**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](#_Toc151323622)

[Актуальность 3](#_Toc151323623)

[Проблема 3](#_Toc151323624)

[Цель 3](#_Toc151323625)

[Задачи 3](#_Toc151323626)

[Теоретическая часть 4](#_Toc151323627)

[Основная информация 4](#_Toc151323628)

[Принцип SOLID 4](#_Toc151323629)

[1. Принцип единственной ответственности 5](#_Toc151323630)

[2. Принцип открытости/закрытости 5](#_Toc151323631)

[3. Принцип подстановки Барбары Лисков 5](#_Toc151323632)

[4. Принцип разделения интерфейса 5](#_Toc151323633)

[5. Принцип инверсии зависимостей 5](#_Toc151323634)

[Фреймворк Bootstrap 6](#_Toc151323635)

[REST API 6](#_Toc151323636)

[Что это такое? 6](#_Toc151323637)

[Архитектура 6](#_Toc151323638)

[Практическая часть 8](#_Toc151323639)

[Заключение 10](#_Toc151323640)

[Список литературы и интернет-источников 11](#_Toc151323641)

# Введение

## Актуальность

Занятия с репетитором сейчас – очень продвинутый и часто встречающийся метод обучения. Сейчас репетиторов становится всё больше и больше. Но для всех их нужд нет хорошего бесплатного приложения.

## Проблема

Репетиторы вынуждены пользоваться примитивными способами, чтобы задавать ДЗ, 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 таблицами: users, tutors, students, parents, lessons, homeworks, tests, help\_materials, video\_lessons.
2. Использовать REST API для изменения, получения и добавления данных.
3. Сделать 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).

## 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 (программный интерфейс приложения) — это код, который позволяет двум приложениям обмениваться данными с сервера.

REST (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 (сделано/не сделано), а также видеть баллы за него и комментарии учителя
13. возможность просмотра профилей репетиторов, комментирования их, выставления оценок
14. возможность добавления репетиторов и родителей (они добавляются с помощью уникального id, который есть у каждого пользователя)
15. Для родителя нужно реализовать следующее:
16. возможность следить за занятиями ребёнка
17. возможность видеть домашнее задание, баллы за него для своих детей
18. возможность добавления нескольких детей (они добавляются 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. Важно держать данные зашифровано!
20. Создать базу данных “tutorcoon” с таблицами: users, tutors, students, parents, lessons, homeworks, tests, help\_materials, video\_lessons.
21. Нарисовать аватарки для всех трёх учётных записей, а также эмблему сайта.
22. Использовать енотов для украшения всего интерфейса.

# Заключение

Все поставленные задачи выполнены. В дальнейшем планируется выложить сайт в интернет, расширить спектр его возможностей (добавить подсчёт доходов, сделать мобильную версию), а также привлечь как можно больше пользователей.

# 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/)