**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ**

Государственное бюджетное общеобразовательное учреждение города Москвы

**ГБОУ Школа № 1530 «Школа Ломоносова»**

Индивидуальный проект по информатике

на тему: **«Создание электронного журнала для репетиторов»**

**Автор:** Вдовина Марина Максимовна 10 И

**Руководитель:** Журкина Юлия Сергеевна

Учитель информатики

Работа допущена к защите «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Подпись руководителя проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**Москва**

**2023**

Оглавление

[Введение 3](#_Toc157806854)

[Актуальность 3](#_Toc157806855)

[Проблема 3](#_Toc157806856)

[Цель 3](#_Toc157806857)

[Задачи 3](#_Toc157806858)

[Теоретическая часть 4](#_Toc157806859)

[Основная информация 4](#_Toc157806860)

[Принцип SOLID 4](#_Toc157806861)

[1. Принцип единственной ответственности 5](#_Toc157806862)

[2. Принцип открытости/закрытости 5](#_Toc157806863)

[3. Принцип подстановки Барбары Лисков 5](#_Toc157806864)

[4. Принцип разделения интерфейса 5](#_Toc157806865)

[5. Принцип инверсии зависимостей 5](#_Toc157806866)

[Фреймворк Bootstrap 6](#_Toc157806867)

[Немного о библиотеках в Python 6](#_Toc157806868)

[Что такое библиотека в Python? 6](#_Toc157806869)

[Flask 6](#_Toc157806870)

[SQLAlchemy 7](#_Toc157806871)

[Requests 7](#_Toc157806872)

[REST API 8](#_Toc157806873)

[Что это такое? 8](#_Toc157806874)

[Архитектура 8](#_Toc157806875)

[Практическая часть 9](#_Toc157806876)

[Заключение 11](#_Toc157806877)

[Список литературы и интернет-источников 12](#_Toc157806878)

# Введение

## Актуальность

На сегодняшний день репетиторство является очень популярной и востребованной отраслью в получении как школьного, так и среднего специального, а также высшего образования.

Чтобы сделать процесс обучения удобным и плодотворным, репетитор должен самостоятельно разрабатывать методики преподавания, составлять наиболее удобный для обеих сторон график занятий, искать материалы по разным темам. Но для всех этих целей нет хорошего и бесплатного приложения, в котором сочетались бы все эти функции.

## Проблема

Репетиторы вынуждены пользоваться примитивными способами (делать вручную или пользоваться большим количеством приложений), чтобы выкладывать домашнее задание, проверять его, ставить оценки ученикам, структурировать свой график. Поэтому я считаю, что этот проект может помочь многим людям: и репетиторам, и их ученикам – в их деятельности.

## Цель

Создать электронный журнал для репетиторов, который поможет преподавателям и их ученикам.

## Задачи

1. Создать базу данных с необходимыми таблицами.
2. Использовать REST API для изменения, получения и добавления данных.
3. Реализовать авторизацию и регистрацию.
4. Использовать Bootstrap.
5. Применить принцип SOLID.

# Теоретическая часть

## Основная информация

1. Вся программа написана с помощью трёх языков: Python (основной код), HTML (создание шаблонов), CSS (создание дизайна). Также используются библиотеки Python: flask, flask\_login, requests, json, sqlalchemy, werkzeug.security, calendar, datetime.
2. В проекте есть работа с базой данных (добавление, редактирование и удаление информации) с помощью REST API.
3. Код написан по принципу SOLID.
4. Для создания интерфейса используется фреймворк Bootstrap.

## Принцип SOLID

SOLID — это аббревиатура, которая описывает пять ключевых принципов проектирования и организации объектно-ориентированного кода. Эти принципы направлены на написание чистого, читабельного и сопровождаемого кода.

Вот как расшифровывается SOLID:

* S: Single Responsibility Principle (Принцип единственной ответственности).
* O: Open-Closed Principle (Принцип открытости/закрытости).
* L: Liskov Substitution Principle (Принцип подстановки Барбары Лисков).
* I: Interface Segregation Principle (Принцип разделения интерфейса).
* D: Dependency Inversion Principle (Принцип инверсии зависимостей).

