

РЕФЕРАТ

Выпускная квалификационная работа бакалавра состоит из 33 страниц, 23 рисунков, 29 используемых источников и 1 приложения.

ВЕБ-СЕРВИС, ЛИЧНЫЙ КАБИНЕТ ПРЕПОДАВАТЕЛЯ, DJANGO

В выпускной квалификационной работе рассматривается веб-сервис, предназначенный для преподавателей. Рассмотрены современные веб-сервисы МАИ, проведён анализ их недостатков. Предложен альтернативный сервис, отвечающий всем необходимым требованиям.

Во введении обосновывается актуальность и цель ВКР, теоретическая и практическая значимость.

В основной части даётся описание объектов исследования, рассматриваются проблемы, которые решает данная ВКР. В разделе 1.1 проводится исторический анализ происхождения веб-сервисов. В разделе 1.2 веб-сайты классифицируются по предоставляемой информации. В разделе 1.3 приводится классификация веб-сайтов по доступности сервисов. В разделе 1.4 веб-сайты классифицируются по цели создания. В разделе 1.5 приводится классификация веб-сайтов по технологии создания. В разделе 1.6 проводится анализ задач личного кабинета в целом. В разделе 1.7 проводится анализ текущих недостатков веб-сервисов МАИ. В разделе 1.8 проводится анализ поставленных задач. В разделе 2.1 описываются методы и средства решения поставленных задач. В разделе 2.2 даётся описание файловой структуры проекта. В разделе 2.3 демонстрируется работа программы, изложена практическая составляющая работы.

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ.....	4
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	6
1.1 Исторический анализ	6
1.2 Классификация веб-сайтов по типу информации	7
1.3 Классификация веб-сайтов по доступности сервисов	9
1.4 Классификация веб-сайтов по цели создания	10
1.5 Классификация веб-сайтов по технологии создания.....	10
1.6 Личный кабинет	11
1.7 Анализ объекта исследования	15
1.8 Анализ поставленных задач.....	16
2 ПРАКТИЧЕСКАЯ ЧАСТЬ	18
2.1 Методы и средства решения задач	18
2.2 Файловая структура проекта.....	21
2.3 Решение задачи создания веб-сервиса	22
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30
ПРИЛОЖЕНИЕ А Разработанный код	33

ВВЕДЕНИЕ

Московскому авиационному институту уже более 90 лет. Сегодня это передовой аэрокосмический вуз страны, ежегодно выпускающий сотни и тысячи высококлассных специалистов.

На базе МАИ постоянно проводятся различные научные исследования, воплощаются в жизнь новые идеи, разрабатываются и внедряются современные технологии. Институт, когда-то бывший лишь Высшим аэромеханическим училищем, стал национальным исследовательским университетом.

Многие исследования и разработки имеют свой приоритет, который может меняться в зависимости от множества различных факторов:

- внешнеполитическая ситуация;
- экологическая катастрофа;
- эпидемиологической обстановка;
- прочие чрезвычайные ситуации различной степени важности.

В нашем случае, косвенной причиной к написанию данной ВКРБ – выпускной квалификационной работы бакалавра – послужила пандемия коронавируса Covid-19.

В связи с переводом всех институтов университета на дистанционное обучение, а также из-за прекращения работы многих иностранных сервисов (таких, как, например, Microsoft Teams), администрация МАИ столкнулась с необходимостью в кратчайшие сроки модернизировать и улучшить ряд сервисов, предназначенных для дистанционного проведения занятий, сдачи домашних заданий, лабораторных и курсовых работ.

Также большинству преподавателей было необходимо немедленно обучиться и начать работать с существующими технологиями. Ситуация осложнялась следующими факторами:

- необходимо в сжатые сроки обработать довольно большой объём информации: пройти обучающие курсы, вебинары, разобраться самостоятельно;
- у веб-сервисов МАИ не самый дружелюбный к пользователю интерфейс, в нём будет сложно разобраться без специальных инструкций.

Целью данной выпускной квалификационной работы бакалавра будет анализ существующих недостатков веб-сервисов МАИ, а также создание про-

тотипа альтернативного веб-сервиса, который был бы прост в изучении и понятен большому числу пользователей, а также имел бы все функции, необходимые преподавателю:

- просмотр расписания, добавление, изменение и удаление занятий;
- просмотр журналов групп, удобное добавление, изменение и удаление оценок по пятибалльной или любой другой шкале;
- удобная постраничная навигация и минималистичный интерфейс;
- удобная связь со сторонними вспомогательными веб-сервисами.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Исторический анализ

Интернет-сайт (также веб-сайт) – это набор взаимосвязанных электронных страниц, которые сгруппированы по разделам и содержат текстовую, графическую и мультимедийную информацию (анимацию, видеоролики и пр.). Веб-сайт располагается в Интернете по определенному адресу – так называемому доменному имени сайта в виде `www.имя_сайта.com`. Каждая страница сайта – это текстовый документ, написанный на языке программирования или просто размеченный с помощью определённых инструментов.

Ещё в 1940-х годах, с созданием немецким учёным Конрадом Цузе первого компьютера, многими учёными продвигалась идея об огромной базе знаний всего человечества с возможностью обмена информацией из любой точки планеты.

Спустя 20 лет, уже в 1962 году американский учёный Джозеф Ликлайдер описал концепцию будущей сети в своей работе "Galactic Network". И уже через пять лет, в 1969 году учёные смогли связать между собой два компьютера ARPA, передав сообщение со второй попытки. Получившуюся сеть назвали ARPANET. Этот момент считается стартовой точкой, откуда берёт своё начало история Интернета.

Позднее, в 1984 году появилась сеть NSFNet, созданная Научным Фондом США. А через пять лет, в 1989 году учёный из швейцарского CERN Тим Бернерс-Ли предложил прототип того Интернета, который в видоизменённой форме дошёл до наших дней - глобальный гипертекстовый проект, называющийся теперь "Всемирной паутиной". Его суть подразумевала публикацию гипертекстовых документов, связанных между собой гиперссылками. Также учёным и его помощниками были изобретены идентификатор URI(а затем URL), протокол HTTP и язык разметки веб-страниц HTML - вещи, без которых невозможно себе представить современный интернет.

Стоит отдельно упомянуть и про попытку создания Интернета в СССР. В 1959 году (за 10 лет до создания ARPANET) советскими учёными Анатолием Китовым и Виктором Глушковым был разработан проект ОГАС - общегосударственной автоматизированной системы, предназначенной для управ-

ления экономикой СССР и её планирования. К сожалению, по разным причинам - сначала из-за внутривластных разногласий, а затем из-за нехватки средств - проект не был утверждён сразу, и впоследствии неоднократно рассматривался, но так никогда и не был реализован.

