**Электронный журнал с продвинутыми функциями**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Кемерово, 2020

# Оглавление

[Оглавление 2](#_Toc34666255)

[Введение 3](#_Toc34666256)

[Актуальность 4](#_Toc34666257)

[Теоретическая часть 5](#_Toc34666258)

[Сайт. Структура сайта. 5](#_Toc34666259)

[Основные составляющие сайта 5](#_Toc34666260)

[История развития языка разметки HTML 6](#_Toc34666261)

[Версии 8](#_Toc34666262)

[История развития языка стилей CSS 9](#_Toc34666263)

[История развития языка программирования Python 11](#_Toc34666264)

[История развития библиотеки для создания сайтов Django 13](#_Toc34666265)

[Практическая часть 15](#_Toc34666266)

[Заключение 17](#_Toc34666267)

[Список литературы 18](#_Toc34666268)

[Приложение 19](#_Toc34666269)

# Введение

В настоящее время технологии окружают нас повсюду - ни одна сфера человеческой жизни не обходится без компьютеров и интернета. Эта тенденция не обошла стороной и образовательные учреждения. Сейчас даже стоит вопрос о роботизации такой профессии, как учитель, что станет новой эрой в этой области. Стоит заметить, что далеко не все преподаватели готовы привыкнуть к современным технологиям. До сих пор многие из них пользуются уже давно устаревшими и неактуальными средствами контроля успеваемости учащихся, но это не практично, поэтому сейчас создаются сервисы, позволяющие хранить эти данные во всемирной сети Интернет.

Удобств у этих сервисов масса:

* Доступность журнала/дневника из любой точки мира, где есть интернет
* Возможность контроля успеваемости на автоматическом уровне, в том числе, доступ к перечню всех оценок, и автоматический подсчет среднего балла
* Получение актуальной информации по изменениям в расписании
* Иметь возможность просмотра заданного Д/З самим учителем

На создание нашего проекта нас натолкнули размышления, которые привели к выводу о том, что электронный журнал не доведен до совершенства и не всегда удобен в использовании, а некоторых функций, которые облегчат работу учителей, родителей и учащихся Городского Классического Лицея вовсе нет. Вследствие всего вышеперечисленного зародилась идея о написании собственного электронного журнала с расширенным функционалом.

**В состав его функций будут входить как основные функции дневника:**

* Выставление отметок
* Выставление меток посещаемости
* Возможность отправки сообщений ученикам
* Расписание

**Так и дополнительные:**

* Возможность рассылки, как домашнего задания, так и различных заметок и уточнений, по поводу образовательного процесса.
* Возможность сдать Дом.Задание прямо в журнале (электронно)
* Выставление меток за сдачу тетради.
* Общение с сотрудниками образовательного учреждения (школы)

Мы гарантируем, что с нашим электронным журналом не будет возникать таких проблем, как:

1. Споры с учителем по поводу заданного домашнего задания.
2. Споры по поводу выполненного домашнего задания
3. Споры по поводу отсутствия предупреждений в изменении расписания
4. Отсутствия предупреждения об итоговых оценках и плохом среднем балле и многое другое

# Актуальность

Мы считаем, что наш проект актуальный на сегодняшний день, т.к. обладает множеством функций:

* Выставление отметок.   
  Главный фактор контроля знаний ученика, позволяющий дать оценку за домашние задания, работу ученика на уроке и разного рода проверочные, самостоятельные, а также контрольные работы.
* Выставление меток посещаемости.

Контроль посещаемости позволяет законным представителям ученика, отследить заинтересованность учащегося в образовательном процессе, и в случае необходимости принять определенные меры.

* Возможность отправки сообщений ученикам.

Общение преподавателя с учащимися в плане образовательного процесса, значительно его облегчит, и такая проблема, как недопонимание между учащимся и преподавателем вовсе исчезнет.

* Просмотр расписания

Это удобно не только в плане носителя, но и в случае каких - либо изменений, учащийся будет осведомлен об этом.

* Возможность рассылки важной (и не очень) информации
* Возможность сдать домашнее задание прямо в журнале (электронно, т.е., написать решение, либо прикрепить фотографию тетради с выполненным Д/З)
* Выставление меток за сдачу тетради с Д/З (Если оно, конечно, не было сдано электронно)
* Общение с сотрудниками образовательного учреждения (школы)
* Возможность отслеживать успеваемость путем просмотра оценок и среднего балла.

# Теоретическая часть

## Сайт. Структура сайта.

**Сайт**, или **веб-сайт** (от англ. **website**: **web** — «паутина, сеть» и **site** — «место», буквально **«место, сегмент, часть в сети»**), — одна или несколько логически связанных между собой веб-страниц; также место расположения контента сервера. Обычно сайт в Интернете представляет собой совокупность связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое. Веб-сайты называются так, потому что доступ к ним происходит по протоколу HTTP.

**Веб-сайт**, как система электронных документов (файлов данных и кода) может принадлежать частному лицу или организации и быть доступным в компьютерной сети под общим доменным именем и IP-адресом или локально на одном компьютере. С точки зрения авторского права сайт является составным произведением, соответственно название сайта подлежит охране наряду с названиями всех прочих произведений.

**Все сайты** в совокупности составляют **Всемирную паутину**, где коммуникация (паутина) объединяет сегменты информации мирового сообщества в единое целое — базу данных и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

## Основные составляющие сайта

Первый раз попав на сайт, уже спустя пару секунд мы понимаем, готовы ли мы иметь дело с данной компанией или стоит искать дальше. А всё потому что некоторые сайты обладают успешными элементами, а другие – наоборот всячески от себя отталкивают своим внешним оформлением посетителей.

**Первая необходимая составляющая** – качественное юзабилити. Сайт должен быть простым и интуитивно понятным, а не виртуальной имитацией лабиринта.

**Вторая составляющая** – качественный контент. Уникальные статьи, яркие иллюстрации и хорошие фотографии являются неотъемлемой частью эффективного сайта. Ваша цель оформить сайт таким образом, чтобы посетителю было интересно находиться и проводить время на нем.

