

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

ФАКУЛЬТЕТ ИКТ

Отчет
по лабораторной работе №2
по курсу «Основы Web-программирования»
Тема:
**РЕАЛИЗАЦИЯ WEB-СЕРВИСОВ СРЕДСТВАМИ Django
REST framework, Vue.js, Muse-UI
Вариант 3**

Выполнила:
Артамонова Валерия
Студентка группы К3343

Проверил:
Говоров А. И.

Санкт-Петербург
2020

Цель работы: овладеть практическими навыками и умениями реализации web-сервисов средствами Django REST framework, Vue.js, Muse-UI.

Оборудование: компьютерный класс.

Программное обеспечение: Python 3.6, Django REST framework, Vue.js, Muse-UI (или аналогичная библиотека), PostgreSQL *.

Вариант 3

Создать программную систему, предназначенную для завуча школы. Она должна обеспечивать хранение сведений о каждом учителе, классном руководстве, о предметах, которые он преподает в заданный период, номере закрепленного за ним кабинета, о расписании занятий. Существуют учителя, которые не имеют собственного кабинета. Об учениках должны храниться следующие сведения: фамилия и имя, в каком классе учится, какую оценку имеет в текущей четверти по каждому предмету.

Завуч должен иметь возможность добавить сведения о новом учителе или ученике, внести в базу данных четвертные оценки учеников каждого класса по каждому предмету, удалить данные об уволившемся учителе и отчисленном из школы ученике, внести изменения в данные об учителях и учениках, в том числе поменять оценку ученика по тому или иному предмету. В задачу завуча входит также составление расписания.

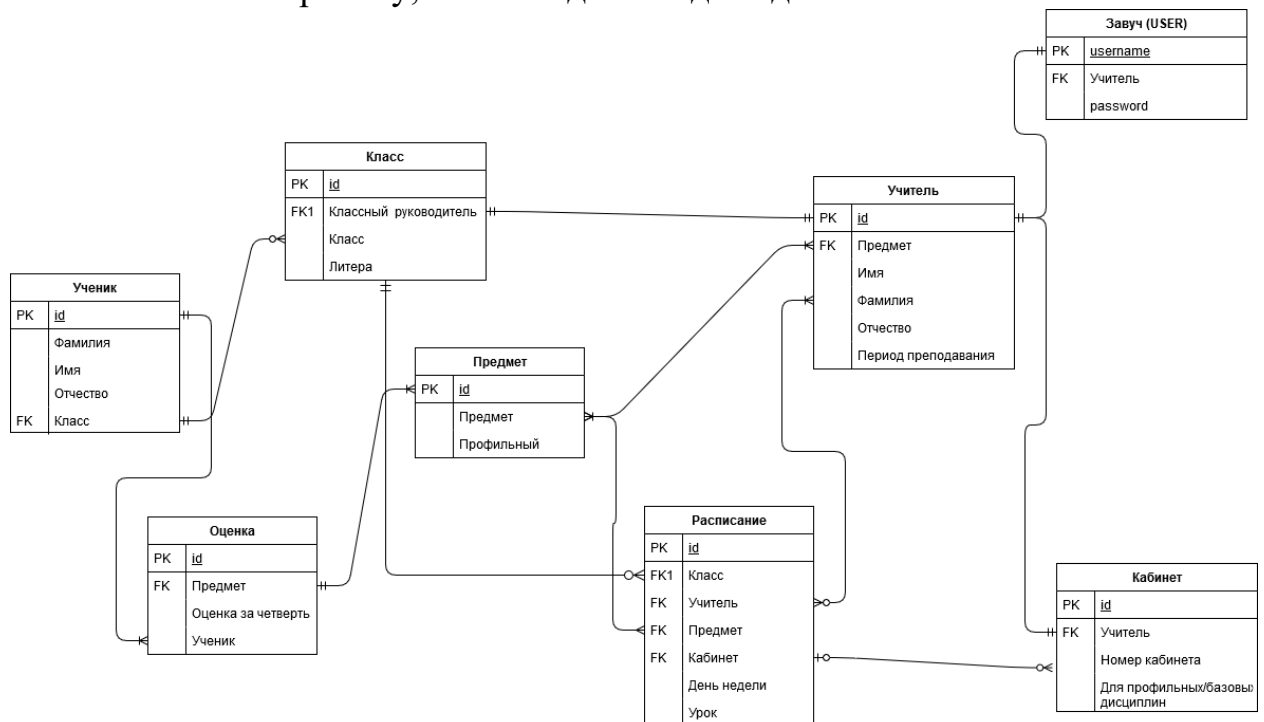
Завучу могут потребоваться следующие сведения:

- Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?
- Сколько учителей преподает каждую из дисциплин в школе?
- Список учителей, преподающих те же предметы, что и учитель, ведущий информатику в заданном классе.
- Сколько мальчиков и девочек в каждом классе?
- Сколько кабинетов в школе для базовых и профильных дисциплин?

Необходимо предусмотреть возможность получения документа, представляющего собой отчет об успеваемости заданного класса. Отчет включает сведения об успеваемости за четверть по каждому предмету. Необходимо подсчитать средний балл по каждому предмету, по классу в целом, указать общее количество учеников в классе. Для класса указать классного руководителя.

Ход работы:

1. Согласно варианту, была создана модель данных



2. Данная модель данных реализована в файле models.py

```
3. from django.db import models
from django.contrib.auth.models import User
```

```
class Subject(models.Model):
    subject = models.CharField("Предмет", max_length=100)
    profile_types = models.TextChoices('profile_types', 'Профильная дисциплина',
    Базовая дисциплина')
    profile = models.CharField("Тип предмета", blank=True, choices=profile_types.choices,
    max_length=100)
```

```
class Meta:
    verbose_name = "Предмет"
    verbose_name_plural = "Предметы"
```

```
def __str__(self):
    return self.subject
```

```
class Teacher(models.Model):
    last_name = models.CharField("Фамилия", max_length=50)
    first_name = models.CharField("Имя", max_length=50)
    second_name = models.CharField("Отчество", max_length=50)
    subject = models.ForeignKey(Subject, on_delete=models.CASCADE)
    teaching_period = models.DateField("Преполагает до")
```

```
class Meta:
    verbose_name = "Учитель"
    verbose_name_plural = "Учителя"
```

```
def __str__(self):
    first_name = self.first_name
    last_name = self.last_name
    second_name = self.second_name
```

```
teacher = last_name + " " + first_name + " " + second_name
return teacher
```

```
class UserProfile(models.Model):
    user_id = models.OneToOneField(User, on_delete=models.CASCADE)
    teacher_name = models.ForeignKey(Teacher, on_delete=models.CASCADE)
```

```
def __str__(self):
    return self.teacher_name
```

```
class Cabinet(models.Model):
    number = models.CharField("Номер кабинета", max_length=4)
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE, related_name='cabinet')
    profile_types = models.TextChoices('profile_types', 'Для_профильных_дисциплин'
    Для_базовых_дисциплин')
    profile = models.CharField("Тип кабинета", blank=True, choices=profile_types.choices,
    max_length=100)
```

```
class Meta:
    verbose_name = "Кабинет"
    verbose_name_plural = "Кабинеты"
```

```
def __str__(self):
    cabinet = self.number + " кабинет"
    return cabinet
```

```
class Klass(models.Model):
    number_types = models.TextChoices('number_types', '1 2 3 4 5 6 7 8 9 10 11')
    number = models.CharField("Класс", blank=True, choices=number_types.choices, max_length=2)
    litera_types = models.TextChoices('litera_types', 'А Б В Г Д Е')
    litera = models.CharField("Литера", blank=True, choices=litera_types.choices, max_length=2)
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE, related_name='klass')
```

```
class Meta:
    verbose_name = "Класс"
    verbose_name_plural = "Классы"
```

```
def __str__(self):
    klass = self.number + " " + self.litera
    return klass
```

```
class Pupil(models.Model):
    last_name = models.CharField("Фамилия", max_length=50)
    first_name = models.CharField("Имя", max_length=50)
    second_name = models.CharField("Отчество", max_length=50)
    gender_types = models.TextChoices('gender_types', 'Мужской Женский')
    gender = models.CharField("Пол", blank=True, choices=gender_types.choices, max_length=10)
    klass = models.ForeignKey(Klass, on_delete=models.CASCADE, related_name='pupils')
```

```
class Meta:
    verbose_name = "Ученик"
    verbose_name_plural = "Ученики"
```

```
def __str__(self):
    pupil = self.last_name + " " + self.first_name + " " + self.second_name
    return pupil
```

```

class Grade(models.Model):
    student = models.ForeignKey(Pupil, on_delete=models.CASCADE, related_name="grades")
    subject = models.ForeignKey(Subject, on_delete=models.CASCADE)
    grade_types = models.TextChoices('grade_types', '2 3 4 5')
    grade = models.CharField("Оценка за четверть", blank=True, choices=grade_types.choices,
max_length=2)

    class Meta:
        unique_together = [
            ("student", "subject", "grade")
        ]
        verbose_name = "Четвертная оценка"
        verbose_name_plural = "Четвертные оценки"

    def __str__(self):
        grade = self.grade
        return grade

class Timetable(models.Model):
    klass_name = models.ForeignKey(Klass, on_delete=models.CASCADE, related_name="timetable")
    lesson_number = models.TextChoices('lesson_number', '1-8:00-8:45 2-8:50-9:35 3-9:40-10:25 4-10:40-11:25 5-11:30-12:15 6-12:20-13:05 7-13:05-13:50 8-14:00-14:45')
    lesson = models.CharField("Урок", blank=True, choices=lesson_number.choices, max_length=50)
    choose_day = models.TextChoices('choose_day', 'Понедельник Вторник Среда Четверг Пятница Суббота')
    day = models.CharField("День недели", blank=True, choices=choose_day.choices,
max_length=100)
    subject_name = models.ForeignKey(Subject, verbose_name="Предмет",
on_delete=models.CASCADE)
    teacher_name = models.ForeignKey(Teacher, verbose_name="Учитель",
on_delete=models.CASCADE)
    cabinet_number = models.ForeignKey(Cabinet, verbose_name="Кабинет",
on_delete=models.CASCADE)

    class Meta:
        unique_together = [
            ("klass_name", "lesson", "day")
        ]
        verbose_name = "Расписание"
        verbose_name_plural = "Расписание"

    def __str__(self):
        timetable = self.day + " " + self.lesson
        return timetable