Первый же сайт Всемирной паутины был запущен Тимом Бернерсом-Ли 6 августа 1991 года. На нём было размещено описание технологии WWW (World Wide Web) и других технологий, изобретённых учёным и описанных выше. Также на сайте была подробная информация о работе серверов и браузеров с инструкцией по их установке.

Стоит отметить, что на сайте позже был размещён ряд ссылок на другие сайты, т.е. сайт Тима Бернерса-Ли был не только первым сайтом, но ещё и первым интернет-каталогом.

По мнению изобретателя, главным достоинством веб-сайтов должен был стать быстрый поиск нужной информации.

Именно этот принцип лежит в основе почти всех современных сайтов.

Если в остальном мире активное распространение Интернета началось сразу после 1991 года, то в России Интернет начал развиваться только во второй половине 1990-х годов - так, например, только в 1995 году был создан первый сайт на русском языке. Сегодня же российский сегмент Интернета (Рунет) активно развивается во всех сферах.

1.2 Классификация веб-сайтов по типу информации

Веб-сайты можно разделить на несколько категорий по типам:

Сайт-визитка. Это самый простой тип сайта. Как правило, подобный сайт содержит информацию о своём владельце (человеке, ИП, компании) и контакты. Такой сайт прост в создании и использовании, а значит, его разработка стоит недорого.

Корпоративный сайт. Такой тип сайта уже намного сложнее, и для его создания необходимо уже не просто несколько HTML-страниц, а полноценная система управления сайтом. Как правило, обычно такие сайты содержат всю информацию о товарах, контактах компании и её деятельности и ставят перед собой цель сформировать имидж компании и предоставить наиболее полную информацию о ней посетителям.

Интернет-магазин. Как правило, основная функция подобного типа сайтов – это продажа различных товаров. На таком сайте будут размещены характеристики товаров, отзывы на них и т.д., а также будет присутствовать возможность заказа этих товаров. Как подкатегорию, можно выделить промо-сайты – они характеризуются упором на раскрутку какого-то одного бренда или товара.

Тематические сайты. Подобный тип сайтов обычно специализируется на одной конкретной тематике, собирая информацию по ней. Это может быть как интернет-журнал о биологии, так и энциклопедия, и даже сборник законов по УК РФ и их правоприменительных практик.

Сайты для общения. В этом типе сайтов присутствует большее разнообразие, чем в других категориях. Отчасти это объясняется различными формами коммуникации пользователей между собой. Например, интернет-порталы. Как правило, подобная категория сайтов по организации и структуре напоминает тематические сайты, отличаясь более развитым функционалом и большим количеством разделов, например, форумом. Интернет-форумы же, в свою очередь, могут существовать и как самостоятельный вид сайта. Как правило, обычно на подобных сайтах есть определённая тематика, на которую пользователи создают темы и комментируют их.

Социальные сети – тип сайтов, созданных в основном для личного общения пользователей между собой. Как правило, на таких сайтах пользователи заводят личные страницы и даже объединяются в группы по интересам.

Поисковые системы. Отдельно стоит упомянуть и такой тип сайтов. Обычно на подобных сайтах реализуются алгоритмы поиска, и, как правило, такие сайты выполняют вспомогательную функцию поиска определённой информации в интернете по запросам пользователей.

Стоит отметить, что у вышеописанных типов сайтов существует ещё множество подкатегорий, которые не были упомянуты в силу очевидности их принадлежности к той или иной категории, а также потому, что целью данной выпускной квалификационной работы бакалавра не ставится анализ существующих категорий сайтов.

Конечно, в Интернете редко можно встретить сайт, который со стопроцентной уверенностью можно было бы отнести к одной из вышеописанных ка-

тегорий. Как правило, большинство сайтов являются смешанными, выполняя не одну, а две или более функций, являясь агрегаторами различных направлений деятельности компании и пользователей. Например, на корпоративном сайте также может быть и интернет-магазин, и форум для общения пользователей, и некое подобие социальной сети, и многое другое – такому сайту даже иногда может понадобиться собственная, внутренняя поисковая система. Таким агрегатором, например, является экосистема веб-сервисов Московского Авиационного института с доменом mai.ru, показанным на рисунке 1.1.

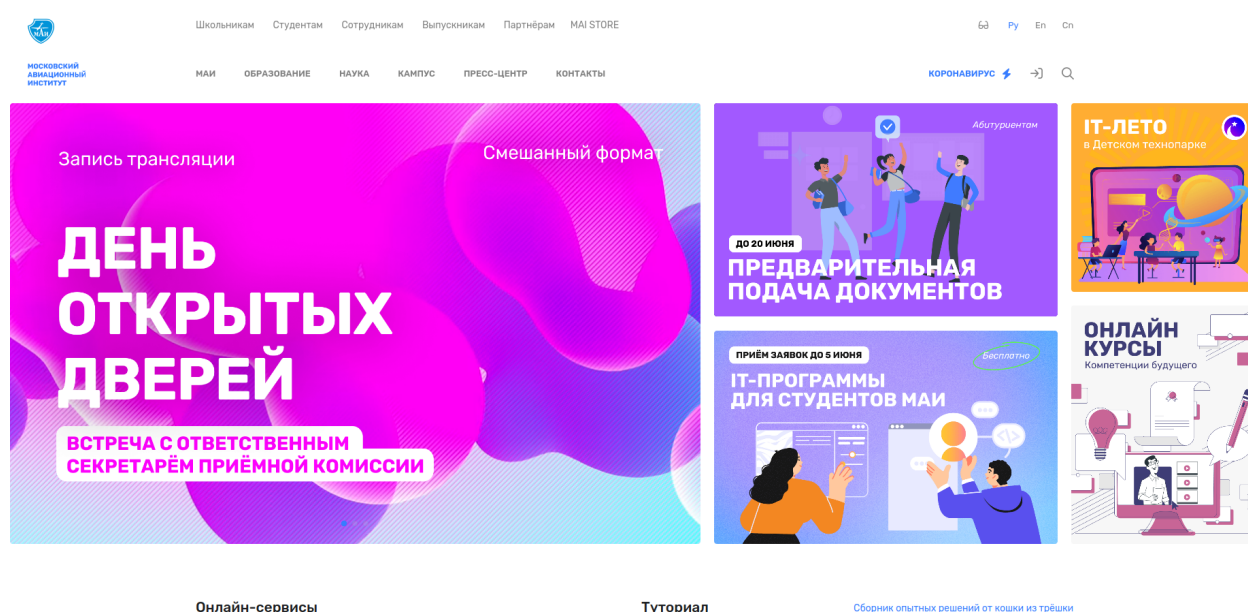


Рисунок 1.1 — Главная страница mai.ru

1.3 Классификация веб-сайтов по доступности сервисов

Веб-сайты могут делиться на две категории:

Открытые. Такой тип сайтов, как правило, доступен для каждого пользователя в Интернете, включая всю содержащуюся на нём информацию, и не требует прохождения процедуры регистрации.