Рассмотрим каждый из принципов более подробно.

### Принцип единственной ответственности

Класс, модуль или функция должны решать только одну конкретную задачу и иметь ограниченную область ответственности.

### Принцип открытости/закрытости

Классы, модули и функции должны быть открыты для расширения, но закрыты для изменения. Это означает, что мы должны иметь возможность создавать новые модели поведения без изменения существующих.

### Принцип подстановки Барбары Лисков

Наследующий класс должен дополнять, а не изменять базовый. Это означает то, что класс наследник не должен кардинально изменять поведение предка, он должен лишь расширять базовый функционал.

### Принцип разделения интерфейса

Этот принцип говорит о том, что "клиенты не должны зависеть от интерфейсов, которые они не используют". Это означает, что объёмный интерфейс должен быть разбит на несколько меньших и более конкретных, чтобы клиенты могли использовать только то, что им нужно.

### Принцип инверсии зависимостей

Классы должны зависеть от абстракций, а не от деталей. Высокоуровневые модули не должны зависеть от низкоуровневых модулей, то есть должна быть чёткая иерархия классов. Оба типа модулей должны зависеть от абстракций.

Таким образом, SOLID — это набор пяти принципов, которые делают код более эффективным, устойчивым и поддерживаемым. Эти принципы направлены на уменьшение зависимостей в коде, облегчение поддержки и упрощение процесса внесения изменений. Важно помнить, что они наиболее эффективны, когда используются вместе.

## Фреймворк Bootstrap

Bootstrap — это **бесплатный CSS-фреймворк с открытым исходным кодом,** предназначенный для быстрой вёрстки адаптивных интерфейсов сайтов и веб-приложений. Bootstrap используется на 19% всех веб-сайтов мира. Его код доступен на GitHub.

В целом, Bootstrap — это набор файлов. После их подключения к странице для верстки станет доступно большое количество готовых компонентов и классов. Классы в Bootstrap делятся на 3 большие группы: 1) для создания сетки — адаптивного макета страницы; 2) для стилизации контента — текста, изображений и т. д.; 3) служебные — для решения вспомогательных задач (например, отображение и скрытие элементов, выравнивание текста на странице и т. д.). Кроме классов, в Bootstrap есть готовые объекты интерфейса (кнопки, выпадающие списки и подсказки, формы, навигационные меню).

## Немного о библиотеках в Python

### Что такое библиотека в Python?

Библиотеки в Python представляют собой наборы связанных модулей. Они содержат пакеты кода, которые можно многократно использовать в разных программах. Это делает программирование на Python более простым и удобным, так как нам не нужно писать один и тот же код снова и снова для разных программ. Принцип модульной организации кода в Python выполняет важную роль в сфере машинного обучения, визуализации данных и т. д.

Далее я рассмотрю несколько основных библиотек, используемых в проекте.

### Flask

Flask — это легковесный веб-фреймворк, который предоставляет минимальный набор инструментов для создания веб-приложений. На нём можно написать как очень простой сайт, так и огромное веб-приложение с большим количеством сервисов.

У Flask много преимуществ, которые выделяют его среди других фреймворков:

1. удобные шаблоны — можно быстро создавать прототипы веб-приложений;
2. перечень инструментов для гибкой настройки сайтов под любые нужды.

### SQLAlchemy

SQLAlchemy — это Python-библиотека, которая позволяет работать с реляционными базами данных с помощью ORM. Реляционные базы данных хранят информацию в виде связанных между собой таблиц.

ORM расшифровывается как object-relational mapping, или «объектно-реляционное отображение». ORM позволяет управлять базами данных с помощью методов объектов в коде и при этом не использовать SQL-запросы.

### Requests

Requests – это клиентская библиотека для Python для HTTP-запросов, которая предоставляет встроенные функциональные возможности для управления как запросом, так и ответом.

