**Инструкция по проекту «Сетевой лицей»**

**для тестирования и разработки**



**Содержание**

[Инструкция по запуску проекта «Сетевой лицей» для тестирования и разработки 2](#_Toc131350104)

[Необходимое программное обеспечение 2](#_Toc131350105)

[Запуск backend 5](#_Toc131350106)

[Запуск frontend 7](#_Toc131350107)

[Описание структуры проекта 8](#_Toc131350108)

[Список использованных источников 10](#_Toc131350109)

# Инструкция по запуску проекта «Сетевой лицей» для тестирования и разработки

## Необходимое программное обеспечение

Программное обеспечение, которое необходимо установить для открытия проекта на локальном ПК:

1. Miniconda Python v 3.9 для вашей ОС0020 [[1]](#_Список_использованных_источников).

Для проверки того, успешно ли прошла установка, необходимо (актуально для ОС Windows) в командной строке (Win+R >> “cmd”>>Enter) ввести команду «conda --version». Если установка прошла успешно, то вы получите ответ на команду с информацией об установленной у вас версии Miniconda (Рис. 1).

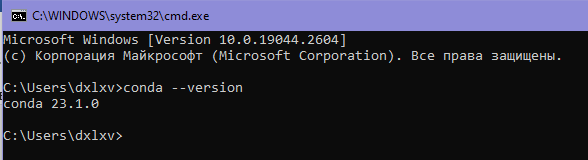


Рисунок 1 – Установка Miniconda

1. Node.js 16.15.1 LTS – среда исполнения Java Script [[2]](#_Список_использованных_источников).

Для проверки корректности установки также используем командную строку (для ОС Windows) введём команду «node» (Рис. 2). При успешной установке вы получите информацию об установленной у вас версии Node.js. Вместе с Node.js должен также установиться пакетный менеджер npm, необходимый для frontend-разработки.

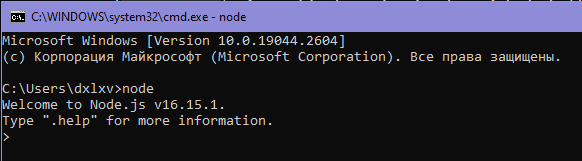


Рисунок 2 – Установка Node.js

1. Git [[3]](#_Список_использованных_источников). Любая версия. Установщик ­Standalone Installer.

Необходим для командной разработки.

1. Visual Studio Code [[4]](#_Список_использованных_источников). Далее настроим VS Code для работы с фреймворком Django.
   1. Откроем VS Code, создадим новый терминал (Terminal>>New Terminal). Выберем вместо PowerShell – Command Prompt.
   2. Установим недостающие плагины (Расширения) (Рис.3).

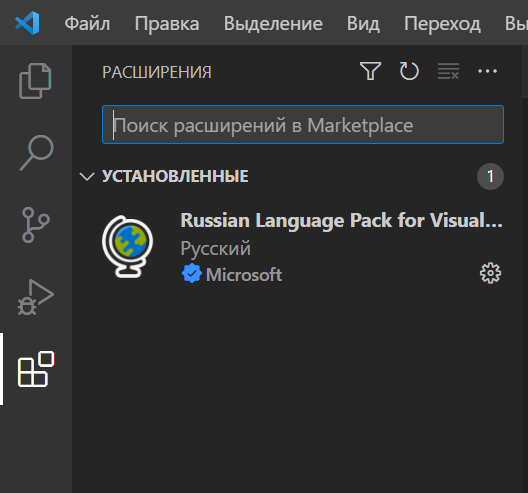


Рисунок 3 – Доступные плагины (расширения)

Нам необходимо установить следующие плагины (рекомендуется скачивать плагины с наибольшим количеством скачиваний):

* Python
* Django
* Vue Language Features (Volar)
* SQL Server (mssql)

Плагин для визуального удобства (необязательно):

– Vscode-icons

1. PostgreSQL v 15.2 [[5]](#_Список_использованных_источников). Скачиваем.

(Внимание! Не поддерживается 32-х разрядной ОС Windows)

Установку производим по умолчанию, имя пользователя и пароль: root. Запускаем pgAdmin 4 (В меню ПУСК). Далее нам необходимо создать базу данных (Рис. 4). Database->Create, название БД – (например) setlic.