Закрытые. На данном типе сайтов для доступа к информации обычно требуется пройти процедуру регистрации либо получить некоторое специальное приглашение, дающее доступ. Для незарегистрированных пользователей информация закрыта.

Конечно же, бывают и полужакрытые сайты, такие, как vk.com, главную страницу которого можно увидеть на рисунке 1.2. Для определённой части

контента требуется регистрация, но некоторые страницы сайта можно увидеть и без авторизации на сайте.

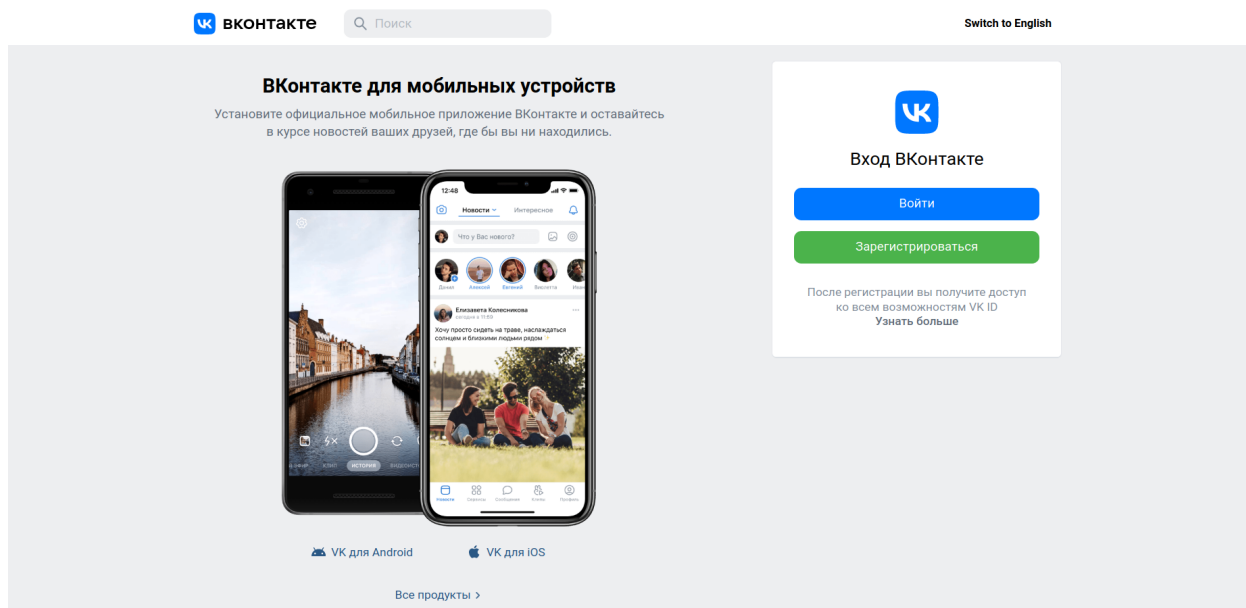


Рисунок 1.2 — Главная страница vk.com

1.4 Классификация веб-сайтов по цели создания

Здесь тоже обычно выделяют две категории веб-сайтов:

Коммерческие. Как правило, это довольно распространённый и доминирующий тип сайтов. Их целью является привлечение клиентов и потенциальных партнёров и информирование посетителей о тех товарах и услугах, которые может предоставить компания-владелец сайта. Как правило, это делается с целью увеличения прибыли. Обычно такие сайты имеют довольно сложную структуру и организацию, а также включают в себя множество сервисов и модулей, направленных как на клиентов, так и на сотрудников компании-владельца.

Некоммерческие. А вот перед подобными сайтами коммерческих задач, направленных на увеличение прибыли, не стоит. Обычно подобные сайты принадлежат государственным, муниципальным и социальным учреждениям или отдельным людям и, как правило, выполняют, в первую очередь, информационную роль, а также служат для обратной связи.

1.5 Классификация веб-сайтов по технологии создания

В этом разделе также рассматриваются два вида сайтов:

Статические. Как правило, у подобных сайтов внешний вид или их содержимое изменяется только тогда, когда изменения вносит администратор вручную, а не при каждом запросе к серверу. Большинство сайтов в Интернете основана именно на такой технологии, поскольку такое решение является экономичным, простым, безопасным, менее уязвимым и более надёжным. Также статические страницы лучше индексируются поисковыми системами.

Динамические. А вот у данного типа сайтов внешний вид и содержимое уже может зависеть от действий пользователей, изменения данных и других причин. Причём, могут применяться как сценарии, исполняемые на стороне клиента, так и на стороне сервера. Как правило, динамические страницы довольно гибки, но с их применением можно потерять производительность, так как каждый запрос, посылаемый к серверу, требует обработки уникального содержания. Поэтому при создании подобных страниц следует оптимизировать работу сайта и сервера в целом, иначе довольно велик риск сильного ухудшения видимости страницы поисковым системам в сравнении со статическими страницами.

1.6 Личный кабинет

Как правило, в роли физического личного кабинета обычно выступает отдельное помещение, в котором человек может работать, не отвлекаясь на другие, внерабочие задачи. Подобно этому, в виртуальном пространстве у пользователя может существовать виртуальный личный кабинет - место на определённом сайте, где хранятся данные пользователя. Именно в таком значении стоит понимать определение личного кабинета в данной выпускной квалификационной работе бакалавра.

Стоит отметить, что для доступа к личному кабинету обычно используется связка логин-пароль, нередко дополнительно усиленная двухфакторной аутентификацией с подтверждением номера телефона или электронной почты. Обычно в личный кабинет можно войти с любого устройства. Хорошим примером личного кабинета может служить учётная запись на Госуслугах или ФНС.

Личный кабинет пользователя на сайте, как правило, решает несколько очень важных задач.

