MusicStore on Windows Platform:

Installation and Configuration Guide

For IT-tech students prepared by: Timo Hyyppä

Date: 20.10.2024

Table of Contents:

Appendix

1. Introduction

This guide provides detailed instructions for cloning, setting up, and testing the MusicStore application on a Windows platform. It includes steps for configuring PostgreSQL, creating and installing self-signed certificates, and setting up the application to run securely over HTTPS. Notes for Linux users are included where necessary.

2. Prerequisites

Ensure you have the following tools installed on your system:

Git: Version control system.

Download and install from Git for Windows.

Python: Programming language (version 3.6 or higher).

Download and install from Python's official website.

PostgreSQL: Database system.

Download and install from PostgreSQL's official website.

OpenSSL: Toolkit for SSL/TLS.

Download and install from OpenSSL for Windows.

3. Cloning the Repository

Open Git Bash and run the following commands:

git clone https://github.com/tihyyti/MusicStore.git

cd MusicStore

4. Setting Up the Virtual Environment

Create and activate a virtual environment:

python -m venv venv

venv\Scripts\activate

5. Installing Dependencies

Install the required Python packages:

pip install -r requirements.txt

6. Setting Up PostgreSQL

Ensure PostgreSQL is running and create a new database and user:

Open pgAdmin or use the psql command-line tool.

Create a new database and user in sql:

CREATE DATABASE musicstore;

CREATE USER musicuser WITH PASSWORD 'yourpassword';

GRANT ALL PRIVILEGES ON DATABASE musicstore TO musicuser;

7. Configuring Environment Variables

Create a .env file in the project root directory with the following content:

FLASK_APP=run.py

FLASK_ENV=development

DATABASE_URL=postgresql://musicuser:yourpassword@localhost:5432/musicstore

8. Initializing the Database

Run the following commands to set up the database schema:

flask db init

flask db migrate

flask db upgrade

9. Creating and Installing Self-Signed Certificates

Generate a self-signed certificate using OpenSSL:

Open Command Prompt and navigate to your project directory.

Run the following command:

openssl req -x509 -newkey rsa:4096 -nodes -out app\certs\cert.pem -keyout app \certs\key.pem -days 365

10. Running the Application

Start the Flask application:

python run.py

11. Installing Certificates in Web Browsers

Firefox

Open Firefox and navigate to Settings > Privacy & Security > Certificates > View Certificates.

Go to the Authorities tab and click on Import.

Select app\certs\cert.pem and import it.

Check the box to Trust this CA to identify websites and click OK.

Microsoft Edge

Open Edge and navigate to Settings > Privacy, search, and services > Security > Manage certificates.

Import the certificate:

Click on Import.

Select app\certs\cert.pem.

Choose Trusted Root Certification Authorities as the store location.

Complete the import process.

Safari (Mac)

Open Keychain Access.

Drag and drop app\certs\cert.pem into the System keychain.

Set the certificate to Always Trust.

12. Appendix

A. Common Errors and Troubleshooting

Virtual Environment Activation Issues: Ensure you are using the correct command for your operating system.

Database Connection Errors: Verify PostgreSQL is running and the database URL in the .env file is correct.

Certificate Warnings in Browsers: Ensure the certificate is correctly installed and trusted in the browser.