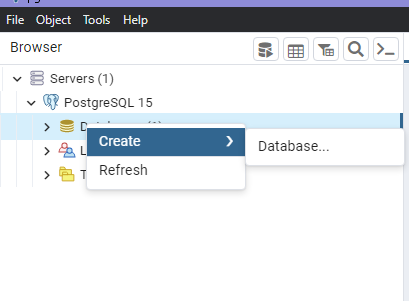


Рисунок 4 – Создание БД

Чтобы корректно работать с проектом, необходимо выгрузить в свой локально развёрнутый проект данные, уже существующие в проекте – бекап.

Для загрузки бекапа в существующую БД надо в pgAdmin нажать ПКМ на ранее созданную пустую БД setlic -> PSQL Tool (Рис. 5)

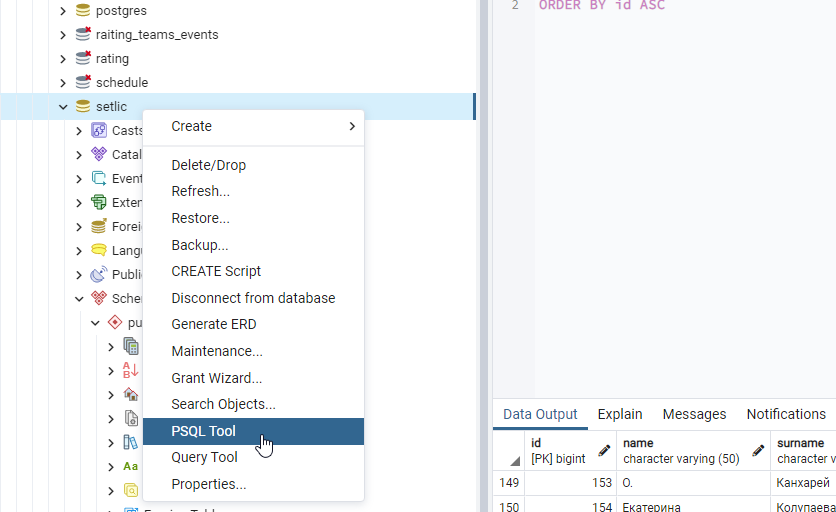


Рисунок 5 – Открытие терминала в Postgre

Справа откроется терминал, куда нужно вписать команду:

\i 'd:\setlic.sql'

Где 'd:\setlic.sql' – это путь к файлу с бекапом, использовать обязательно одинарные кавычки. Ваш путь может отличаться. Нажимаем Enter (Рис. 6).

Ожидаем пока бекап полностью не загрузится в БД.

