Technical Installation Manual

Intorduction

WriteMe is a revolutionary platform for collaborative writing. Using Natural Language Processing, it offers tools to enhance narrative structures, character development, and emotional impact. Like GitHub for writers, users can share, create, and enjoy stories with community support, transforming writing into a collaborative experience

This system contains a Next.js (webapp) and a python AI service (backend) to be installed to reach full functionality. The following needs to be installed for each section:

Additionaly a database (postgres) and a object storage (s3) need to be provisioned.

Webapp (Next.js):

- `git`
- `fnm` / `node`
- `pnpm`

AI Service:

- `poetry`
- `python`

Database:

- Postgres

Object Storage:

- S3/Minio

Pre-Requisites

Development with Docker

PC Hardware	Minimum Requirements
CPU	4 Core CPU @4.0GHz
Memory	12 GB
Storage	10GB

Software, Packages & Services	Version Required
Browser: Chrome/Brave	Latest
fnm	lastest
pnpm	latest
Docker	latest
Postgres	>=16
nx	latest
Node	v20.11.1
S3	AWS Account
GitHub	OAuth App
Google	OAuth App

How To Intall Prerequisites

The following assumes WriteMe has been cloned from GitHub and your current working directory is the cloned folder.

Next application

- Node / `fnm` (node version manager) Unix:

```
curl -fsSL https://fnm.vercel.app/install | bash && fnm install
```

Windows:

```
winget install Schniz.fnm
fnm install
```

for any additional step please refer to the guide

- Pnpm (Performant Node Package Manager)

Unix:

```
curl -fsSL https://get.pnpm.io/install.sh | sh -
```

Windows:

```
iwr https://get.pnpm.io/install.ps1 -useb | iex
```

for any additional step please refer to the guide

AI service Backend

This assumes that your current working directory is the `ml_backend` folder.

- python

Based on your current operating system, choose and official installer.

- poetry

Unix:

```
curl -sSL https://install.python-poetry.org | python3 -
```

Windows:

```
pipx install poetry
```

- fastapi

```
pip install "fastapi[standard]"
```

- (optional) Set your Virtual Environment

Depending on your shell

```
source .venv/bin/activate.[fish|bash|zsh]
```

Local Developement

- Docker

Ubuntu

```
sudo apt update && sudo apt install docker docker-compose
```

For Windows users:

[Docker Desktop](#)

For other operating system please refer to the following [guide](#).

Environment Variables

The project assumes that the following Environment variables are set.

```
# Authenication

AUTH_DRIZZLE_URL=postgres://postgres:password@localhost:5432/db (postgres
connection string)
AUTH_SECRET=<random 32bit string>
GITHUB_ID=<obtained from GitHub Apps>
GITHUB_SECRET=<obtained from GitHub Apps>
GOOGLE_CLIENT_ID=<obtained from Google Developer Console>
GOOGLE_CLIENT_SECRET=<obtained from Google Developer Console>
```

```
# Database - connection parameters to your database

DB_HOST=localhost
DB_USER=postgres
DB_PASSWORD=password
DB_NAME=db
DB_PORT=5432

# S3 - Assumes you have an S3 or compatible service

S3_REGION=<Obtained from AWS Console>
S3_ENDPOINT=<Obtained from AWS Console>
S3_SECRET_KEY=<Obtained from AWS Console>
S3_ACCESS_KEY=<Obtained from AWS Console>
S3_BUCKET=<Obtained from AWS Console>

# Public Endpoint of the AI Service
NEXT_PUBLIC_AI_ENDPOINT=https://ai.writeme.co.za
```

Installation

- all additional node dependencies

```
pnpm i
```

- Install additional python dependecies

This assumes `ml_backend` is your current working directory.

```
poetry install
```

Deployment/Running

WriteMe (Next application)

- Development Build

```
pnpm exec nx run writeme:dev
```

- Production build

```
export NODE_ENV=production
pnpm exec nx run writeme:start
```

both options run the server on port `3000` by default.

http://localhost:3000

WriteMe Docs website

```
pnpm exec nx run writeme-docs:start
```

This runs the documentation website on port `3001` by default.

http://localhost:3001

AI Service

This assumes that `ml_backend` is your current working directory.

```
fastapi run ml_backend/app/app.py
```

Postgres for Development

Unix

```
sudo docker-compose up
```

this will start a developement postgres instance on your local machine.

`Ctrl+C` in order to stop the instance.

Seeding the database

```
pnpm tsx ./apps/writeme/db/seed.ts
```

> note first run may take upto 10 minutes to start (based on internet speed), as machine
> learning models will be downloaded.

This runs the service on port `8000` by default.

http://localhost:8000/docs

✏️ Edit this page