**LUMINA: Deployment Manual**

**AI-Assisted Mental Health Support Platform**  
**Version:** 1.0  
**Prepared by:**

**Date:**05/06/2025

**1. Introduction**

This document provides detailed instructions to deploy the LUMINA application into a production environment. It is intended for system administrators, DevOps engineers, and developers who are not part of the core development team.

**2. System Overview**

LUMINA is a Flask-based mental health support platform that offers:

* Real-time AI chatbot powered by **GPT-4**
* WebRTC voice communication
* Google and GitHub OAuth authentication
* SQLite for local storage (can be upgraded to PostgreSQL)
* Interactive progress dashboard using Chart.js

**3. Prerequisites**

**3.1 Server Requirements**

* OS: Ubuntu 20.04 / CentOS 7+ / macOS (for dev testing)
* RAM: Minimum 8GB
* Python: 3.10+
* Node.js: 18+
* Nginx or Apache (for reverse proxy)
* HTTPS setup via Let’s Encrypt

**3.2 External Services**

* OpenAI API key (for GPT-4 access)
* Google and GitHub OAuth credentials
* Domain and SSL certificate

**4. Project Setup**

**4.1 Clone Repository**

git clone https://github.com/your-repo/lumina.git

cd lumina

**4.2 Create Virtual Environment**

python3 -m venv venv

source venv/bin/activate

pip install -r requirements.txt

**5. Configuration**

**5.1 Environment Variables (.env)**

HF\_TOKEN=your-huggingface-api-token

LITELLM\_API\_KEY=your-litellm-api-key

OPENAI\_API\_KEY=your-openai-api-key

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

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

GITHUB\_CLIENT\_ID=your-github-client-id

GITHUB\_CLIENT\_SECRET=your-github-client-secret

SECRET\_KEY=your-flask-secret-key

**6. Database Setup**

**6.1 SQLite (default for local)**

python scripts/init\_db.py

**6.2 (Optional) PostgreSQL Setup**

* Create database and user
* Update SQLAlchemy config in config.py

**7. Frontend Setup**

cd frontend

npm install

npm run build

Copy the built assets to Flask's /static directory.

**8. Running the Application**

**8.1 Development**

flask run

**8.2 Production**

gunicorn -w 4 -b 0.0.0.0:8000 app:app

Use Nginx as a reverse proxy with HTTPS enabled.

**9. WebRTC & Socket Server (if applicable)**

cd realtime-server

npm install

node server.js

**10. Monitoring & Logs**

* Logs are stored in /var/log/lumina/
* Use supervisord, systemd, or pm2
* Monitor performance with htop or journalctl

**11. Backup & Recovery**

* Regularly back up .env, database (.db), and logs
* Use scheduled jobs or cron for automation

**12. Troubleshooting**

| **Issue** | **Possible Fix** |
| --- | --- |
| ModuleNotFoundError | Ensure virtual environment is activated and all packages are installed |
| CORS errors | Verify frontend/backend ports and allowed origins |
| OAuth failure | Check Google/GitHub credentials and redirect URIs |