**Руководство программиста**

по теме «Информационная система для магазина по продаже ПО»

СОДЕРЖАНИЕ

[1. Инсталляция программы 3](#_Toc182785445)

[1.1. Руководство по установке проекта 3](#_Toc182785446)

[1.1.1. Предварительные требования 3](#_Toc182785447)

[1.1.2. Клонирование проекта 3](#_Toc182785448)

[1.1.3. Настройка переменных окружения 3](#_Toc182785449)

[1.1.4. Запуск MySQL через Docker 4](#_Toc182785450)

[1.1.5. Установка и запуск Backend (FastAPI) 5](#_Toc182785451)

[1.1.6. Установка и запуск Frontend (Nuxt) 5](#_Toc182785452)

[1.1.7. Проверка работы проекта 6](#_Toc182785453)

[1.1.8. Возможные проблемы и их решения 7](#_Toc182785454)

[2. Проверка работоспособности информационной системы 8](#_Toc182785455)

[3. Поддержка программы 9](#_Toc182785456)

# Инсталляция программы

## Руководство по установке проекта

Данное руководство поможет установить и запустить проект, который включает:

1. MySQL в Docker.

2. Frontend на Nuxt.js.

3. Backend на FastAPI.

### Предварительные требования

- Установленный Docker (https://www.docker.com/).

- Установленный Docker Compose (https://docs.docker.com/compose/).

- Установленный Node.js (https://nodejs.org/) версии 20 или выше.

- Установленный Python (https://www.python.org/) версии 3.8 или выше.

- Установленный Git (https://git-scm.com/).

### Клонирование проекта

Клонируйте репозиторий с GitHub:

git clone https://github.com/PapaSavage/UP03

cd up03

### Настройка переменных окружения

Для MySQL cоздайте файл `.env` в корневой папке или используйте предоставленный `.env.example`. Пример:

MYSQL\_ROOT\_PASSWORD=your\_root\_password

MYSQL\_DATABASE=your\_database\_name

MYSQL\_USER=your\_user

MYSQL\_PASSWORD=your\_password

Для Backend (FastAPI) cоздайте файл `.env` в папке `backend` (или используйте `.env.example`):

DATABASE\_URL=mysql+pymysql://your\_user:your\_password@mysql/your\_database\_name

SECRET\_KEY=your\_secret\_key

Для Frontend (Nuxt) cоздайте файл `.env` в папке `frontend` (или используйте `.env.example`):

API\_BASE\_URL=http://localhost:8000/api

### Запуск MySQL через Docker

Перейдите в папку с `docker-compose.yml` (если он в корне проекта, останьтесь там). Запустите контейнеры:

docker-compose up -d

Проверьте, что MySQL запущен:

docker ps

### Установка и запуск Backend (FastAPI)

1. Перейдите в папку `backend`:

cd backend

2. Создайте и активируйте виртуальное окружение (опционально):

python -m venv venv

source venv/bin/activate # Для Windows: venv\Scripts\activate

3. Установите зависимости:

pip install -r requirements.txt

4. Выполните миграции базы данных:

alembic upgrade head

5. Запустите сервер:

uvicorn main:app --host 0.0.0.0 --port 8000 --reload

### Установка и запуск Frontend (Nuxt)

1. Перейдите в папку `frontend`:

cd ../frontend

2. Установите зависимости:

npm install

3. Запустите сервер:

npm run dev

Nuxt будет доступен по адресу [http://localhost:3000](http://localhost:3000).

### Проверка работы проекта

1. Убедитесь, что все сервисы работают:

* MySQL (контейнер Docker).
* Backend FastAPI (по адресу [http://localhost:8000/docs] (http://localhost:8000/docs)).
* Frontend Nuxt (по адресу [http://localhost:3000](http://localhost:3000)).

2. Проверьте подключение Nuxt к FastAPI, отправив запрос с фронтенда.

### Возможные проблемы и их решения

1. Ошибка подключения к базе данных.

* Проверьте настройки `DATABASE\_URL` в `.env` для Backend.
* Убедитесь, что MySQL контейнер запущен.

2. Nuxt не подключается к API.

* Проверьте значение `API\_BASE\_URL` в `.env` для Frontend.
* Убедитесь, что FastAPI запущен.

3. Ошибки миграций базы данных.

* Проверьте настройки подключения в FastAPI.
* Убедитесь, что MySQL контейнер доступен по указанным в `.env` параметрам.

# Проверка работоспособности информационной системы

Для проверки работоспособности системы выполните следующие действия:

1. Запустите приложение по приведённой инструкции.
2. Перейдите по ссылке <http://localhost:3000>.
3. Проверьте работу всех кнопок, включая добавление, удаление, редактирование данных, оформление заказов и генерацию отчетов, чтобы убедиться, что они работают корректно.

# Поддержка программы

Для поддержания работоспособности и актуальности информационной системы выполните следующие действия:

1. **Обновление зависимостей**: регулярно обновляйте Python и все используемые библиотеки. Это поможет избежать проблем с совместимостью и повысит безопасность приложения.
2. **Отслеживание отзывов и ошибок**: проверяйте репозиторий на GitHub на предмет новых отзывов, сообщений об ошибках и запросов от пользователей. При возникновении проблем оперативно анализируйте и исправляйте их, чтобы поддерживать высокое качество работы программы.