## REST API

### Что это такое?

REST API – это способ взаимодействия сайтов и веб-приложений с сервером. Технологию REST API применяют везде, где пользователю сайта или веб-приложения нужно предоставить данные с сервера.

API (Application Programming Interface — программный интерфейс приложения) — это код, который позволяет двум приложениям обмениваться данными с сервера.

REST (REpresentational State Transfer – передача состояния представления)— это способ создания API с помощью протокола HTTP.

### Архитектура

REST API основывается на протоколе передачи гипертекста HTTP (стандартный протокол в интернете, созданный для передачи гипертекста). Каждый объект на сервере в HTTP имеет свой уникальный URL-адрес в строгом последовательном формате.

В REST API есть 4 метода HTTP, которые используют для действий с объектами на серверах:

1. GET (получение информации о данных)
2. DELETE (удаление данных)
3. POST (добавление данных)
4. PUT (редактирование данных)

В каждом HTTP-запросе есть заголовок, за которым следует описание объекта на сервере — это и есть его состояние.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

# Практическая часть

С самого начала была поставлена задача реализовать следующие модули:

1. Создать три учётные записи: репетитор, ученик, родитель.
2. Для каждого пользователя главная страница – это расписание на текущий месяц(календарь), пользователю нужно кликать на определённый день и смотреть, какие уроки у него будут в этот день и в какое время. Каждый пользователь должен иметь доступ к видеоурокам, тестам и справочным материалам по разным предметам.
3. Для репетитора нужно реализовать следующее:
4. добавление учеников (ученик добавляется с помощью уникального id, который есть у каждого пользователя);
5. добавление нового занятия (репетитор сам может придумать название и назначить дни и время занятий);
6. прикрепление учеников к одному занятию, если они занимаются группами;
7. формирование отчётов об успеваемости учеников для каждого курса (диаграмма);
8. возможность задавать домашнее задание, проверять его и выставлять баллы;
9. возможность редактирования своего профиля (информации о себе).
10. Для ученика нужно реализовать следующее:
11. возможность смотреть свои баллы по курсу и сравнивать их с баллами одногруппников;
12. возможность смотреть ДЗ и изменять его статус (выполнено/не выполнено), а также видеть баллы за него и комментарии учителя;
13. возможность просмотра профилей репетиторов, комментирования их, выставления оценок;
14. возможность добавления репетиторов и родителей (реализуется с помощью уникального id, который есть у каждого пользователя).
15. Функционал родительского аккаунта включает в себя:
16. возможность следить за занятиями ребёнка;
17. возможность видеть домашнее задание, баллы за него у своих детей;
18. возможность добавления нескольких детей (реализовано с помощью уникального id, который есть у каждого пользователя).
19. Создать регистрацию и авторизацию на сайте для пользователей. Каждый пароль должен кэшироваться и в таком виде храниться в базе данных.
20. Создать базу данных tutorcoon с таблицами: users, tutors, students, parents, lessons, homeworks, tests, help\_materials, video\_lessons.
21. Спроектировать изображения профиля для трёх учётных записей, а также эмблему сайта.
22. Связать внешний вид некоторых элементов сайта с енотами.

# Заключение

Все поставленные задачи выполнены. В дальнейшем планируется разместить сайт на платформе, приобрести доменное имя и расширить функционал в соответствии со следующими пунктами:

1. добавить подсчёт доходов;
2. оптимизировать сайт для браузеров мобильных устройств;
3. вести рейтинг репетиторов.

А также нужно привлекать как можно больше пользователей. В дальнейшей перспективе планируется воспользоваться услугами агентств для раскрутки сайта.

# Список литературы и интернет-источников

1. URL: <https://academy.yandex.ru>
2. URL: [CSS: Cascading Style Sheets | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Web/CSS)
3. Лутц Марк. Изучаем Python. / Лутц Марк. - Москва: Диалектика Вильямс, 2020. С. - 720.
4. URL: [Bootstrap · The most popular HTML, CSS, and JS library in the world. (getbootstrap.com)](https://getbootstrap.com/)