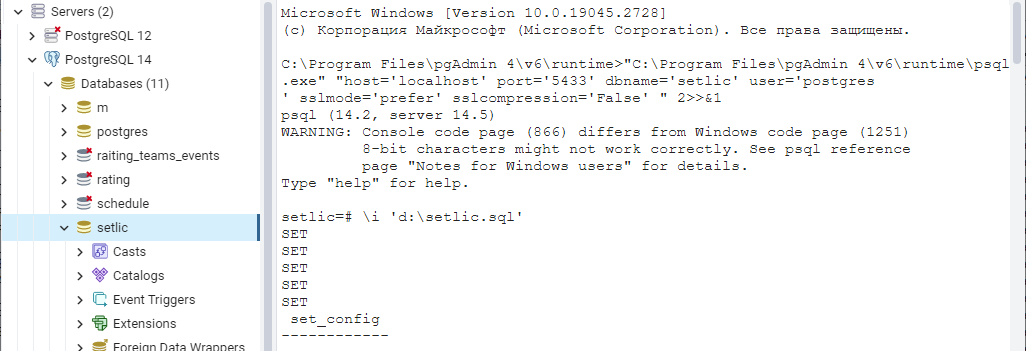


Рисунок 6 – Загрузка бекапа

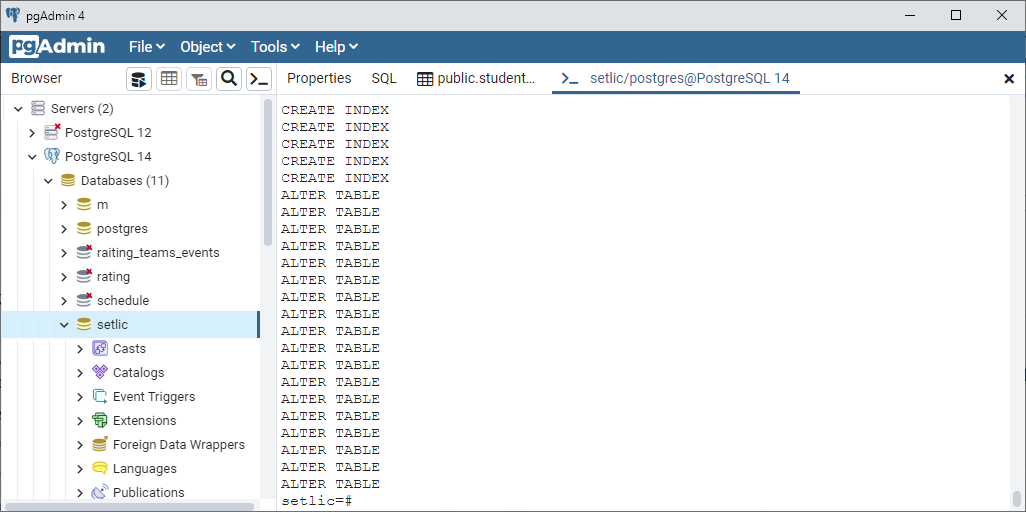


Рисунок 7 – Загрузка прошла успешно

После импорта бекапа вы имеете на локальном ПК полную копию БД с сервера (Рис.7).

В папке проекта lessons создадим файл local.py со следующим содержанием:

DATABASES = {

'default':{

'ENGINE': 'django.db.backends.postgresql',

'NAME': 'setlic', // имя базы данных, которую вы создали в PostgreSQL

'USER': 'root', //имя пользователя

'PASSWORD': 'root', //пароль пользователя

}}

## Запуск backend

В структуре проекта находится файл requirements.txt, где указано недостающее программное обеспечение, чтобы его установить нужно в терминале VS Code (Command Prompt) ввести команду:

pip install -r requirements.txt

Далее прописываем следующие команды:

1. python manage.py makemigrations
2. python manage.py migrate
3. python manage.py createsuperuser

Имя пользователя: admin

Адрес электронной почты: [admin@mail.ru](mailto:admin@mail.ru) (или любой другой)

Password: admin (консоль при вводе пароля не отображает символы)

1. В дальнейшем рекомендуется сменить пароль админа на локальной машине, сделать это можно с помощью команды (Рис. 8):

python manage.py changepassword admin

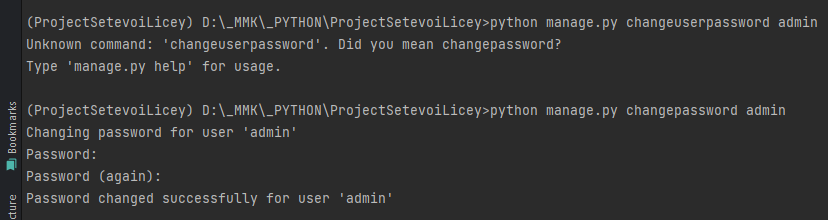


Рисунок 8 – Смена пароля учётной записи админа

Теперь вы имеете возможность входить по новому паролю под пользователем admin.

1. python manage.py runserver

При корректной установке и настройке имеем следующее (Рис. 9):