**Третья составляющая** - яркий, красивый и привлекательный дизайн. Всё-таки пользователь, перед тем, как начать пользоваться сайтом, должен его увидеть

**Четвёртая составляющая** - качественная работа без сбоев.

Данные составляющие сайта не могут существовать без кода.

**Код** - это текст, в котором разработчики описывают дизайн и поведение сайта на языках программирования. Код подразделяется на интерфейсный и серверный.

Интерфейсным кодом описывается дизайн. С его помощью компьютеры отображают элементы на сайте. Чтобы написать интерфейсный код, программисты используют языки HTML, CSS и Javascript. Вёрстка сайта — это процесс создания интерфейсного кода.

Серверным кодом описывается поведение сайта. С его помощью сервер определяет действие, которое произойдёт, если пользователь нажмёт на кнопку или введёт текст. У одностраничных сайтов такого кода может и не быть. Серверный код нужен, если на сайте есть возможность создать аккаунт, купить товар или услугу. Серверный код пишут на PHP, Java, Ruby, Python и др.

## История развития языка разметки HTML

В 1989 году Тим Бернерс-Ли предложил руководству международного центра высоких энергий (CERN) проект распределенной гипертекстовой системы, которую он назвал World Wide Web (WWW), Всемирная паутина. Первоначально идея системы состояла в том, чтобы при помощи гипертекстовой навигационной системы объединить все множество информационных ресурсов CERN в единую информационную систему.

Одним из компонентов технологии создания распределенной гипертекстовой системы World Wide Web стал язык гипертекстовой разметки HTML (HyperText Markup Language – язык гипертекстовой разметки документов), разработанный Тимом Бернерсом-Ли на основе стандарта языка разметки печатных документов — SGML (Standard Generalised Markup Language, стандартный обобщенный язык разметки). Дэниел В. Конноли написал для него Document Type Definition — формальное описание синтаксиса HTML в терминах SGML.

Разработчики HTML смогли решить две задачи:

1. предоставить дизайнерам гипертекстовых баз данных простое средство создания документов;
2. сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет выбора теговой модели описания документа. Такая модель широко применяется в системах подготовки документов для печати.

Язык НТМL позволяет размечать электронный документ, который отображается на экране с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные метки, иллюстрации, аудио- и видеофрагменты и так далее. В состав языка вошли развитые средства для создания различных уровней заголовков, шрифтовых выделений, различные списки, таблицы и многое другое.

Вторым важным моментом, повлиявшим на судьбу HTML, стало то, что в качестве основы был выбран обычный текстовый файл.

Таким образом, гипертекстовая база данных в концепции WWW — это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметка) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки). Гипертекстовые ссылки, устанавливающие связи между текстовыми документами, постепенно стали объединять самые различные информационные ресурсы, в том числе звук и видео; в результате возникло новое понятие — гипермедиа.

Такой подход предполагает наличие еще одного компонента технологии — интерпретатора языка. В World Wide Web функции интерпретатора разделены между Web-сервером гипертекстовой базы данных и интерфейсом пользователя. Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, обеспечивает предпроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

## Версии

**RFC 1866— HTML 2.0, одобренный как стандарт 22 сентября 1995 года;**

**HTML 3.2 — 14 января 1997 года;**

**HTML 4.0 — 18 декабря 1997 года;**

HTML 4.01 (изменения, причём более значительные, чем кажется на первый взгляд) — 24 декабря 1999 года;

ISO/IEC 15445:2000 (так называемый ISO HTML, основан на HTML 4.01 Strict) — 15 мая 2000 года.

HTML 5— в разработке. Конец разработки запланирован на 2014 год.

HTML 5.1 начал разрабатываться примерно 19 декабря 2012 года.

Официальной спецификации HTML 1.0 не существует. До 1995 года существовало множество неофициальных стандартов HTML. Чтобы стандартная версия отличалась от них, ей сразу присвоили второй номер.

Версия 3 была предложена Консорциумом всемирной паутины(W3C) в марте 1995 года и обеспечивала много новых возможностей, таких как создание таблиц, «обтекание» изображений текстом и отображение сложныхматематических формул, поддержка gif формата. Даже при том, что этот стандарт был совместим со второй версией, реализация его была сложна для браузеров того времени. Версия 3.1 официально никогда не предлагалась, и следующей версией стандарта HTML стала 3.2, в которой были опущены многие нововведения версии 3.0, но добавлены нестандартные элементы, поддерживаемые браузерамиNetscape NavigatorиMosaic.

В версии HTML 4.0 произошла некоторая «очистка» стандарта. Многие элементы были отмечены как устаревшие и нерекомендованные (англ.deprecated). В частности, элемент font, используемый для изменения свойств шрифта, был помечен как устаревший (вместо него рекомендуется использовать таблицы стилей CSS).

В 1998 году консорциум Всемирной паутины начал работу над новым языком разметки, основанном на HTML 4, но соответствующим синтаксису XML. Впоследствии новый язык получил название XHTML. Первая версия XHTML 1.0 одобрена в качестве Рекомендации консорциума Всемирной паутины26 января 2000 года.

Планируемая версия XHTML 2.0 должна была разорвать совместимость со старыми версиями HTML и XHTML, но 2 июля 2009 года консорциум Всемирной паутины объявил, что полномочия рабочей группы XHTML2 истекают в конце2009 года. Таким образом, была приостановлена вся дальнейшая разработка стандарта XHTML 2.0.

В настоящее время Консорциум всемирной паутины разрабатывает HTML версии 5. Черновой вариант спецификации языка появился в Интернете 20 ноября 2007 года.

## История развития языка стилей CSS

CSS (Cascading Style Sheets – каскадные таблицы стилей) – язык описания внешнего вида web-документа, созданного при помощи языка разметки (HTML).

CSS – это широкий спектр технологий, который одобрен консорциумом W3C (World Wide Web Consortium (Консорциума Всемирной Паутины)) и получил такое название, как «Web Стандарты». В 1990 г. возникла необходимость стандартизировать их в какие-то определенные единые правила, благодаря которым веб-дизайнеры и программисты проектировали бы сайты.