```

4. Для создания отображений была использована абстракция ViewSet

```

5. from rest_framework import generics, permissions, viewsets, renderers
from django_filters.rest_framework import DjangoFilterBackend
from .service import TimetableFilter

from .models import Teacher, Timetable, Klass, Pupil, Cabinet, Subject, Grade
from .serializers import (TeacherSerializer, TeacherDetailSerializer, TeacherAddSerializer,
PupilSerializer,
PupilDetailSerializer, GradeCreateSerializer, PupilAddSerializer, TimetableSerializer,
TimetableAddSerializer, KlassSerializer, SubjectSerializer, CabinetSerializer,
KlassDetailSerializer, KlassAddSerializer, GradeSerializer)

class TeacherViewSet(viewsets.ModelViewSet):
    """CRUD для модели Учитель"""
    queryset = Teacher.objects.all()

```

```
def get_serializer_class(self):
    if self.action == 'list':
        return TeacherSerializer
    elif self.action == 'update':
        return TeacherDetailSerializer
    elif self.action == 'create':
        return TeacherAddSerializer
    elif self.action != 'list':
        return TeacherDetailSerializer
```

```
class PupilViewSet(viewsets.ModelViewSet):
    """CRUD для модели Ученик"""
    queryset = Pupil.objects.all()
```

```
def get_serializer_class(self):
    if self.action == 'list':
        return PupilSerializer
    elif self.action == 'update':
        return PupilSerializer
    elif self.action == 'create':
        return PupilAddSerializer
    elif self.action != 'list':
        return PupilDetailSerializer
```

```
class GradeViewSet(viewsets.ModelViewSet):
    """CRUD для модели Оценка"""
    queryset = Grade.objects.all()
```

```
def get_serializer_class(self):
    if self.action == 'create':
        return GradeCreateSerializer
    elif self.action != 'create':
        return GradeSerializer
```

```
class TimetableViewSet(viewsets.ModelViewSet):
    """CRUD для модели Расписание"""
    queryset = Timetable.objects.all()
    filter_backends = (DjangoFilterBackend,
                       )
    filterset_class = TimetableFilter
```

```
def get_serializer_class(self):
    if self.action == 'list':
        return TimetableSerializer
    elif self.action == 'retrieve':
        return TimetableSerializer
    elif self.action == 'update':
        return TimetableSerializer
    elif self.action == 'create':
        return TimetableAddSerializer
```

```
class KlassViewSet(viewsets.ModelViewSet):
    """CRUD для модели Класс"""
    queryset = Klass.objects.all()
```

```
def get_serializer_class(self):
    if self.action == 'list':
```

```
        return KlassSerializer
    elif self.action == 'create':
        return KlassAddSerializer
    elif self.action != 'list':
        return KlassDetailSerializer
```

```
