🚀Flask App Hosting Guide — CyberPanel + OpenLiteSpeed + Gunicorn + MySQL (Hostinger VPS)

This guide walks you through the **complete process** of hosting a Flask web app with a MySQL database on **Hostinger VPS** using **CyberPanel** (with OpenLiteSpeed).

Server and App Overview

Component	Description
OS	Ubuntu 22.04 LTS
Web Panel	CyberPanel (OpenLiteSpeed)
Domain	`mba.mybillingapp.com`
App Repo	https://github.com/anni990/mybillingapp
App Entry Point	`run.py`
Branch to deploy	`anni`
Database	MySQL (via CyberPanel UI)
App Server	Gunicorn
Proxy Server	OpenLiteSpeed (reverse proxy to Gunicorn)

🏠1. SSH into VPS

```
ssh root@72.60.100.122
```

If you get a *host identification changed* warning:

```
ssh-keygen -R 72.60.100.122
ssh root@72.60.100.122
```

---

2. Update and Install Dependencies

```
apt update && apt upgrade -y
apt install git python3 python3-venv python3-pip -y
```

---

3. Install CyberPanel + OpenLiteSpeed

Run:

```
sh <(curl https://cyberpanel.net/install.sh || wget -O - https://
cyberpanel.net/install.sh)
```

During setup: - Select **OpenLiteSpeed** - Install **Full Service (Recommended)** - Choose **Yes** for PowerDNS, Postfix, and Pure-FTPD - Note the admin credentials (or reset later using below command)

Reset admin password if needed:

```
adminPass YOUR_NEW_PASSWORD
```

Access CyberPanel:

```
https://<your-server-ip>:8090
```

🪡4. Create Website in CyberPanel

1. Login → **Websites → Create Website**
2. Fill:
3. **Domain**: `mba.mybillingapp.com`
4. **Package**: Default
5. **Owner**: admin
6. **PHP Version**: (any, not used by Flask)
7. Click **Create Website**
8. Once created, navigate to:
9. `/home/mba.mybillingapp.com/public_html`

🚛5. Issue SSL Certificate

In CyberPanel: - Go to **SSL → Manage SSL** - Choose domain → Click **Issue SSL**

Test HTTPS:

```
https://mba.mybillingapp.com
```

You should see a CyberPanel default page.

---

6. Clone Flask App (Correct Branch)

SSH into the website root:

```
cd /home/mba.mybillingapp.com/public_html
rm -rf *  # Remove old code
git clone -b anni https://github.com/anni990/mybillingapp.git .
```

🏖️7. Setup Python Virtual Environment

```
python3 -m venv venv
source venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

🗻8. Install Database Dependencies (pyodbc Fix)

If you see this error: `libodbc.so.2: cannot open shared object file`

Run:

```
apt install unixodbc unixodbc-dev -y
```

If you use SQL Server:

```
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
curl https://packages.microsoft.com/config/ubuntu/22.04/prod.list > /etc/apt/
sources.list.d/mssql-release.list
apt update -y
ACCEPT_EULA=Y apt install msodbcsql18 -y
```

Reinstall pyodbc:

```
pip uninstall -y pyodbc
pip install pyodbc
```

🏠 9. Database Setup in CyberPanel

1. In CyberPanel → **Databases → Create Database**
2. Database Name: `billingdb`
3. Username: `billinguser`
4. Password: `StrongPass123`
5. Upload schema file:
6. Go to **Database → phpMyAdmin**
7. Choose your database → **Import → Choose** `schema.sql`
8. Update Flask config (e.g., `config.py` or `.env`):

```
SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://
billinguser:StrongPass123@localhost/billingdb'
```

---

10. Create `wsgi.py`

In `/home/mba.mybillingapp.com/public_html/wsgi.py`:

```
from run import app

if __name__ == "__main__":
    app.run()
```

---

11. Test Gunicorn

```
cd /home/mba.mybillingapp.com/public_html
source venv/bin/activate
gunicorn --bind 127.0.0.1:8000 wsgi:app
```

If it starts successfully, stop it with **Ctrl + C**.

---

12. Create Gunicorn Systemd Service

```
nano /etc/systemd/system/mybillingapp.service
```

Paste:

```
[Unit]
Description=Gunicorn instance for mba.mybillingapp.com
After=network.target

[Service]
User=root
Group=www-data
WorkingDirectory=/home/mba.mybillingapp.com/public_html
Environment="PATH=/home/mba.mybillingapp.com/public_html/venv/bin"
ExecStart=/home/mba.mybillingapp.com/public_html/venv/bin/gunicorn --workers
3 --bind 127.0.0.1:8000 wsgi:app

[Install]
WantedBy=multi-user.target
```

Enable service:

```
systemctl daemon-reload
systemctl enable mybillingapp
systemctl start mybillingapp
systemctl status mybillingapp
```

---

13. Configure OpenLiteSpeed Reverse Proxy

Step 1: Create External App (Web Server)

1. Go to **Virtual Hosts → mba.mybillingapp.com → External App**
2. Click **Add New**
3. Fill in:
4. **Name**: `gunicorn`
5. **Type**: `Web Server`
6. **Address**: `127.0.0.1:8000`
7. **Max Connections**: `100`
8. **Initial Request Timeout**: `60`
9. **Retry Timeout**: `0`
10. **Connection Keepalive Timeout**: `60`
11. **Response Buffering**: `No`
12. Save changes.

Restart LiteSpeed to apply:

```
/usr/local/lsws/bin/lswsctrl restart
```

Step 2: Create Context (Reverse Proxy)

1. Login to OpenLiteSpeed panel:

   ```
   https://<your-server-ip>:7080
   ```

2. Virtual Hosts → `mba.mybillingapp.com`
3. Go to **Context → Add**
4. Type: `Proxy`
5. URI: `/`
6. Web Server: select `gunicorn`
7. Enable: Yes
8. Save and **Graceful Restart** server.

---

14. Firewall Configuration

```
apt install ufw -y
ufw allow 22
ufw allow 80
ufw allow 443
ufw allow 8090
ufw allow 7080
ufw enable
```

---

15. Verification Steps

Test	Command / URL	Expected
Check Gunicorn	`systemctl status mybillingapp`	Active (running)
Check Reverse Proxy	`curl -I http://127.0.0.1:8000`	200 OK
Check HTTPS	https://mba.mybillingapp.com	Flask app loads
Check SSL	`sudo certbot certificates`	Valid cert shown

If `503 Service Temporarily Unavailable` → check Gunicorn status or proxy port mapping.

🚤Common Issues and Fixes

Issue	Cause	Fix
`libodbc.so.2 not found`	Missing UnixODBC	`apt install unixodbc unixodbc-dev -y`
503 error	Gunicorn not running	`systemctl restart mybillingapp`
SSL not loading	Certificate not issued	Re-issue SSL in CyberPanel
Wrong branch code	Master branch cloned	`git clone -b anni ...`

🚢Final Note

Your Flask app is now live at: 👉**https://mba.mybillingapp.com**

You can update code anytime with:

```
cd /home/mba.mybillingapp.com/public_html
git pull origin anni
systemctl restart mybillingapp
```

🛖Verification Summary

Run all checks to ensure proper deployment:

```
sudo systemctl status mybillingapp
sudo lsof -i:8000
curl -I https://mba.mybillingapp.com
```

If all show success → Deployment 👈🏽complete!