Концепция каскадных таблиц стилей была предложена Хоконом Виум Ли – норвежским ученым и специалистом в области информационных технологий, работавшим в то время на консорциум W3C.

17 декабря 1996 года была принята первая версия СSS как рекомендация W3C. 18 мая 1998 года принята вторая версия CSS, CSS2. 8 сентября 2009 года утверждена CSS2.1, как исправленная версия CSS2. В настоящий момент существует СSS3.

**CSS1**

В первой версии CSS имелась возможность задавать гарнитуру и размер шрифта, а еще изменять его стиль: обычный, курсив или полужирный. Также определять рамки, фоны, цвета текста и другие элементы страницы. Можно задавать расстояние между словами, межстрочные отступы и межсимвольный интервал. А также производить выравнивание текста, таблиц, изображений.

Имелись свойства внутренних и внешних отступов и рамок, ширины, высоты и блоков. А также входили в данную спецификацию ограниченные средства по позиционированию различных элементов, например, clear и float.

**CSS2**

CSS2 основана на CSS1, а также сохранила обратную совместимость с добавлением некоторых функций, а именно:

* Возникло фиксированное, абсолютное и относительное позиционирование. С помощью чего появилась возможность управлять размещением элементов без табличной верстки по странице
* Для разных носителей возможность устанавливать разными стилями
* Для звуковых носителей появилась возможность определять громкость и голос
* Возможно устанавливать на нечетных и четных страницах во время печати различные элементы
* Расширился механизм селекторов
* Возможность добавлять содержимое, которое не содержится в исходном документе

Консорциум W3C не поддерживает CSS2 и настоятельно рекомендует применять CSS2.1. CSS2.1 принята 2011 года 7 июня. Она основывается на CSS.2. Исправлены ошибки и удалены некоторые функции.

**CSS3**

В отличие от предыдущих версий спецификация разбита на модули, разработка и развитие которых идёт независимо. CSS3 основан на CSS2.1, дополняет существующие свойства и значения и добавляет новые.

**CSS4**

CSS4 разрабатывается консорциумом W3C с 29 сентября 2011 года. Модули CSS4 построены на основе CSS3 и дополняют их новыми свойствами и значениями. Все они существуют пока в виде черновиков (working draft).

## История развития языка программирования Python

Создание языка Python начиналось довольно медленно и неуверенно. Главным энтузиастом, который в 1990 году пытался воплотить Python в реальность, стал Гвидо Ван Россум. Именно этот человек, работая над разработкой языка ABC в Голландском институте CWI, понял, что хотел бы создать нечто новое. Это послужило стартом для написания нового интерпретатора; конечно, не без использования некоторых идей, взятых с ABC.

Интересным моментом выступает то, что первый рабочий прототип Python был создан на домашнем Макинтоше Гвидо, да и еще за пару выходных. Что касается распространения, то делалось это с помощью Интернет.

В 1996 году, когда данный проект набирал критическую массу, к разработке подключился Стив Маевский, который был довольно известным в сети, так как вел свой блог «Сравнительная критика языков программирования». Стив, как и Гвидо был поклонником Macintosh, возможно, это и послужило основой их сотрудничества. Стоить отметить, что язык получил название «Python» не в честь вида змей, как ошибочно считают многие разработчики. Во времена разработки «Питона» Гвидо любил смотреть комедийное шоу «Воздушный цирк Монти Пайтона», поэтому и назвал своей проект в честь Монти Пайтона.

Так как Питон имел отличный потенциал и свободно распространялся через Интернет, в него появилось ядро последователей — люди, которые были заинтересованы в развитии Python как языка программирования. В начале своего пути, этот язык имел вид небольшого интерпретатора с малым количеством функций и полным отсутствием ООП, что всех не устраивало и мотивировало на дальнейшее развитие языка.

Уже в 1991 году стали появляться первые средства ООП разработки.

Спустя некоторое временя, Гвидо предложили должность в корпорации CNRI, которая находится в Америке. Недолго думая, Гвидо покинул Голландию и принялся за работу. Занимаясь проектами компании, он часто использовал Python для решения многих задач, а в свободное время занимался его развитием как интерпретатора.

По такой схеме Питон развивался до 1999 года и получил версию 1.5.2. По достижении этой планки в жизни Гвидо начались изменения. Компания все больше загружала его работой, что сильно уменьшало время для любимого занятия.

Это побудило задуматься Гвидо о целесообразности такой работы, в результате чего, он принял решение искать спонсора, который даст возможность работать только над развитием языка. Так как к тому времени в интернете уже существовало немалое сообщество пользователей, фирма BeOpen решила принять участие в продвижении Python. По контракту с CNRI Гвидо обязался выпустить версию 1.6, что он и сделал перед уходом. Работая с BeOpen он показал миру версию 2.0. Многие утверждают, что версия 2.0 дала сильный толчок в социальном плане. А все потому, что процесс развития языка стал более открытым. Гвидо перевел все данные на SourceForge, что сильно понравилось сообществу, которое требовало внедрения возможности участия в разработке кода. Кроме этого в то время появился Юникод, а это большой шаг вперед. К Юникоду создали новый механизм регулярных выражений, который мог работать как с обычными строками, так и с Юникодовыми.

Через некоторое время в компании BeOpen начались проблемы. Они решили, что Гвидо должен работать усердней и приносить деньги, а не только просить их, на развитие проекта. Такое поведение не пришлось по душе Гвидо – он уволился и начал размышлять куда идти дальше. Во многих интервью, Гвидо рассказывает, что этот фрагмент жизни был переломный.

В этот раз, свое финансирование ему предложила компания Digital Creations – авторы Zope. Как не странно, но это было пятое предложение от них, на что Гвидо согласился. В этой компании вся команда разработчиков «питона» получила большие возможности, что дало плоды. В том же году был выпущена версия 2.1. Теперь в питоне появились новые объекты с языков closures и иерархия: функции можно вкладывать друг в друга, сохраняя при этом доступ к переменным окружающих функций. Это сильно изменит язык, а главное сильно улучшит его подходы к способу программированию. На данный момент существует версия 3.5.1, что демонстрирует его развитие, ведь каждый год разработчики проделывают огромную работу. Все это превратило простой интерпретатор в очень популярный язык программирования, который используется как первый в обучении миллионов студентов по всему миру.

