

Deployment Guide - AI Mental Health Companion

This guide provides step-by-step instructions for deploying the AI Mental Health Companion in various environments.

Prerequisites

System Requirements

- **Python:** 3.11 or higher
- **Memory:** Minimum 4GB RAM (8GB+ recommended for LLM)
- **Storage:** 10GB+ free space
- **Network:** Stable internet connection

Required Accounts

- **TiDB Cloud:** For database hosting
- **LM Studio:** For local LLM hosting
- **Twilio** (Optional): For WhatsApp integration
- **Domain/Hosting** (Production): For public deployment

Local Development Setup

1. Environment Preparation

Bash

```
# Clone or navigate to project directory
cd mental_health_ai
```

```
# Create virtual environment (recommended)
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt
```

2. Database Configuration

TiDB Cloud Setup

1. **Create Account:** Sign up at [TiDB Cloud](#)
2. **Create Cluster:**
 - Choose "Developer Tier" for free usage
 - Select your preferred region
 - Note down connection details
3. **Configure Database:**
4. **Download SSL Certificate:** Download the CA certificate for secure connections

3. LM Studio Setup

Installation

1. **Download:** Visit [LM Studio](#) and download for your OS
2. **Install:** Follow the installation instructions
3. **Download Model:**
 - Recommended: `microsoft/Phi-3-mini-4k-instruct-gguf` or `mistralai/Mistral-7B-Instruct-v0.2-GGUF`
 - Choose appropriate quantization based on your hardware

Configuration

1. **Load Model:** Open LM Studio and load your chosen model

2. **Start Server:**

- Go to "Local Server" tab
- Click "Start Server"
- Default URL: `http://localhost:1234`
- Note: Keep LM Studio running while using the application

4. Environment Variables

Create a `.env` file in the project root:

Plain Text

```
# Database Configuration
TIDB_HOST=gateway01.us-west-2.prod.aws.tidbcloud.com
TIDB_USER=your_tidb_username
TIDB_PASSWORD=your_tidb_password
TIDB_DATABASE=mental_health_db
TIDB_PORT=4000
TIDB_CA_PATH=path/to/isrgrootx1.pem

# LM Studio Configuration
LM_STUDIO_URL=http://localhost:1234/v1
LM_STUDIO_MODEL=your-model-name

# Authentication
SECRET_KEY=your-super-secret-key-change-in-production
JWT_SECRET=your-jwt-secret-key-change-in-production

# Twilio (Optional - for WhatsApp)
TWILIO_ACCOUNT_SID=your_twilio_account_sid
TWILIO_AUTH_TOKEN=your_twilio_auth_token
TWILIO_PHONE_NUMBER=your_twilio_whatsapp_number

# Application Settings
DEBUG=True
LOG_LEVEL=INFO
```

5. Running the Application

Streamlit Web App (Primary Interface)

Bash

```
streamlit run streamlit_app.py
```

- Access at: `http://localhost:8501`
- Features: Full web interface with all functionality

API Server (For External Integrations)

Bash

```
python api_server.py
```

- Access at: `http://localhost:5001`
- Features: REST API endpoints

Authentication Server

Bash

```
python auth_system.py
```

- Access at: `http://localhost:5002`
- Features: User registration and authentication

WhatsApp Integration (Optional)

Bash

```
python whatsapp_integration.py
```

- Access at: `http://localhost:5000`
- Features: WhatsApp bot functionality

Production Deployment

Option 1: Cloud Platform Deployment

Heroku Deployment

1. **Prepare Files:**
2. **Deploy:**

AWS/GCP/Azure Deployment

1. **Container Deployment:**
2. **Build and Deploy:**

Option 2: VPS/Dedicated Server

Server Setup (Ubuntu/Debian)

Bash

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install Python and dependencies
sudo apt install python3.11 python3.11-venv python3-pip nginx -y

# Create application user
sudo useradd -m -s /bin/bash mentalhealth
sudo su - mentalhealth

# Clone and setup application
git clone <your-repo> mental_health_ai
cd mental_health_ai
python3.11 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Nginx Configuration

Plain Text

```
# /etc/nginx/sites-available/mental-health-ai
server {
    listen 80;
    server_name your-domain.com;

    location / {
        proxy_pass http://localhost:8501;
        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;
    }

    location /api {
        proxy_pass http://localhost:5001;
        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;
    }
}
```

Systemd Service

Plain Text

```
# /etc/systemd/system/mental-health-ai.service
[Unit]
Description=Mental Health AI Companion
After=network.target

[Service]
Type=simple
User=mentalhealth
WorkingDirectory=/home/mentalhealth/mental_health_ai
Environment=PATH=/home/mentalhealth/mental_health_ai/venv/bin
ExecStart=/home/mentalhealth/mental_health_ai/venv/bin/streamlit run
streamlit_app.py --server.address 0.0.0.0 --server.port 8501
Restart=always

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

Bash