class CabinetViewSet(viewsets.ModelViewSet):
    """Отображение для модели Кабинет"""
    queryset = Cabinet.objects.all()
    serializer_class = CabinetSerializer
```

```
class SubjectViewSet(viewsets.ModelViewSet):
    """Отображение для модели Предмет"""
    queryset = Subject.objects.all()
    serializer_class = SubjectSerializer
```

6. Полученные интерфейсы в панели Django Admin

Администрирование Django

Администрирование сайта

SCHOOL		
User profiles	+ Добавить	✎ Изменить
Кабинеты	+ Добавить	✎ Изменить
Классы	+ Добавить	✎ Изменить
Предметы	+ Добавить	✎ Изменить
Расписание	+ Добавить	✎ Изменить
Ученики	+ Добавить	✎ Изменить
Учителя	+ Добавить	✎ Изменить
Четвертные оценки	+ Добавить	✎ Изменить
ПОЛЬЗОВАТЕЛИ И ГРУППЫ		
Группы	+ Добавить	✎ Изменить
Пользователи	+ Добавить	✎ Изменить
ТОКЕН АУТЕНТИФИКАЦИИ		
Токены	+ Добавить	✎ Изменить

а. Создание нового Кабинета

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало » School » Кабинеты

Выберите Кабинет для изменения

ДОБАВИТЬ КАБИНЕТ +

Действие:

▼

Выполнить

Выбрано 0 объектов из 1

☐ КАБИНЕТ

☐ 1 кабинет

1 Кабинет

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало » School » Кабинеты » Добавить Кабинет

Добавить Кабинет

Номер кабинета:

2

Teacher:

Параничева Любовь Александровна

▼

✎

+

Тип кабинета:

Для Профильных Дисциплин

▼

СОХРАНИТЬ

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

б. Создание нового Класса

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > School > Классы

Выберите Класс для изменения

ДОБАВИТЬ КЛАСС +

Действие:

Выполнить

Выбрано 0 объектов из 4

☐ КЛАСС

☐ 3 Д

☐ 7 В

☐ 5 Б

☐ 11 А

4 Классы

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > School > Классы > Добавить Класс

Добавить Класс

Класс:

8

Литера:

Е

Teacher:

Заостровская Ольга Анновна

СОХРАНИТЬ

с. Создание нового Предмета

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > School > Предметы

Выберите Предмет для изменения

ДОБАВИТЬ ПРЕДМЕТ +

Действие:

Выполнить

Выбрано 0 объектов из 9

☐ ПРЕДМЕТ

☐ Физическая культура

☐ История

☐ Рисование

☐ Химия

☐ Русская литература

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > School > Предметы > Добавить Предмет

Добавить Предмет

Предмет:

Информатика

Тип предмета:

Профильная Дисциплина

СОХРАНИТЬ

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

д. Создание нового Расписания

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > School > Расписание > Добавить Расписание

Добавить Расписание

Klass name:

11 А

Урок:

2-8:50-9:35

День недели:

Вторник

Предмет:

Химия

Учитель:

Пак Елена Иннокентьевна

Кабинет:

1 кабинет

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > School > Расписание

✔ The Расписание "Вторник 2-8:50-9:35" was added successfully.