Язык Python настолько обширный, гибкий и простой в изучении, что на нём можно даже писать сайты, но для этого нужна специальная библиотека - Django.

## История развития библиотеки для создания сайтов Django

Классический процесс разработки приложения выглядит так:

1. Создать приложение с нуля.
2. Создать другое приложение с нуля.
3. Реализовать работу первого приложения со вторым.
4. Переработать код так, чтобы оба приложения использовали общие функции.
5. Повторить шаги 2-4 несколько раз.
6. Вы изобрели среду разработки.

Django органично росла во время разработки реальных приложений, созданных командой разработки в Лоуренсе, штат Канзас, США. Она родилась в конце 2003 когда программисты газеты «Lawrence Journal-World», Эдриан Холовати (Adrian Holovaty) и Симон Виллисон (Simon Willison), начали использовать язык Python для разработки своих приложений. Команда World Online, ответственная за разработку и поддержку нескольких локальных новостных сайтов, процветала в среде разработчиков, диктуемой срочностью работы журналиста. Для сайтов, включая LJWorld.com, Lawrence.com и KUsports.com, журналисты и менеджеры требовали, чтобы новые возможности и целые приложения были разработаны максимально быстро, часто счёт шёл на дни или на часы. Таким образом, Симон и Эдриан создали среду разработки экономящую время, исходя из своих потребностей — это был единственный способ, с помощью которого они могли создавать управляемые приложения в жёстких рамках сроков.

Летом 2005, развив эту среду до такого состояния, что она поддерживала большинство сайтов World Online, команда разработчиков, включая Якоба Каплан-Мосса (Jacob Kaplan-Moss), решила выпустить среду в виде программного обеспечения с открытым исходным кодом. Они выпустили его в июле 2005 под именем Django в честь джазового гитариста Джанго Рэйнхарда (Django Reinhardt).

Теперь, по прошествии нескольких лет, Django является грамотным проектом с открытым исходным кодом с десятками тысяч пользователей, над которым работают разработчики по всей планете. Два автора из команды World Online («Великодушные пожизненные диктаторы», Эдриан и Якоб) всё ещё осуществляют общее руководство над развитием среды разработки, но теперь оно в большей степени зависит совместных усилий команды разработчиков. World Online обеспечивает другие важные аспекты, такие как время разработчиков, маркетинговые материалы и хостинг/канал для сайта среды.

Эта история уместна, потому что помогает объяснить два ключевых аспекта. Первый аспект — основное назначение Django. Так как Django родился в новостной среде, он предоставляет некоторые средства (такие как административный интерфейс, описанные в главе «Интерфейс администратора Django»), которые хорошо подойдут для контент-ориентированных сайтов, таких как Amazon.com, craigslist.org и washingtonpost.com, которые предоставляют динамическую информацию из базы данных. Существует разница между быть особенно эффективным в чём-то и быть неэффективным в другом.

Вторым аспектом является то, как происхождение Django сформировало культуру его сообщества открытого исходного кода. Так как Django был получен из реального кода, а не был академической разработкой или коммерческим продуктом, он полностью сфокусирован на решение проблем разработки, с которыми сталкивались и продолжают сталкиваться его авторы. В результате, Django постоянно совершенствуется. Основатели среды имеют свой интерес в том, чтобы Django экономил их время, создавал приложения лёгкие в обслуживании и хорошо работал под нагрузкой. При отсутствии других причин, разработчики мотивируются своими собственными эгоистичными желаниями сэкономить своё время и наслаждаться своей работой.

# Практическая часть

Сайт электронного журнала создан на основе языка программирования Python в связке с фреймворком-библиотекой для создания сайтов Django. Пользователь сайта видит веб-страницы написанные на HTML в связке в JavaScript и CSS файлами от Bootstrap. Для редактирования файлов сайта был использован *PyCharm (Community edition)*, текстовый редактор *Atom*; для облачного хранения файлов сайта был использован *GitHub*; для хостинга (размещения сайта в интернете) был использован хостинг *pythonanywhere.com*.

На сайте есть три типа личных кабинета:

1. Директор
2. Учитель
3. Ученик

У каждого из кабинетов есть свои страницы:

Для директора:

1. Главная(новостная) страница
2. Настройки школы
3. Профиль

Для учителя:

1. Главная(новостная) страница
2. Мой класс
3. Мои уроки
4. Профиль

Для учителя:

1. Главная(новостная) страница
2. Дневник
3. Статистика
4. Расписание
5. Профиль

На главной странице можно найти перечень новостей, которые опубликовал директор. Кликнув на новость, можно полностью с ней ознакомиться, при этом, если вы зашли как директор, вы можете добавлять, изменять и удалять новости.

В “настройках школы” с аккаунта директора можно полностью настроить образовательное учреждение: задать название, зарегистрировать учителей, учеников, создать классы, добавить расписание и т.п.

На вкладке “профиль” можно посмотреть свой профиль, то есть Ф.И.О. и статус аккаунта.

На вкладке “Мой класс” с аккаунта учителя можно посмотреть список учеников класса, руководителем которого является данный учитель.

На вкладке “Мои уроки” с аккаунта учителя проводится процесс обучения: выбрав предмет и класс учитель может проставлять отметки ученикам за последние две недели, а также задавать домашние задания.

На вкладке “Дневник” с аккаунта ученика можно посмотреть последние полученные отметки и актуальные домашние задания.

На вкладке “Статистика” с аккаунта ученика можно контролировать успеваемость, путём просмотра всех отметок а также среднего балла по каждому предмету.

На вкладке “Расписание” можно посмотреть расписание для своего класса, причём, для удобства, нажав по специальной кнопке, можно перейти на текущий день, где текущий урок будет подсвечен

На данном этапе сайт полностью готов. Перейти на него можно по ссылке: [**http://networkdiary.pythonanywhere.com**](http://networkdiary.pythonanywhere.com)

# Заключение

Проделанная работа позволила создать полезный продукт и, методом проб и ошибок понять, как создаются веб-сайты.