Сбор информации о пользователе. Поскольку при регистрации, как правило, пользователь указывает о себе такие данные, как электронную почту, телефон, ФИО (а в некоторых случаях даже возраст, местонахождение, должность, платёжные данные и т.д.), это позволяет чётко идентифицировать каждого пользователя и, при необходимости, воспользоваться этими данными. Например, социальные сети очень часто уточняют предпочтения пользователя, чтобы показывать ему подходящий контент. Также этой информацией можно будет воспользоваться в том случае, если пользователь забыл часть данных для авторизации и не может войти в личный кабинет. Такая процедура обычно называется восстановлением логина и/или пароля.

Повышение лояльности пользователя. Как было отмечено выше, сервисы могут использовать личные данные пользователя. Как правило, это делается в основном для удобства и комфорта самого пользователя: в социальных сетях показывается подходящий контент, в интернет-магазинах появляются определённые персональные предложения, скидки и бонусы, на Госуслугах можно посмотреть историю походов к определённому врачу, а на сайте ФНС посмотреть, все ли налоги уплачены. Привязка к сервису номера телефона, аккаунта в социальной сети или других уникальных идентификаторов помогает сервису максимально подстроиться под интересы пользователя и уменьшает вероятность выбора им конкурентного сервиса.

Личное рабочее пространство пользователя. Как правило, личный кабинет в веб-сервисе необходим пользователю для быстрой и удобной работы с тем или иным сервисом: всё зависит от задачи, которая стоит перед пользователем. В интернет-магазине, например, пользователю необходимо как можно скорее заказать на дом те или иные товары. И поскольку такие необходимые данные, как его адрес и данные банковской карты, уже хранит в себе его личный кабинет (как, например, это делает личный кабинет на сайте Wildberries, показанный на рисунке 1.3), остаётся только нажать кнопку заказа, что, несомненно, очень удобно для пользователя.

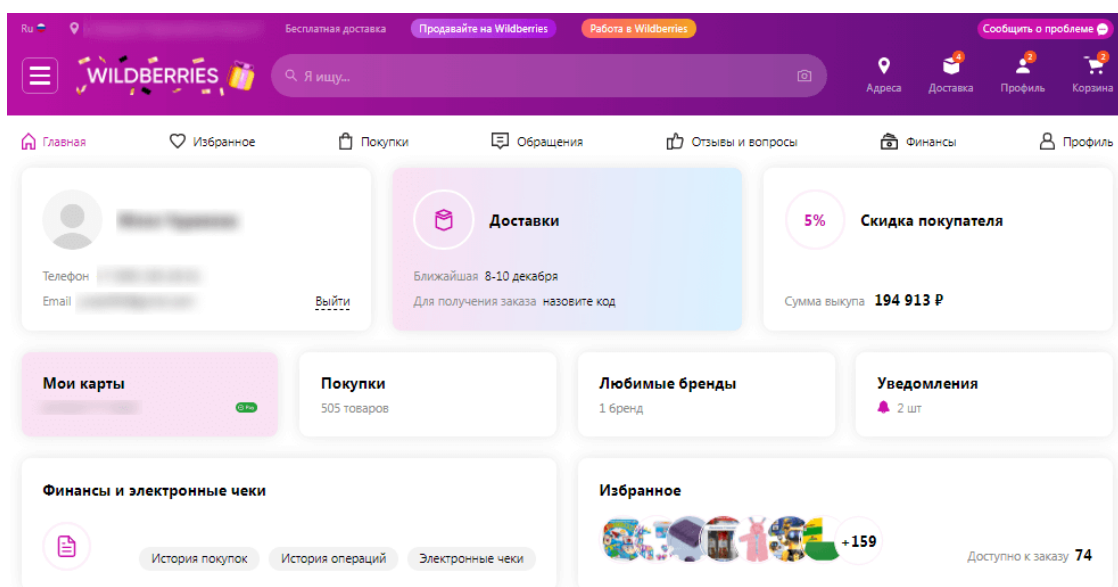


Рисунок 1.3 — Личный кабинет пользователя на Wildberries

Маркетинг. Имея личные данные пользователя, помимо генерации персональных предложений, можно использовать обширные возможности сквозной аналитики, персонализируя рекламу для пользователя и отправляя ему рассылки по электронной почте.

Разумеется, подобные возможности у владельцев и администрации веб-сервиса присутствуют и без личного кабинета, но в таком случае они будут очень ограничены и привязаны больше к IP-адресу пользователя. Личный кабинет позволит отслеживать поведение каждого пользователя без привязки к IP-адресу, что, несомненно, облегчит задачу. Конечно, стоит отметить, что задача маркетинга стоит больше перед интернет-магазинами, нежели перед, например, учебными веб-сервисами, но о такой задаче тоже было необходимо упомянуть.

К сожалению, в данной выпускной квалификационной работе бакалавра не рассматриваются вопросы безопасности личных кабинетов, возможные утечки баз данных с личными данными клиентов и последствия подобных ситуаций. Предполагается, что результаты данной работы, показанные в главе 2, будут использоваться исключительно в целях удобства и комфорта пользователей.

Стоит отметить, что вышеописанные задачи лишь частично описывают возможности личных кабинетов, поскольку, разумеется, есть множество

программных реализаций личных кабинетов, устроенных намного сложнее и имеющих многоступенчатую структуру и большое количество разнообразных возможностей. Но поскольку в данной выпускной квалификационной работе бакалавра личный кабинет призван выполнять ограниченное количество задач, подобное следует считать допустимым.

Для того, чтобы создать личный кабинет, удовлетворяющий пользователя, необходимо чётко понимать проблемы, которые испытывает пользователь с отсутствием такового либо с использованием конкурентного сервиса, его цели, желание и пр. Любая задача, решение которой берёт на себя разрабатываемый веб-сервис, должна решаться для пользователя быстро и легко. Для этого необходимо продумать структуру сервиса так, чтобы поиск необходимой информации и функций сервиса не отнимал у пользователя много времени и сил, а также облегчал возможность интуитивно ориентироваться на сайте.

Как правило, при создании веб-сервиса часто заимствуются и применяются уже готовые решения, что, на первый взгляд, выглядит логично, ведь это существенно облегчает разработку. С другой стороны, множество этих готовых решений находятся в открытом доступе, что может повлечь за собой плохие последствия, например, утечку персональных данных пользователя.

Стоит отметить, что даже при удобстве личного кабинета, когда его возможности отвечают всем желаниям пользователя, может потребоваться дополнительная мотивация для того, чтобы пользователь всё же создал свой личный кабинет в определённом веб-сервисе. Некоторые сервисы предоставляют доступ к полному спектру своих возможностей только после авторизации, а до неё - лишь к ограниченному, обещают какие-либо выгодные либо интересные предложения для пользователя или максимально упрощают процесс регистрации, предлагая привязать к личному кабинету аккаунт в социальной сети (например, Google).