Выберите Расписание для изменения

ДОБАВИТЬ РАСПИСАНИЕ +

Действие:

Выполнить

Выбрано 0 объектов из 5

☐ РАСПИСАНИЕ

☐ Вторник 2-8:50-9:35

☐ Понедельник 1-8:00-8:45

☐ Вторник 1-8:00-8:45

☐ Понедельник 2-8:50-9:35

☐ Понедельник 1-8:00-8:45

5 Расписание

е. Добавление нового Ученика

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > School > Ученики

Выберите Ученик для изменения

ДОБАВИТЬ УЧЕНИК +

Действие:

Выполнить

Выбрано 0 объектов из 5

☐

УЧЕНИК

☐

Царев Иван Дмитриевич

☐

Цветиков Антон Борисович

☐

Артамонова Валерия Евгеньевна

☐

Дробышев Олег Николаевич

☐

Матченко Анастасия Александровна

5 Ученики

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > School > Ученики > Добавить Ученик

Добавить Ученик

Фамилия:

Будкевич

Имя:

Полина

Отчество:

Геннадьевна

Пол:

Женский

Klass:

8 E

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ

f. Добавление нового Учителя

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > School > Учителя

Выберите Учителя для изменения

ДОБАВИТЬ УЧИТЕЛЬ +

Действие:

Выполнить

Выбрано 0 объектов из 9

☐

УЧИТЕЛЬ

☐

Ергешев Сергей Сабирович

☐

Румянцева Татьяна Леонидовна

☐

Заостровская Ольга Анновна

☐

Пак Елена Иннокентьевна

☐

Иванов Иван Иванович

☐

Версаль Марина Григорьевна

☐

Параничева Любовь Александровна

☐

Гамель Елена Викторовна

☐

Брылевская Лариса Ивановна

9 Учителя

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > School > Учителя > Добавить Учителя

Добавить Учителя

Фамилия:

Попов

Имя:

Алексей

Отчество:

Сергеевич

Subject:

Информатика

Преподает до:

02.06.2028

Сегодня

Внимание: Ваше локальное время опережает время сервера на 3 часа.

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ

g. Добавление новой Четвертной оценки

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > School > Четвертные оценки

Выберите Четвертная оценка для изменения

ДОБАВИТЬ ЧЕТВЕРТНАЯ ОЦЕНКА +

Действие:

Выполнить

Выбрано 0 объектов из 3

☐

ЧЕТВЕРТНАЯ ОЦЕНКА

☐

5

☐

4

☐

5

3 Четвертные оценки

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, USER. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > School > Четвертные оценки > Добавить Четвертная оценка

Добавить Четвертная оценка

Student:

Артамонова Валерия Евгеньевна

Subject:

Информатика

Оценка за четверть:

5

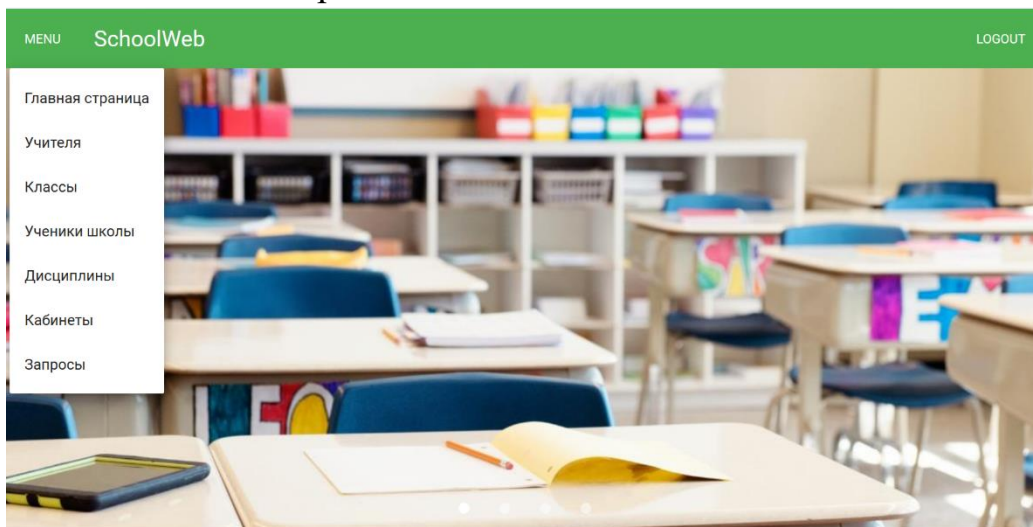
Сохранить и добавить другой объект

Сохранить и продолжить редактирование

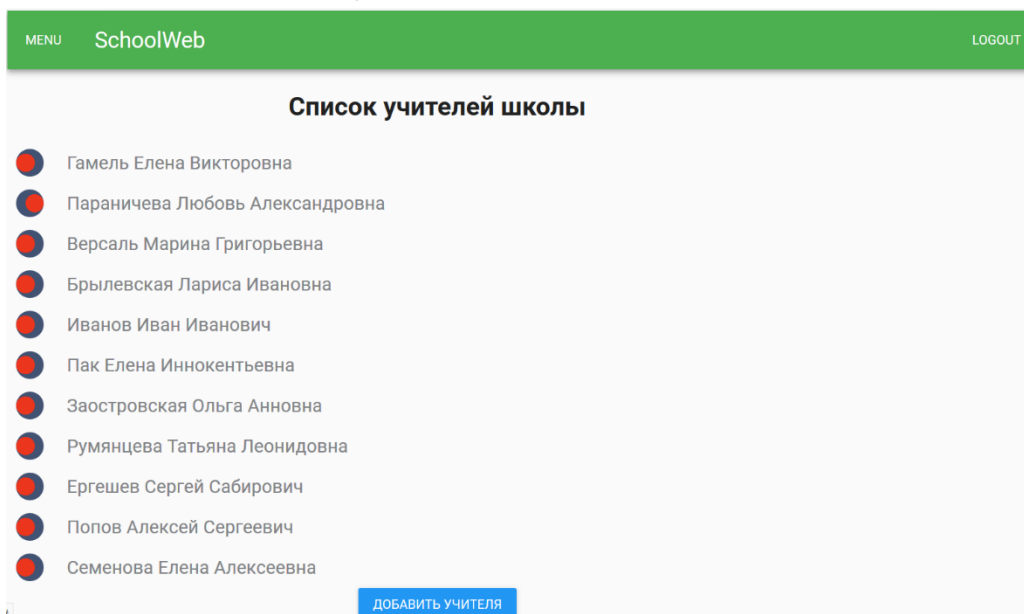
СОХРАНИТЬ

7. Созданные интерфейсы во Vue JS

а. Главная страница



б. Список всех учителей школы



с. Добавление нового учителя

Фамилия	Синицына
Имя	Марина
Отчество	Ивановна
Предмет	Рисование
Преподает	2024-07-30
до Год- Месяц-День	
<button>Submit</button>	

d. Досье учителя

MENU

SchoolWeb

LOGOUT

Досье учителя

Фамилия	Брылевская
Имя	Лариса
Отчество	Ивановна
Предмет	Математика
Имеет классное руководство	11 "А"
Имеет закрепленный за собой кабинет	Кабинет №1
Преподает до	2022-12-25

ИЗМЕНИТЬ ДАННЫЕ ОБ УЧИТЕЛЕ

УДАЛИТЬ УЧИТЕЛЯ

e. Изменение данных об учителе

MENU

SchoolWeb

LOGOUT

Досье учителя

Фамилия	Брылевская Семенова
Имя	Лариса Лариса
Отчество	Ивановна Ивановна
Предмет	Математика
Имеет классное руководство	11 "А"
Имеет закрепленный за собой кабинет	Кабинет №1
Преподает до	2022-12-25 2026-07-25

ВНЕСТИ ИЗМЕНЕНИЯ

УДАЛИТЬ УЧИТЕЛЯ

f. Список классов

MENU

SchoolWeb

LOGOUT

Список классов

11 "А"

ПОКАЗАТЬ УЧЕНИКОВ

ПОКАЗАТЬ РАСПИСАНИЕ

ПОЛУЧИТЬ ОТЧЕТ ПО КЛАССУ

5 "Б"

ПОКАЗАТЬ УЧЕНИКОВ

ПОКАЗАТЬ РАСПИСАНИЕ

ПОЛУЧИТЬ ОТЧЕТ ПО КЛАССУ

7 "В"