Результатом данной работы является удобный, простой в понимании электронный журнал. Файловая система проекта насчитывает 14 python-скриптов, 36 html-документов, а также базу данных, в которую закладывается информация так, что она никогда оттуда не пропадёт.

В результате были выполнены следующие задачи:

1. Был изучен один из продвинутых способов создания сайта;
2. Был изучен способ работы с базами данных;
3. Было изучено 2 языка: HTML и Python;
4. Был разработан дизайн в стилистике сайта ГКЛ;
5. Были протестированы все функции, в том числе работоспособность на компьютерах и телефонах.

Дальнейшее развитие проекта заключается в технической поддержке, а также обновлении дизайна и добавлении новых функций.

# Список литературы

1. Гончаров А. Самоучитель HTML. – СПб.: Питер, 2002. – 240 с.

2. Дейнеко И.В. Презентация работы библиотеки: Методика разработки и создания библиотечной компьютерной презентации или стендового доклада. – М.: Чистые пруды, 2005. – 32с. – (Библиотечка «Первого сентября», серия «Библиотека в школе». Вып.4.).

3. Дуванов А.А. Кухня web-мастера Сидорова: Основы практического web-дизайна. – М.: Чистые пруды, 2005. – 32с. – (Библиотечка «Первого сентября», серия «Информатика»).

4. Жадько Н.В. Проектное развитие библиотек. – М.: Елена, 2000. – 96с.

5. Информатика: Энциклопедия для детей. Т.22/ Глав. ред. Е. А. Хлебалина, вед. науч. ред. А.Г. Леонов. – М.: Аванта+, 2003. – с.511 - 516.

6. Карпов И.П. Интернет-филолог// Литература. – 2005. - №15. – с.36-41.

7. Леонтьев В.П. Новейшая энциклопедия Интернет. – М.: ОЛМА-ПРЕСС, 2002. – 607с.: ил.

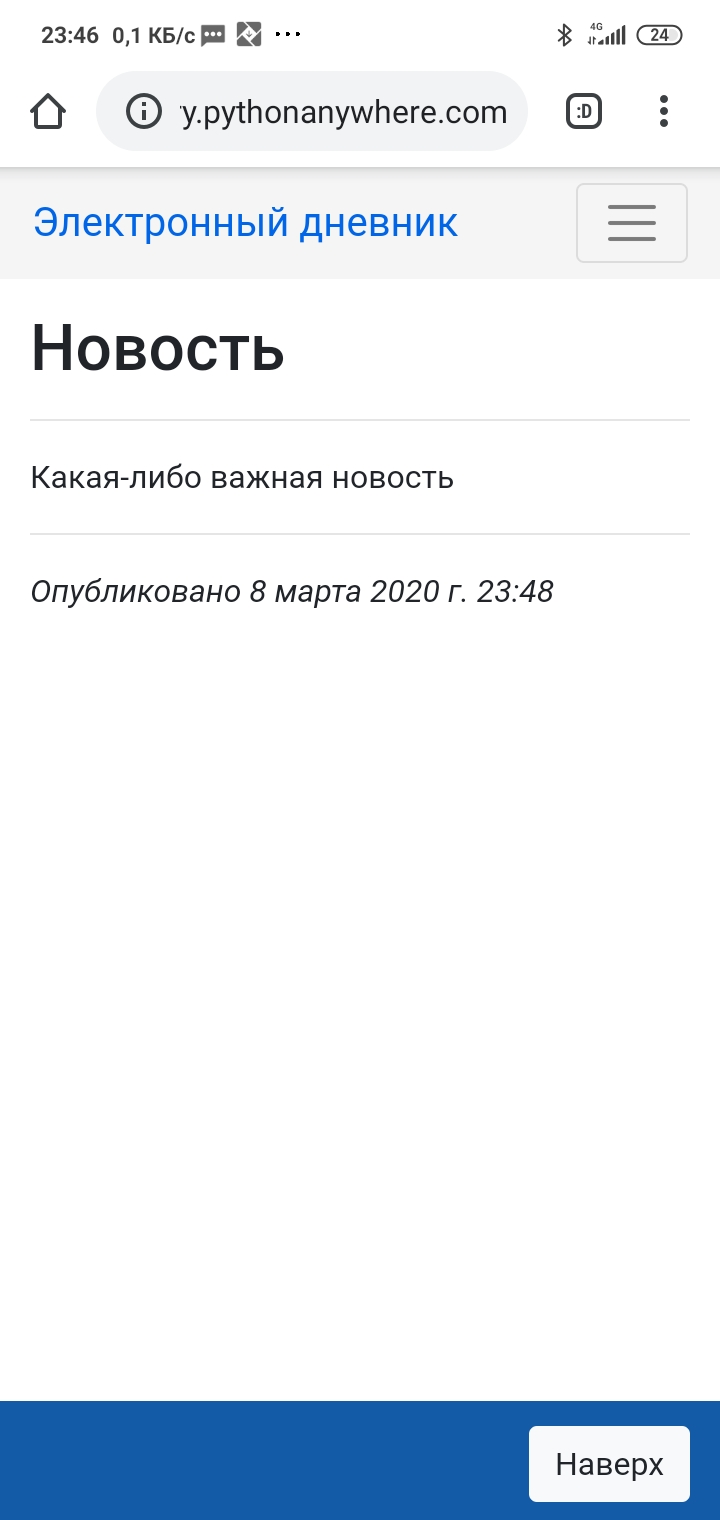
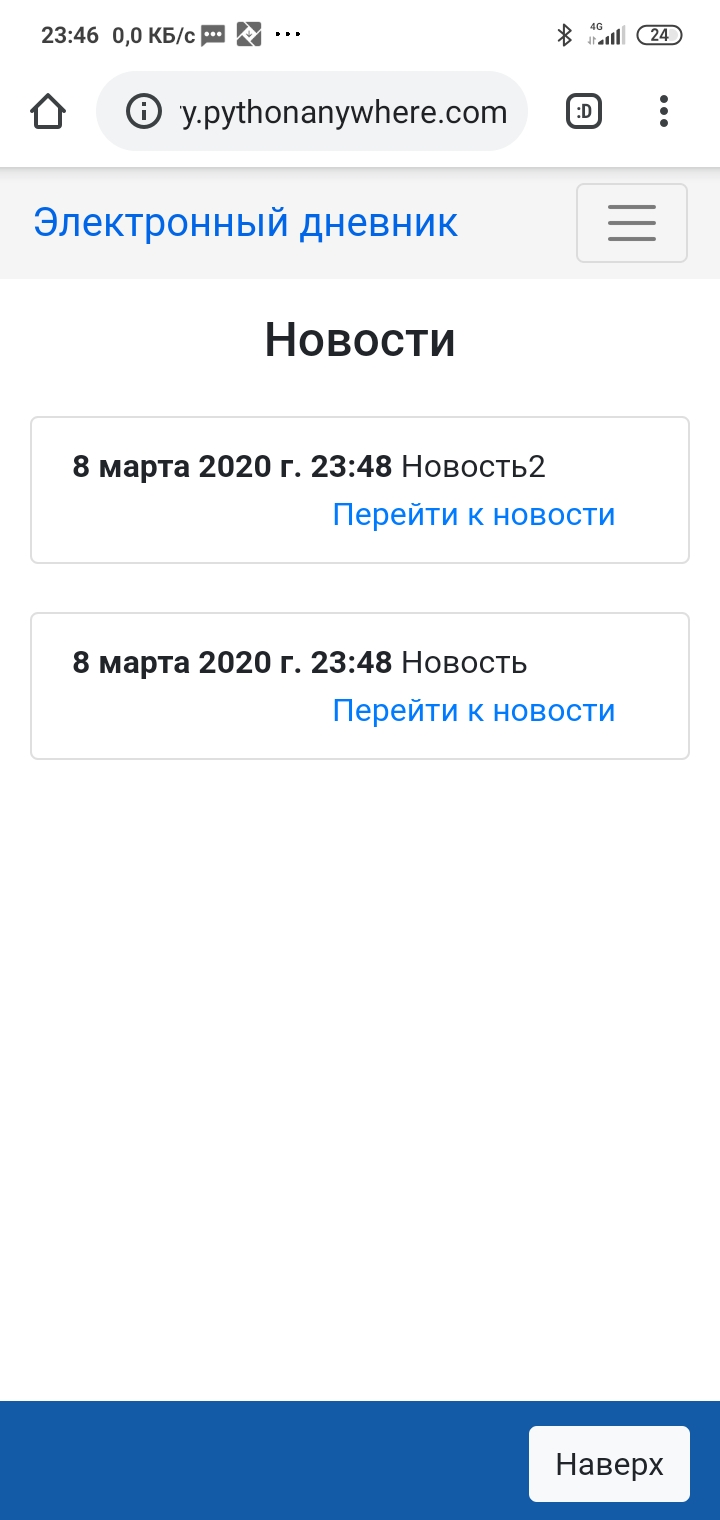
8. Леонтьев В.П. Новейшая энциклопедия персонального компьютера 2001. – М.: ОЛМА-ПРЕСС, 2001. – 847с.: ил.