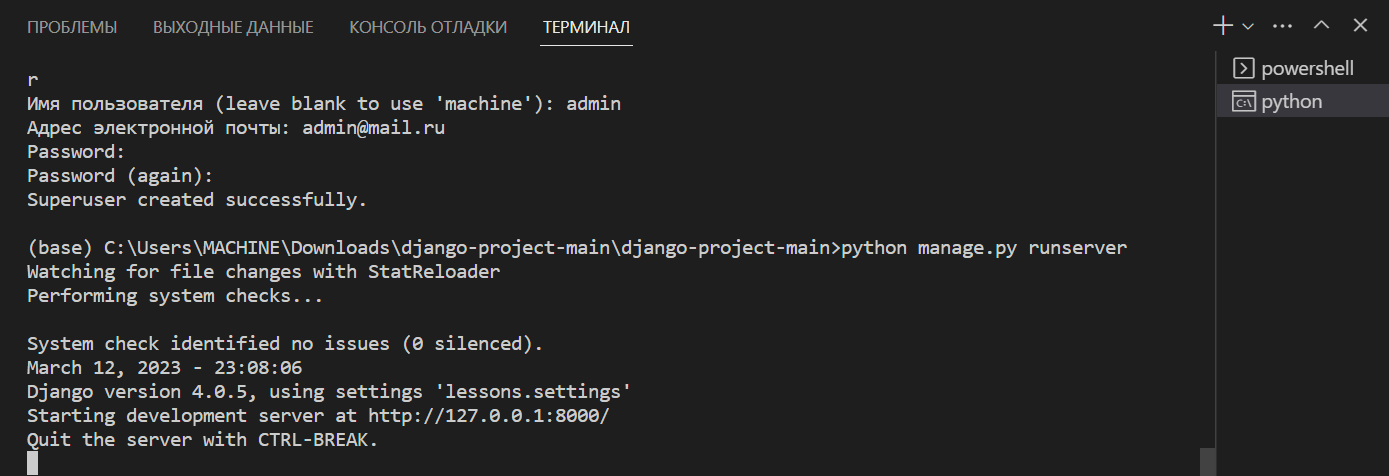


Рисунок 9 – Запуск backend

Для открытия бэкенда нужно перейти по URL после «Starting development server at….». Для перехода в раздел администрирования необходимо в адресной строке браузера привести адрес к виду: 127.0.0.1:8000/admin/

Если всё прошло успешно, то входим под пользователем admin в учётную запись Django (Рис. 10):

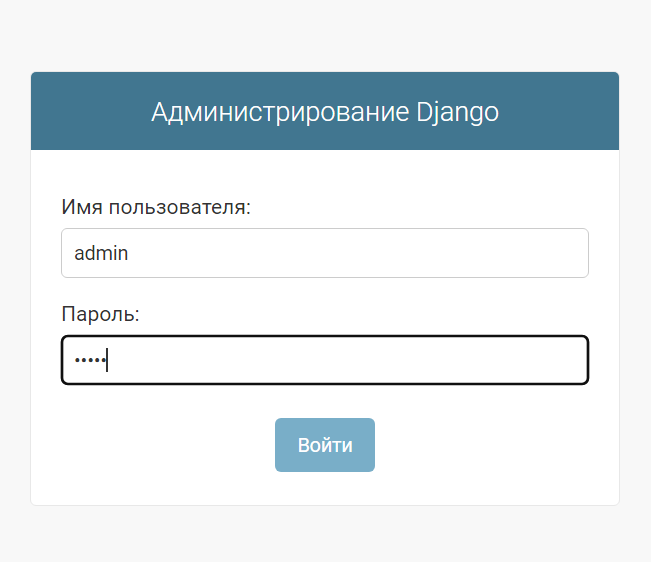


Рисунок 10 – Администрирование Django

## Запуск frontend

Для открытия фронтенда (корректная работа возможна только при предварительном открытии бэкенда), нужно прописать в терминале следующие команды:

1. cd client //переходим в директорию client
2. npm install //устанавливаем пакетный менеджер
3. npm run dev //запускаем отладку

В терминале должна появиться следующая информация (Рис. 11):

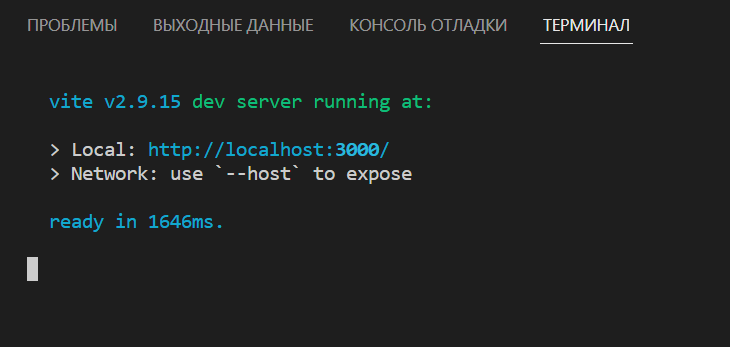


Рисунок 11 – Запуск фронтенда

Переходим по ссылке >Local. На этом открытие проекта завершено (Рис. 12).