ПОКАЗАТЬ УЧЕНИКОВ

ПОКАЗАТЬ РАСПИСАНИЕ

ПОЛУЧИТЬ ОТЧЕТ ПО КЛАССУ

3 "Д"

ПОКАЗАТЬ УЧЕНИКОВ

ПОКАЗАТЬ РАСПИСАНИЕ

ПОЛУЧИТЬ ОТЧЕТ ПО КЛАССУ

8 "Е"

ПОКАЗАТЬ УЧЕНИКОВ

ПОКАЗАТЬ РАСПИСАНИЕ

ПОЛУЧИТЬ ОТЧЕТ ПО КЛАССУ

g. Список учеников выбранного класса

MENU

SchoolWeb

LOGOUT

Список учеников 11 "А" класса

Матченко Анастасия Александровна

Дробышев Олег Николаевич

Артамонова Валерия Евгеньевна

Цветиков Антон Борисович

Чочуева Марьям Мухтарбековна

ДОБАВИТЬ УЧЕНИКА

h. Добавление нового ученика

Добавление нового ученика

Фамилия

Степанов

Имя

Артем

Отчество

Геннадьевич

Пол

☒ Мужской ☐ Женский

Класс

11А

Submit

i. Список всех учеников школы

MENU

SchoolWeb

LOGOUT

Список учеников школы

Артамонова Валерия Евгеньевна

Матченко Анастасия Александровна

Цветиков Антон Борисович

Будкевич Полина Геннадьевна

Иванова Иванка Ивановна

Red Hot Chili Peppers

Дробышев Олег Николаевич

Чочуева Марьям Мухтарбековна

Степанов Артем Геннадьевич

ДОБАВИТЬ УЧЕНИКА

j. Досье ученика

MENU

SchoolWeb

LOGOUT

Досье ученика

Фамилия	Матченко
Имя	Анастасия
Отчество	Александровна
Пол	Женский
Класс	11

Оценки за четверть

ДОБАВИТЬ ОЦЕНКУ

ИЗМЕНИТЬ ОЦЕНКУ

УДАЛИТЬ ОДНУ ИЗ ОЦЕНОК

Математика	5
Русский язык	4
История	4
Химия	5

ИЗМЕНИТЬ ДАННЫЕ ОБ УЧЕНИКЕ

УДАЛИТЬ УЧЕНИКА

k. Изменение данных об ученике

MENU

SchoolWeb

LOGOUT

Досье ученика

Фамилия	Матченко Дробышева
Имя	Анастасия Анастасия
Отчество	Александровна Александровна
Пол	Женский
Класс	11

Оценки за четверть

ДОБАВИТЬ ОЦЕНКУ

ИЗМЕНИТЬ ОЦЕНКУ

УДАЛИТЬ ОДНУ ИЗ ОЦЕНОК

Математика	5
Русский язык	4
История	4
Химия	5

ВНЕСТИ ИЗМЕНЕНИЯ

УДАЛИТЬ УЧЕНИКА

1. Список дисциплин, преподаваемых в школе

MENU

SchoolWeb

LOGOUT

Список дисциплин, преподаваемых в школе

Математика

Русский язык

Английский язык

Физика

Русская литература

Химия

Рисование

История

Физическая культура

Информатика

m. Список кабинетов школы

MENU

SchoolWeb

LOGOUT

Список кабинетов школы

Кабинет №1

Кабинет №2

n. Вход

MENU

SchoolWeb

LOGIN

UserName

user

A character with a username of 6-12 length

Password

••••

Please enter password

ВОЙТИ

Нет логина и пароля? Тогда Вам необходимо [зарегистрироваться](#)

o. Регистрация

MENU

SchoolWeb

LOGIN

Пожалуйста, заполните следующие поля

Поля должны содержать не менее 8 символов.

99879vjgv

Password

••••••••visibility

Password again

••••••••visibility

SIGN UP

Вывод: в ходе выполнения лабораторной работы были получены навыки создания web-приложения с помощью web-фреймворка Django REST языка программирования Python, web-фреймворка Vue языка программирования JavaScript и JS Muse-UI.