 4 --bind unix:touch-not.sock -m 007 app:app

[Install]

WantedBy=multi-user.target

```

**## 6. Nginx Configuration**

Create Nginx server block:

```bash

sudo nano /etc/nginx/sites-available/touch-not

```

Add the following configuration:

```nginx

server {

    listen 80;

    server\_name your\_domain.com;

    location / {

        include proxy\_params;

        proxy\_pass http://unix:/var/www/touch-not/touch-not.sock;

    }

    location /static/ {

        alias /var/www/touch-not/static/;

    }

}

```

Enable the configuration:

```bash

sudo ln -s /etc/nginx/sites-available/touch-not /etc/nginx/sites-enabled

sudo nginx -t

sudo systemctl restart nginx

```

**## 7. SSL/TLS Configuration (Let's Encrypt)**

```bash

# Install Certbot

sudo apt install -y certbot python3-certbot-nginx

# Obtain and install certificate

sudo certbot --nginx -d your\_domain.com

```

**## 8. Start Application**

```bash

# Start and enable Gunicorn service

sudo systemctl start touch-not

sudo systemctl enable touch-not

# Check status

sudo systemctl status touch-not

```

**## 9. Database Migration**

```bash

# Activate virtual environment

source /var/www/touch-not/venv/bin/activate

# Run migrations

flask db upgrade

```

**## 10. Security Considerations**

1. Configure UFW Firewall:

```bash

sudo ufw allow 'Nginx Full'

sudo ufw allow ssh

sudo ufw enable

```

2. Set proper file permissions:

```bash

sudo chown -R www-data:www-data /var/www/touch-not

sudo chmod -R 755 /var/www/touch-not

```

3. Secure MySQL:

- Use strong passwords

- Regular security updates

- Backup strategy

**## 11. Monitoring and Maintenance**

1. Check logs:

```bash

sudo journalctl -u touch-not

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

```

2. Regular maintenance:

```bash

# Backup database

mysqldump -u touch\_not\_user -p cash\_collection > backup.sql

# Update system packages

sudo apt update && sudo apt upgrade -y

# Restart services if needed

sudo systemctl restart touch-not nginx

```

**## Troubleshooting**

1. If application fails to start:

- Check Gunicorn logs: `sudo journalctl -u touch-not`

- Verify environment variables

- Check database connectivity

2. If Nginx returns 502 Bad Gateway:

- Ensure Gunicorn is running

- Check socket permissions

- Verify Nginx configuration

3. Database connection issues:

- Verify MySQL credentials

- Check MySQL service status

- Ensure proper database permissions