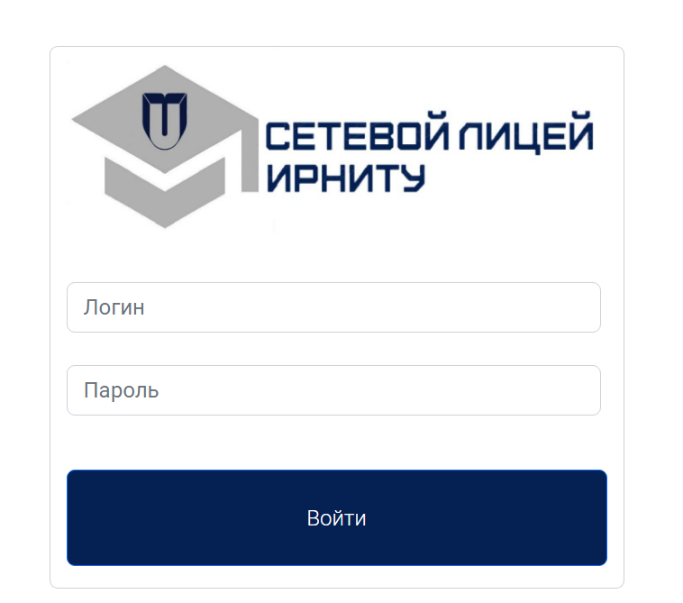
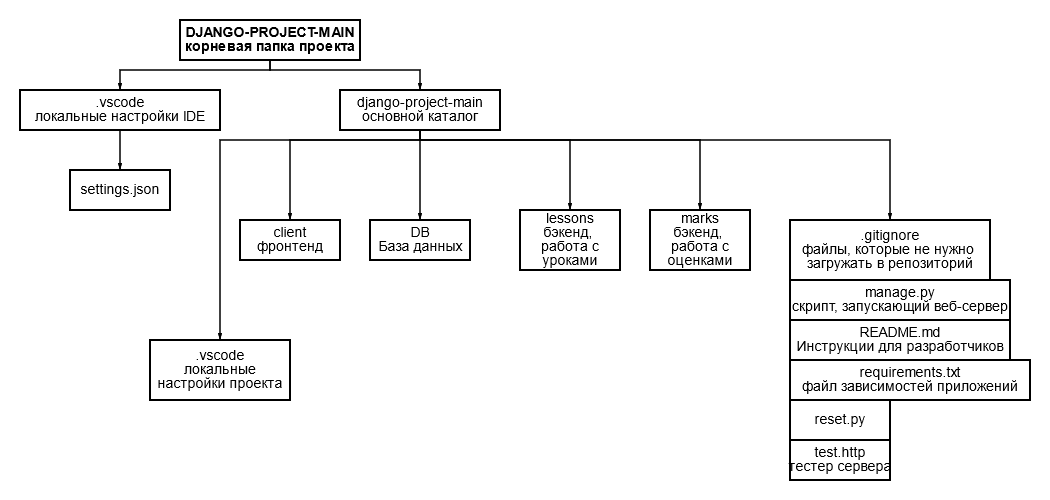


Рисунок 12 – Запуск фронтенда

# Описание структуры проекта

Корневая папка проекта «Сетевой лицей» называется Django-project-main. Далее находятся две папки: .vscode, где находятся локальные настройки IDE для текущего проекта и папка django-project-main, содержащая непосредственно файлы сайта.



Рассмотрим наиболее важные для разработки разделы, а именно: client, lessons, marks.

В модуях «lessons» и «marks» находятся файлы бэкенда, язык разработки ­– Python, фреймворк Django. Для работы с этими модулями необходимы знания и навыки в Django и Python.