В случае, если Московский авиационный институт решит воспользоваться результатами данной выпускной квалификационной работы бакалавра и внедрить решение, описываемое в главе 2, мотивация для регистрации не будет необходима, поскольку посторонний не будет иметь возможности зарегистрироваться, а логины и пароли будут выдаваться администрацией.

1.7 Анализ объекта исследования

Как было неоднократно отмечено выше, с цифровизацией образовательного процесса необходимо иметь удобный и понятный инструмент для фиксации результатов экзаменов и/или лабораторных работ, чтобы в дальнейшем можно было обратиться к ним.

Как уже было отмечено выше, у Московского Авиационного института есть целая система веб-сервисов, отвечающих за различные сферы жизни и деятельности учреждения. Основным является домен `mai.ru`, а на доменах уровнем ниже располагаются разные разделы. Например, по адресу `store.mai.ru` находится интернет-магазин вещей с символикой МАИ, по адресу `lms.mai.ru` располагается учебная система типа LMS (от англ. Learning Management System), в которой преподаватели могут выдавать студентам задания, проверять их выполнение, выставлять оценки и даже проводить занятия с помощью специального подсервиса BigBlueButton, а по адресу `my.mai.ru` находится личный кабинет студента, который можно видеть на рисунке 1.4 и в котором есть возможность увидеть оценки за курсы по предметам, своё расписание, семестровый план и многое другое.

The screenshot shows the student portal interface. On the left is a sidebar with a user profile for 'Тимофей Бердин' and a menu with options like 'Образование', 'Профиль', 'Оценки', 'Расписание', 'Семестровый план', 'Учебная часть', and 'Сервисы'. The main area is titled 'Табель успеваемости' (Academic Performance Table) and displays student information: Group M80-4076-18, Status 'Продолжает обучение', and a table of courses with grades.

Курс	Семестр	Дисциплина	Форма сдачи	Оценка
4	8	Безопасность жизнедеятельности	Эч	Эч
4	8	Вычислительные сети и телекоммуникации	Р	5
4	8	Математическая экономика	Эч	Эч
4	8	Математическое моделирование и вычислительный эксперимент	Р	5
4	8	Математическое моделирование и вычислительный эксперимент	КР	5

Рисунок 1.4 — Главная страница портала `my.mai.ru`

Но стоит отметить, что, хоть у преподавателей и есть личный кабинет в системе LMS МАИ, главная страница которой показана на рисунке 1.5, он зачастую вызывает у них затруднения в работе. В ходе неоднократных личных

бесед было выяснено, что преподавателям бывает трудно разобраться в функциях, которые доступны в личном кабинете, в том, как выставлять оценки, в том, как проводить занятия. Нередко отмечалось, что преподаватели стараются пользоваться набором сторонних веб-сервисов, используя, например, для учёта оценок Google Таблицы, а для проведения занятий такие платформы, как ZOOM или Google Meetings.

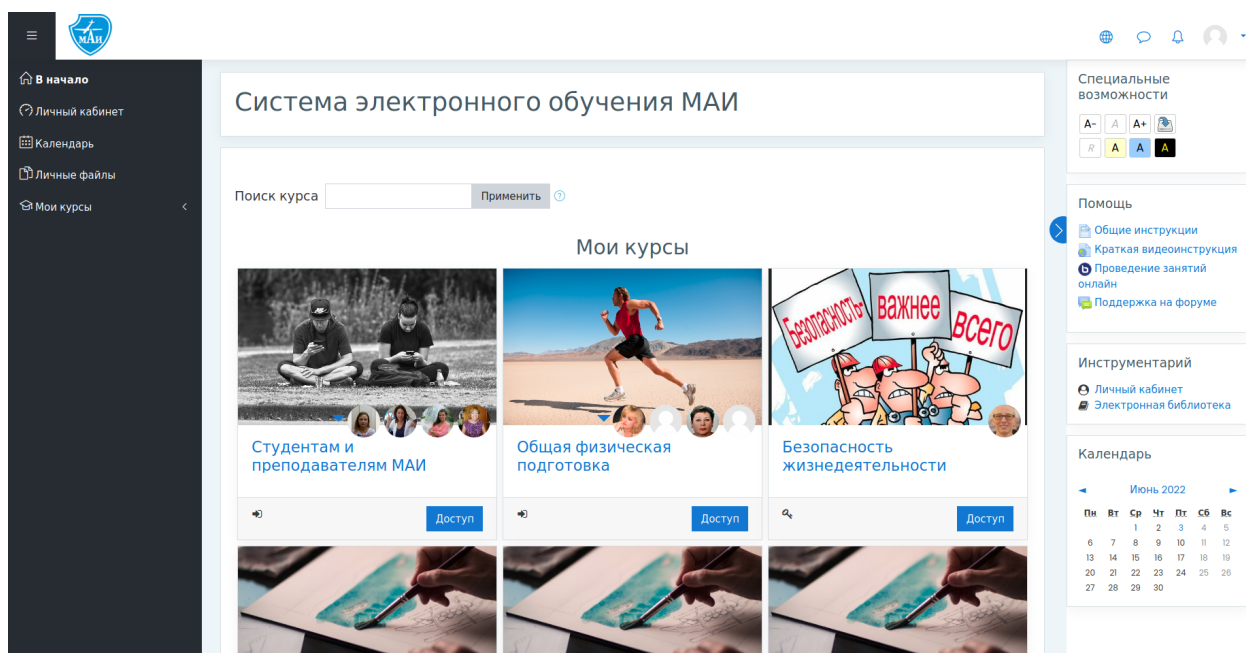


Рисунок 1.5 — Главная страница портала lms.mai.ru

Был отмечен и ряд других проблем, менее значительных, но от этого не теряющих своей актуальности, например, сложность навигации по системе веб-сервисов МАИ без специальной инструкции.

Очевидно, рассматриваемая в данной ВКР проблема актуальна, и её предлагается решить созданием прототипа веб-сервиса, отвечающего за личный кабинет преподавателя, который был бы прост в изучении и интуитивно понятен, а также собирал бы в себе все необходимые преподавателю функции.

Цель работы определена. Рассмотрим возможные подходы к задачам.

1.8 Анализ поставленных задач

Как правило, преподавателю необходимы следующие функции, которые должны присутствовать в веб-сервисе:

- просмотр расписания, добавление, изменение и удаление занятий;

- просмотр журналов групп, удобное добавление, изменение и удаление оценок по пятибалльной шкале;
- просмотр домашних заданий и их оценивание;
- удобная постраничная навигация и минималистичный интерфейс;
- удобная связь со сторонними вспомогательными веб-сервисами.