```
# Enable and start service
sudo systemctl enable mental-health-ai
sudo systemctl start mental-health-ai
sudo systemctl enable nginx
sudo systemctl start nginx
```



Security Configuration

SSL/TLS Setup (Let's Encrypt)

Bash

```
# Install Certbot
sudo apt install certbot python3-certbot-nginx -y

# Obtain certificate
sudo certbot --nginx -d your-domain.com

# Auto-renewal
sudo crontab -e
# Add: 0 12 * * * /usr/bin/certbot renew --quiet
```

Firewall Configuration

Bash

```
# UFW setup
sudo ufw allow ssh
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

Environment Security

Bash

```
# Secure environment file
chmod 600 .env
```

```
chown mentalhealth:mentalhealth .env
```



Monitoring and Maintenance

Log Management

Bash

```
# Application logs
tail -f /var/log/mental-health-ai.log

# Nginx logs
tail -f /var/log/nginx/access.log
tail -f /var/log/nginx/error.log

# System logs
journalctl -u mental-health-ai -f
```

Health Checks

Bash

```
# Create health check script
cat > health_check.sh << 'EOF'
#!/bin/bash
curl -f http://localhost:8501/_stcore/health || exit 1
curl -f http://localhost:5001/ || exit 1
EOF

chmod +x health_check.sh
```

Backup Strategy

Bash

```
# Database backup (TiDB Cloud has automatic backups)
# Application data backup
tar -czf backup_$(date +%Y%m%d).tar.gz mental_health_ai/

# Automated backup script
cat > backup.sh << 'EOF'
```



```
#!/bin/bash
BACKUP_DIR="/home/mentalhealth/backups"
mkdir -p $BACKUP_DIR
tar -czf $BACKUP_DIR/backup_$(date +%Y%m%d_%H%M%S).tar.gz mental_health_ai/
find $BACKUP_DIR -name "backup_*.tar.gz" -mtime +7 -delete
EOF
```

Troubleshooting

Common Issues

LM Studio Connection Issues

Bash

```
# Check if LM Studio is running
curl http://localhost:1234/v1/models

# Restart LM Studio server
# In LM Studio: Stop Server → Start Server
```

Database Connection Issues

Bash

```
# Test TiDB connection
python -c "
import pymysql
conn = pymysql.connect(
    host='your-host',
    user='your-user',
    password='your-password',
    database='your-db',
    port=4000,
    ssl_ca='path/to/cert.pem'
)
print('Database connection successful!')
conn.close()
"
```

Memory Issues

Bash

```
# Monitor memory usage
htop
free -h

# Optimize for low memory
export STREAMLIT_SERVER_MAX_UPLOAD_SIZE=50
export STREAMLIT_SERVER_MAX_MESSAGE_SIZE=50
```

Performance Optimization

Application Optimization

Python

```
# In streamlit_app.py, add caching
@st.cache_data
def load_model():
    # Cache expensive operations
    pass

@st.cache_resource
def get_database_connection():
    # Cache database connections
    pass
```

Server Optimization

Bash

```
# Increase file limits
echo "* soft nofile 65536" >> /etc/security/limits.conf
echo "* hard nofile 65536" >> /etc/security/limits.conf

# Optimize Python
export PYTHONOPTIMIZE=1
```



Mobile Considerations

Progressive Web App (PWA)

Add to `streamlit_app.py` :

Python

```
# PWA configuration
st.set_page_config(
    page_title="Mental Health AI",
    page_icon="🧠",
    layout="wide",
    initial_sidebar_state="collapsed" # Better for mobile
)

# Add mobile-friendly CSS
st.markdown("""
<style>
@media (max-width: 768px) {
    .main .block-container {
        padding-top: 1rem;
        padding-left: 1rem;
        padding-right: 1rem;
    }
}
</style>
""", unsafe_allow_html=True)
```

Next Steps

After successful deployment:

1. **Test All Features:** Verify chat, mood tracking, journaling, etc.
2. **Monitor Performance:** Check logs and system resources
3. **User Testing:** Gather feedback from initial users
4. **Security Audit:** Review security configurations
5. **Backup Verification:** Test backup and restore procedures
6. **Documentation:** Update any deployment-specific documentation

Support

For deployment issues:

1. Check the troubleshooting section above
2. Review application logs
3. Verify all environment variables are set correctly
4. Ensure all external services (TiDB, LM Studio) are running
5. Test individual components separately

Happy Deploying! 🚀