9. Симонович С.В., Мураховский В.И. Интернет у вас дома: Полное руководство для начинающего пользователя. – М.: АСТ-ПРЕСС: Инфорком-Пресс, 2001. – 432с.

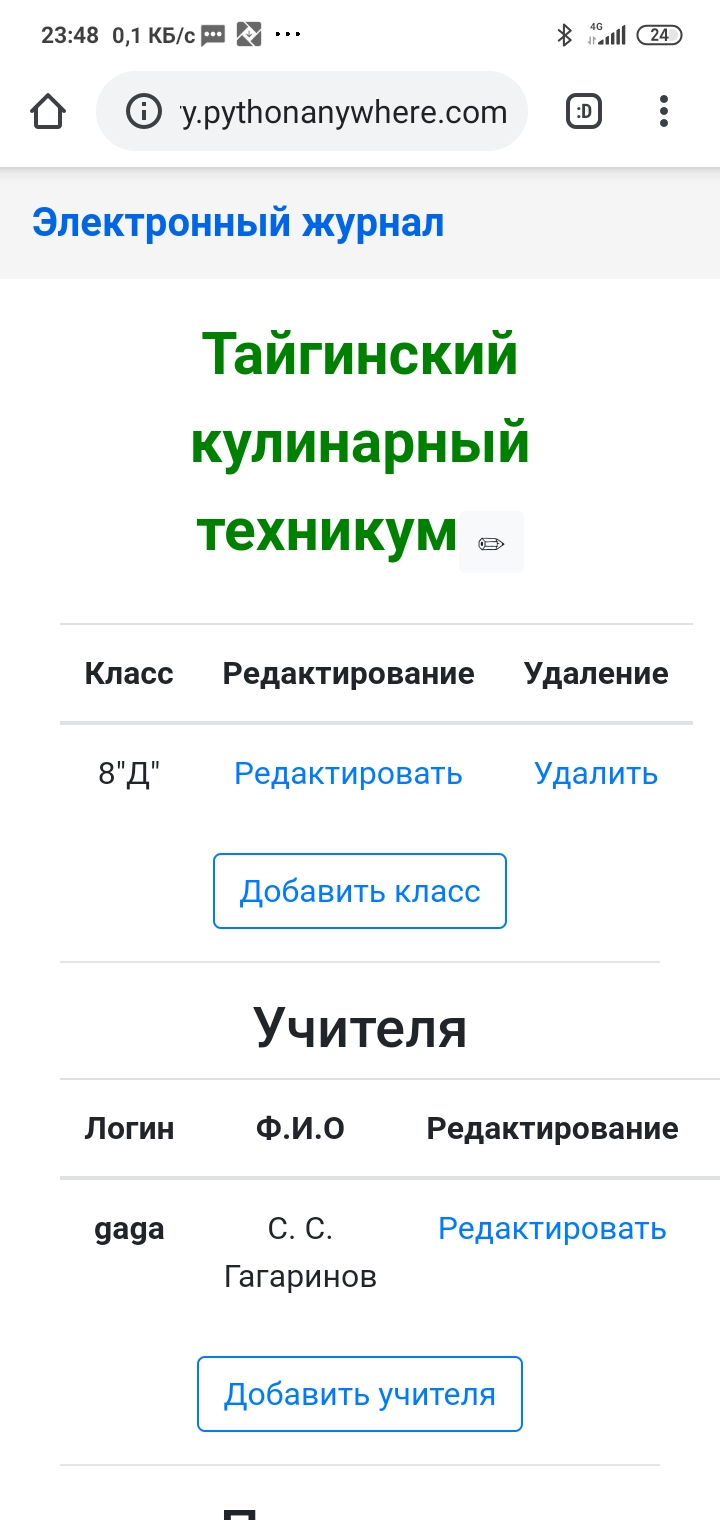
10. Угринович Н.Д. Практикум по информатике и информационным технологиям. – М.: Лаборатория Базовых знаний, 2002. – 394с.: ил.