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Методы и средства решения задач

Рассмотрим методы и средства, выбранные для решения задач, описанных в конце главы 1.

Архитектура веб-сервиса имеет классическую клиент-серверную структуру, как показано на рисунке 2.1. Клиент обращается к серверу, выполняя запрос, сервер же, обрабатывая данный запрос, обращается к базе данных, берёт необходимую информацию и отправляет информацию обратно на клиент.

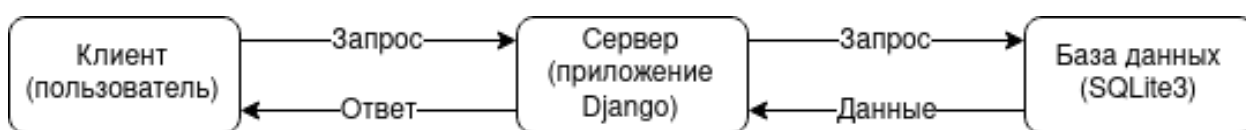


Рисунок 2.1 — Архитектура приложения

В качестве СУБД была выбрана компактная встраиваемая SQLite 3. Такой выбор обоснован тем, что движок данной СУБД - не отдельно работающий процесс, а библиотека, которая встраивается в основную программу и таким образом становится её составной частью. Из-за этого в качестве протокола обмена используется API библиотеки, что позволяет сильно упростить саму программу и уменьшает затрачиваемый ресурс и время отклика. Также стоит отметить, что SQLite хранит всю базу данных, схема которой показана на рисунке 2.2, в одном файле на том устройстве, где выполняется программа.

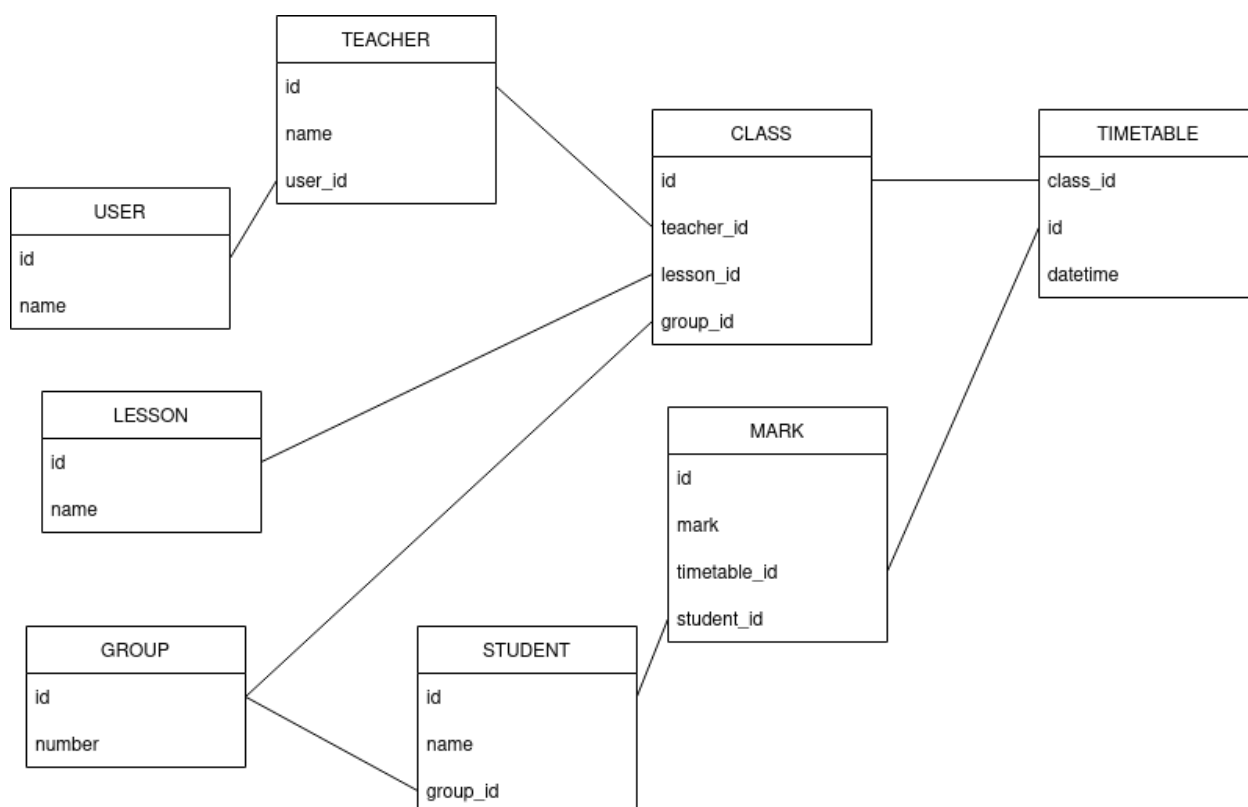


Рисунок 2.2 — Архитектура базы данных

Бэкенд веб-сервиса написан на языке программирования Python версии 3.8.8. Данный язык программирования был разработан в 1991 голландским программистом Гвидо ван Россумом с целью создания удобного скриптового объектно-ориентированного языка. Отличительной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как С или С++. Был выбран, в первую очередь из-за довольно широких возможностей благодаря большому количеству написанных библиотек, а также из-за простого и удобного в использовании синтаксиса.

Стоит уточнить, что для создания бэкенда веб-сервиса использовался, в большей степени, именно фреймворк Django 4.0.3. Данный фреймворк вышел в релиз в 2008 году и был создан группой разработчиков, использовавших

повторяющиеся шаблоны и функции, которые было в итоге решено объединить. Выбор именно этого фреймворка обоснован его удобством в создании, запуске сайта, а также его администрировании и управлении.

Подробный листинг кода можно посмотреть в приложении А.

Здесь стоит остановиться на архитектуре, которая используется в Django - шаблоне MVT (Model-View-Template).

Как правило, структура MVT состоит из трёх частей:

- model (модель) - часть, выступающая в качестве интерфейса данных и отвечающая за их ведение, представленная обычно базой данных;
- view (представление) - часть, отвечающая за пользовательский интерфейс;
- template (шаблон) - часть, состоящая из статических частей HTML и некоторого синтаксиса, описывающего вставку динамических частей.

Преимущество такой архитектуры заключается в гибкости и простоте модификации различных элементов веб-сервиса.