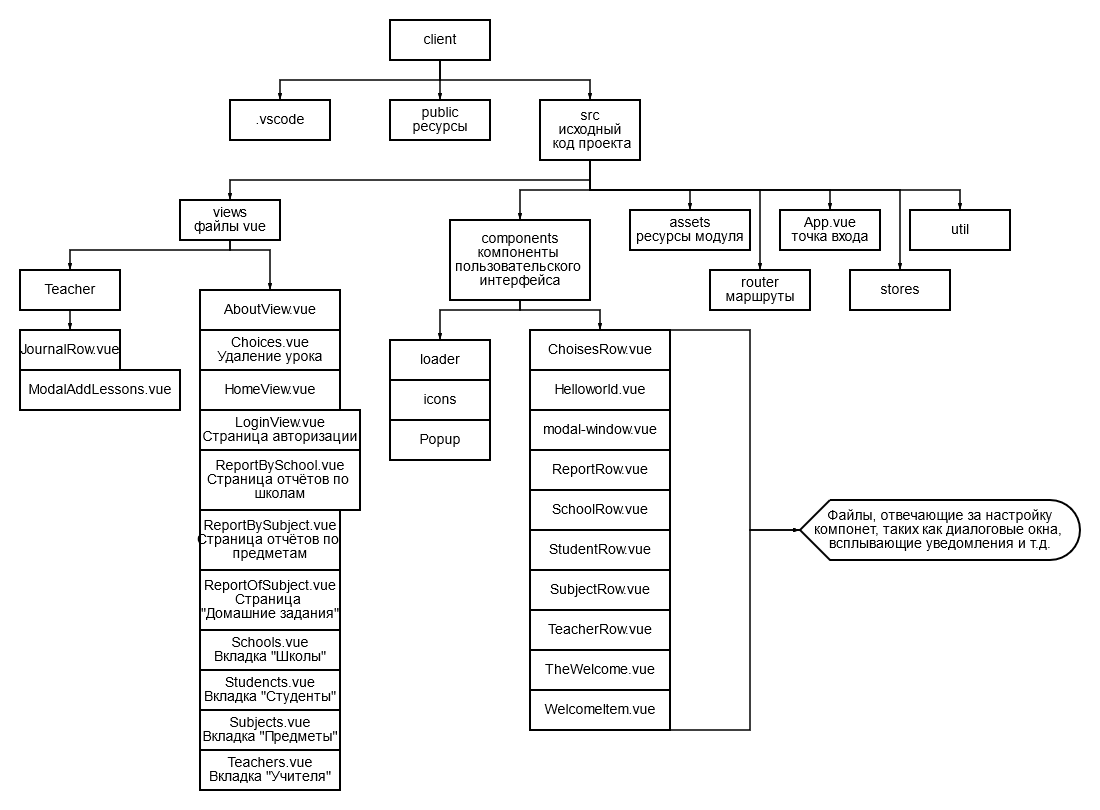
Для написания фронтенда был использован фреймворк Vue.js, более подробно с ним можно ознакомится в источнике [6]. Фреймворк используется непосредственно для разработки той части приложения, которая выполняется на стороне клиента. На диаграмме кратко описаны модули раздела client, для облегчения навигации по структуре проекта.

Рассмотрим наиболее важные компоненты, на которые стоит обратить внимание разработчикам:

Файлы с расширением .vue являются непосредственно страницами веб-приложения. Для работы с данными файлами рекомендуются базовые знания таких языков как HTML5, JavaScript и CSS.

В модуле «lessons» находится файл local.py, содержащий в себе данные для открытия проекта на конкретной машине, в нашем случае о настройках локальной базы данных.

Текстовый файл requiremets.txt – файл зависимостей приложений, который позволяет правильно установить окружение, изменять данные в нём – не рекомендуется.



Представленная выше диаграмма отражает основные назначения каждого компонента модуля client, полезна для ориентирования при фронтенд-разработки.

# Список использованных источников

1. Conda URL: <https://docs.conda.io> (дата обращения 15.03.2023)
2. Node v16.15.1 (LTS) URL: <https://nodejs.org/de/blog/release/v16.15.1> (дата обращения 15.03.2023)
3. Git URL: <https://git-scm.com/downloads> (дата обращения 15.03.2023)
4. Visual Studio Code URL: <https://code.visualstudio.com/> (дата обращения 15.03.2023)
5. PostgreSQL Downloads URL: <https://www.postgresql.org/download/> (дата обращения 15.03.2023)
6. Vue.js URL: <https://v3.ru.vuejs.org/> (дата обращения 15.03.2023)