# Приложение

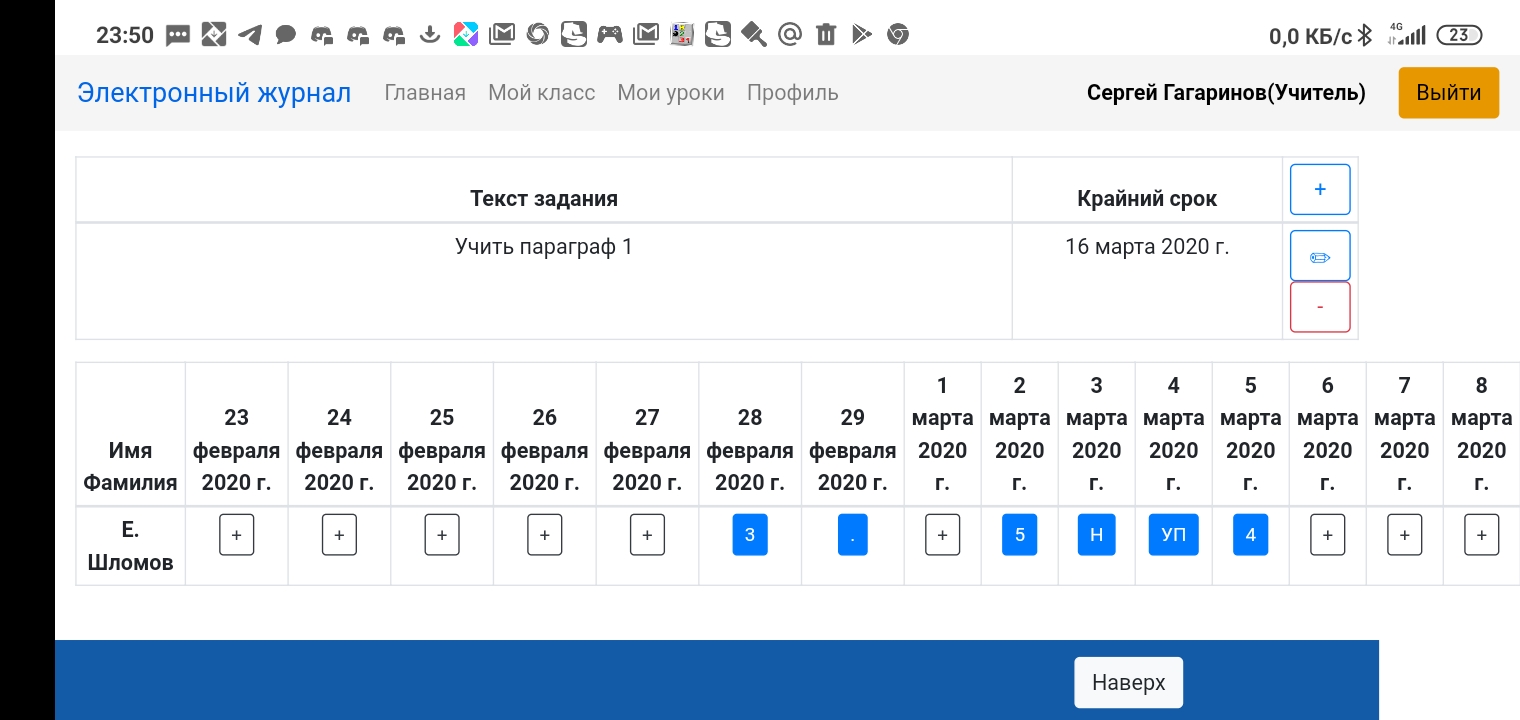
1. Главная страница



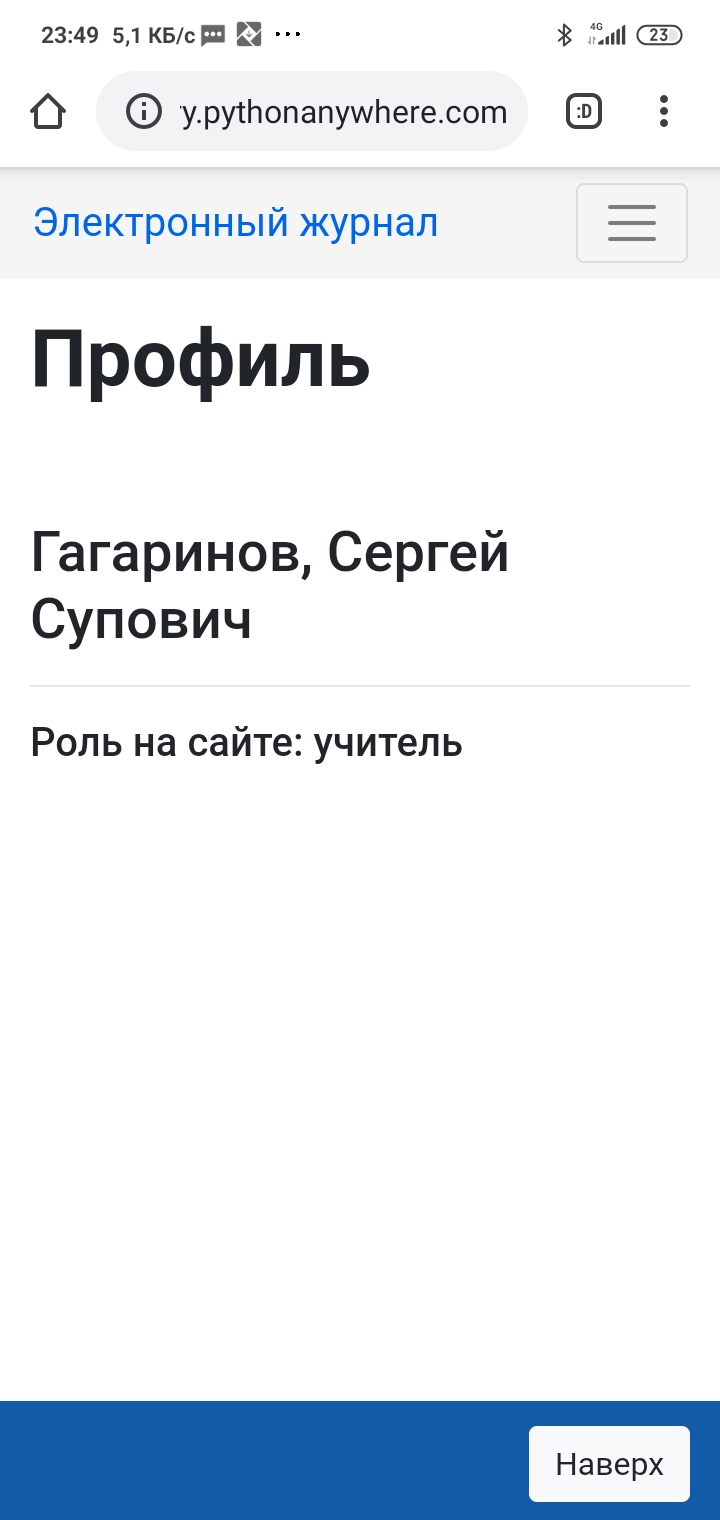
1. Управление школой