Фронт веб-сервиса разработан преимущественно с использованием языка гипертекстовой разметки HTML. Как было упомянуто в разделе 1.1, данный язык был создан в 1991 году Тимом Бернерсом-Ли с целью, изначально подразумевающей облегчение процесса вёрстки, как средство структурирования и форматирования документов без привязки к средствам отображения. С помощью HTML можно быстро и просто создать красиво оформленный документ.

Важно дополнить, что для оформления единого визуального стиля веб-сервиса были использованы технология CSS и набор шаблонов и стилей Bootstrap, что позволило дизайну веб-сервиса стать удобным и дружелюбным к пользователю.

Также стоит упомянуть, что для дизайна веб-сервиса были сформулированы следующие требования в соответствии со стандартами W3C (World Wide Web Consortium) [1]: "

- дизайн сайта должен быть понятным, удобным и корректно отображаться в любом браузере, а сам сайт должен иметь оригинальный графический дизайн, обеспечивающий надлежащую функциональность;
- на страницах сайта должны быть обеспечены удобство и полнота на-

вигационных элементов;

- основной контент, меню и другие элементы навигации должны быть доступны без горизонтального прокручивания полностью открытого окна браузера при разрешении экрана 1024x768 точек;
- дизайн (визуальное оформление, расположение информационных блоков) главной страницы может отличаться от дизайна типовых страниц с сохранением стилового единства сайта;
- приветствуется наличие красивых графических объектов, использование небольших, но стильных графических элементов в оформлении страниц сайта;
- дизайн сайта должен быть достаточно «легким» по объему графических элементов и обеспечивать как можно большую скорость загрузки страниц сайта;
- необходима легкая идентификация раздела и подраздела, в котором находится пользователь;
- дизайн должен предусматривать возможность развития структуры сайта за счет разработки и включения в ее состав новых разделов и подразделов."

Страницы веб-сервиса, рассматриваемого в данной выпускной квалификационной работе бакалавра, созданы с учётом всех вышеперечисленных требований.

2.2 Файловая структура проекта

Рассмотрим файловую структуру проекта, чтобы понимать, за что отвечает тот или иной написанный код. С полным листингом кода можно ознакомиться в приложении А.

Рассмотрим основные файлы проекта.

Файл `manage.py`, как правило, отвечает за взаимодействие с веб-сервисом через консоль (командную строку). С его помощью можно запускать и останавливать сервер, запускать миграцию баз данных и синхронизировать их, создать суперпользователя и т.д.

Файл `settings.py` содержит все настройки веб-сервиса: регистрацию всех приложений, конфигурацию баз данных, настройки даты и времени, располо-

жение статических файлов и т.д.

Файл `urls.py` содержит в себе все ссылки проекта и функции для их вызова.

Файл `wsgi.py` предназначен для связи приложения с сервером

Файл `db.sqlite3` содержит базу данных SQLite3, принцип работы которой описан в разделе 2.1.

2.3 Решение задачи создания веб-сервиса

Итак, готовый проект представляет собой веб-приложение, бэкенд которого написан на языке Python, а именно с использованием фреймворка Django, фронт - с использованием языка HTML, таблицы стилей CSS, и набора шаблонов Bootstrap, а база данных составлена с использованием СУБД SQLite3.

Рассмотрим полученные результаты.

Загрузив сайт, пользователь попадает на страницу, где ему будет предложено авторизоваться, как видно на рисунке 2.3. Стоит отметить, что самостоятельная регистрация пользователей не предусматривается - в базу данных их добавляет администратор сайта.

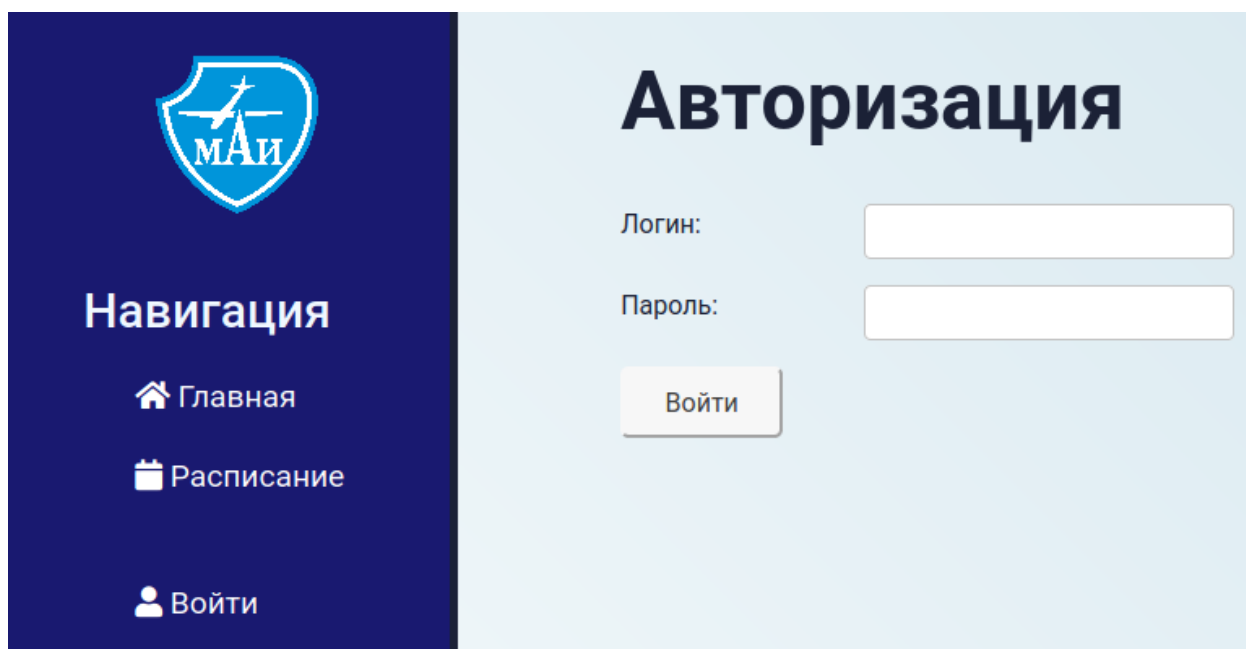


Рисунок 2.3 — Страница авторизации пользователя

Разумеется, если пользователь введёт неверные данные, ему будет выведена соответствующая ошибка, которую можно видеть на рисунке 2.4.

