**Руководство программиста**

по теме «Информационная система для автошколы»

Оглавление

[1. Инсталляция программы 3](#_Toc182739455)

[1.1. Руководство по установке проекта 3](#_Toc182739456)

[1.1.1. Предварительные требования 3](#_Toc182739457)

[1.1.2. Клонирование проекта 3](#_Toc182739458)

[1.1.3. Настройка переменных окружения 3](#_Toc182739459)

[1.1.4. Запуск MySQL через Docker 4](#_Toc182739460)

[1.1.5. Установка и запуск Backend (FastAPI) 5](#_Toc182739461)

[1.1.6. Установка и запуск Frontend (Nuxt) 5](#_Toc182739462)

[1.1.7. Проверка работы проекта 6](#_Toc182739463)

[1.1.8. Возможные проблемы и их решения 7](#_Toc182739464)

[2. Проверка работоспособности информационной системы 8](#_Toc182739465)

[3. Поддержка программы 10](#_Toc182739466)

# Инсталляция программы

## Руководство по установке проекта

Данное руководство поможет установить и запустить проект "Автошкола", включающий:

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/drivingschool

cd drivingschool

### 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`:

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`:

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.

1. Nuxt не подключается к API.

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

1. 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

* Перейдите по адресу <http://localhost:3000>.
* Войдите в систему или зарегистрируйтесь, если это предусмотрено функционалом.

Функции системы

Добавление студентов:

* В разделе управления студентами можно добавить нового ученика, заполнив соответствующую форму.

Назначение курсов:

* Назначьте студенту курсы в зависимости от выбранной категории прав.

Управление инструкторами:

* Добавьте или редактируйте информацию об инструкторах.

Планирование занятий:

* Создавайте расписание занятий для каждого инструктора и группы студентов.

Просмотр успеваемости:

* Отслеживайте посещаемость и результаты тестов студентов.

Формирование отчетов:

* Генерируйте отчеты по успеваемости студентов, активности инструкторов и доходам автошколы.

Администраторская панель

* Если вы администратор, вы можете управлять всей системой: студентами, инструкторами, курсами и отчетами.

Проверка связи

* Убедитесь, что система отображает актуальные данные. Если что-то не работает:
* Проверьте работу Backend (FastAPI) по адресу <http://localhost:8000/docs>.
* Проверьте правильность настроек API в файле .env Frontend.

## 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.