1. Выставление отметок и опубликовывание домашнего задания



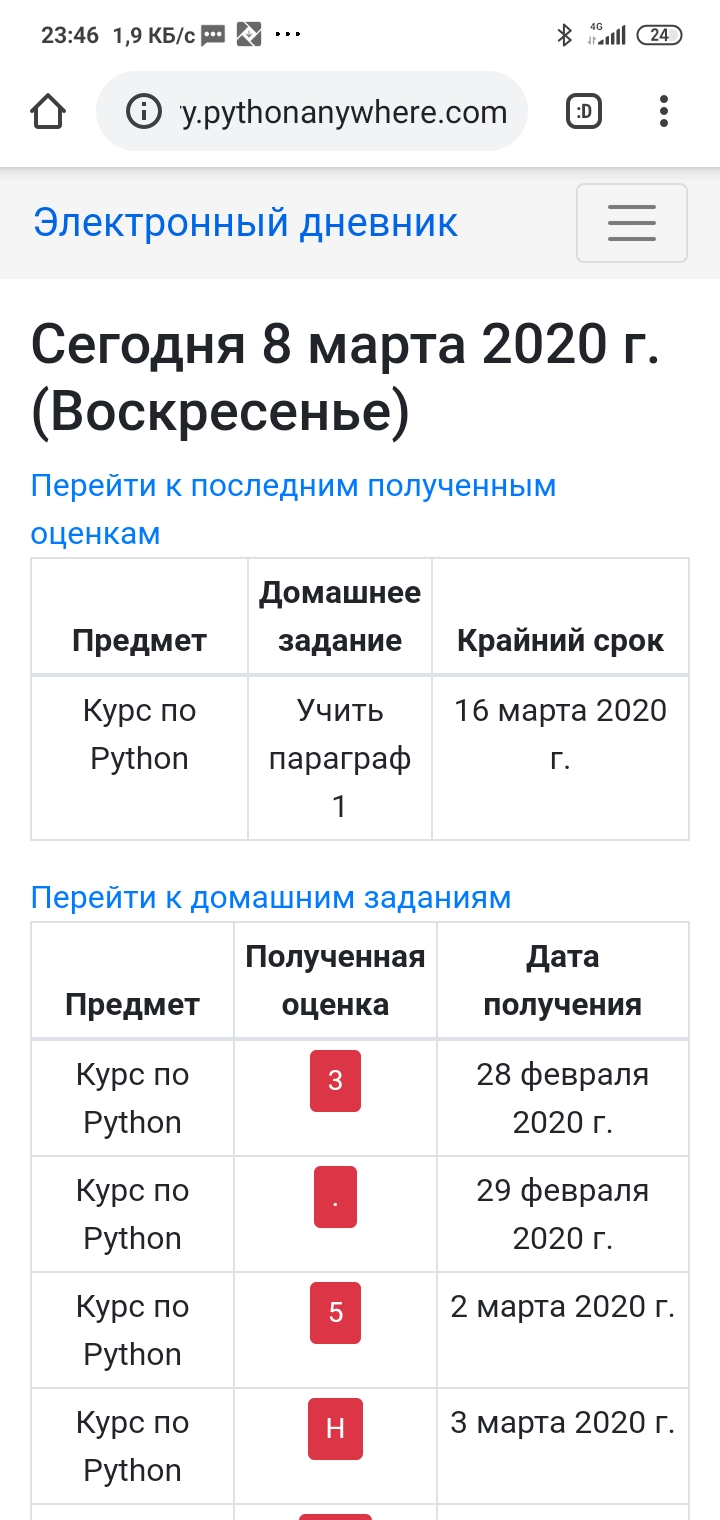
1. Профиль



1. Расписание



1. Отметки



1. Статистика