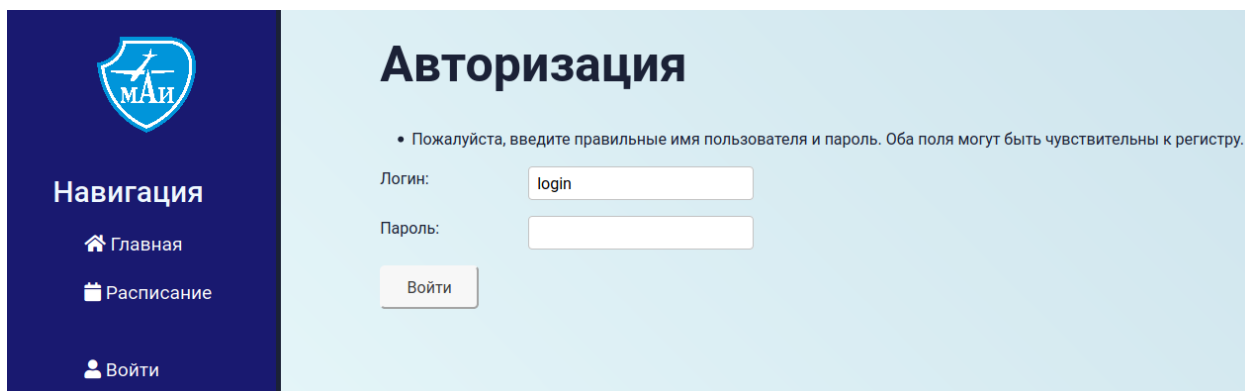


Рисунок 2.4 — Страница авторизации после ввода неверных данных

Преподаватель может перейти на страницу со своим расписанием, которое можно видеть на рисунке 2.5, и посмотреть даты и время своих занятий, предметы, которые он ведёт, группы, тип занятия и аудитории.

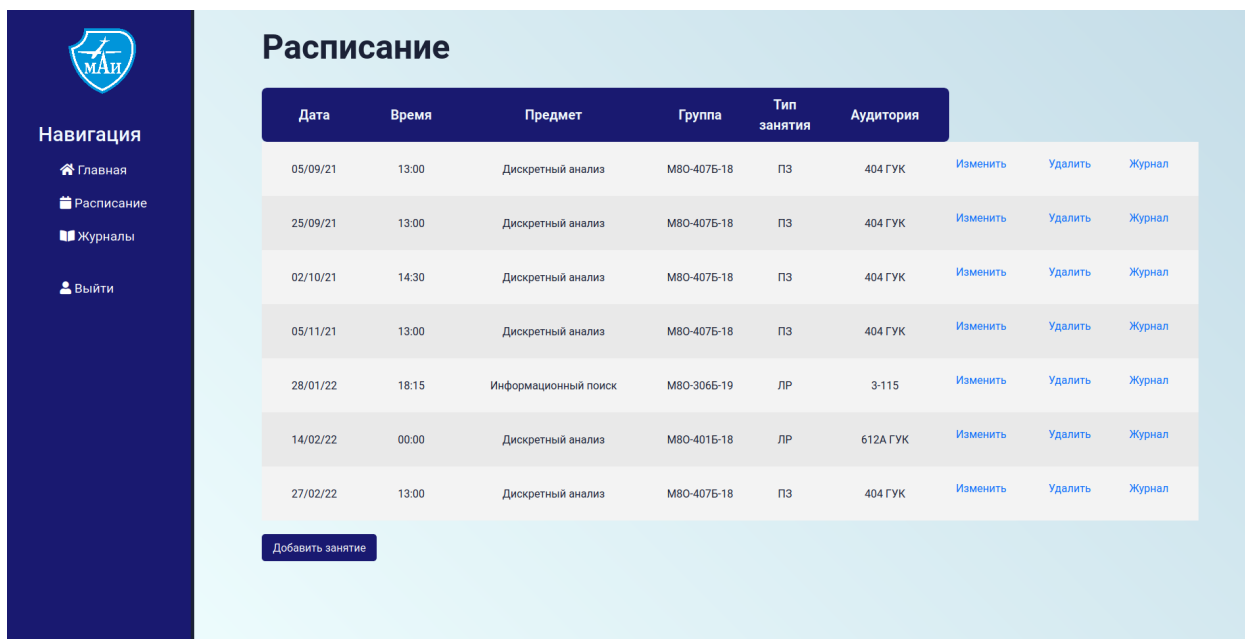



Рисунок 2.5 — Страница с расписанием для преподавателей

На эту страницу может зайти и неавторизованный пользователь, как можно видеть на рисунке 2.6. В таком случае, ему будут недоступны функции добавления, изменения и удаления занятия из расписания, о которых будет рассказано ниже.



Навигация

Главная

Расписание

Войти

Расписание

Дата	Время	Предмет	Группа	Тип занятия	Аудитория	Преподаватель
05/09/21	13:00	Дискретный анализ	M80-407Б-18	ПЗ	404 ГУК	Кухтичев Антон Алексеевич
25/09/21	13:00	Дискретный анализ	M80-407Б-18	ПЗ	404 ГУК	Кухтичев Антон Алексеевич
02/10/21	14:30	Дискретный анализ	M80-407Б-18	ПЗ	404 ГУК	Кухтичев Антон Алексеевич
05/11/21	13:00	Дискретный анализ	M80-407Б-18	ПЗ	404 ГУК	Кухтичев Антон Алексеевич
28/01/22	18:15	Информационный поиск	M80-306Б-19	ЛР	3-115	Кухтичев Антон Алексеевич
14/02/22	00:00	Дискретный анализ	M80-401Б-18	ЛР	612А ГУК	Кухтичев Антон Алексеевич
10/05/22	00:00	Дискретный анализ	M80-407Б-18	ПЗ	7-401	Кухтичев Антон Алексеевич

Рисунок 2.6 — Страница с расписанием для неавторизованных

Преподаватель может добавлять занятие в своё расписание, указывая дату и время, а также аудиторию, и выбирая предмет и группу в выпадающем списке, что показано на рисунке 2.7.


<div>  </div> <div> Навигация <div> <div>Главная</div> <div>Расписание</div> <div>Журналы</div> <div>Выйти</div> </div> </div>	<div>Добавление занятия</div> <div> <div>Дата и время:</div> <div>10.05.22</div> </div> <div> <div>Аудитория:</div> <div>7-401</div> </div> <div> <div>Занятие:</div> <div>Дискретный анализ, M80-407Б-18</div> </div> <div> <div>Добавить</div> </div>
---	---

Рисунок 2.7 — Добавление занятия в расписание

При необходимости перенести дату и/или время занятия преподаватель может использовать соответствующую функцию, как видно на рисунке 2.8.

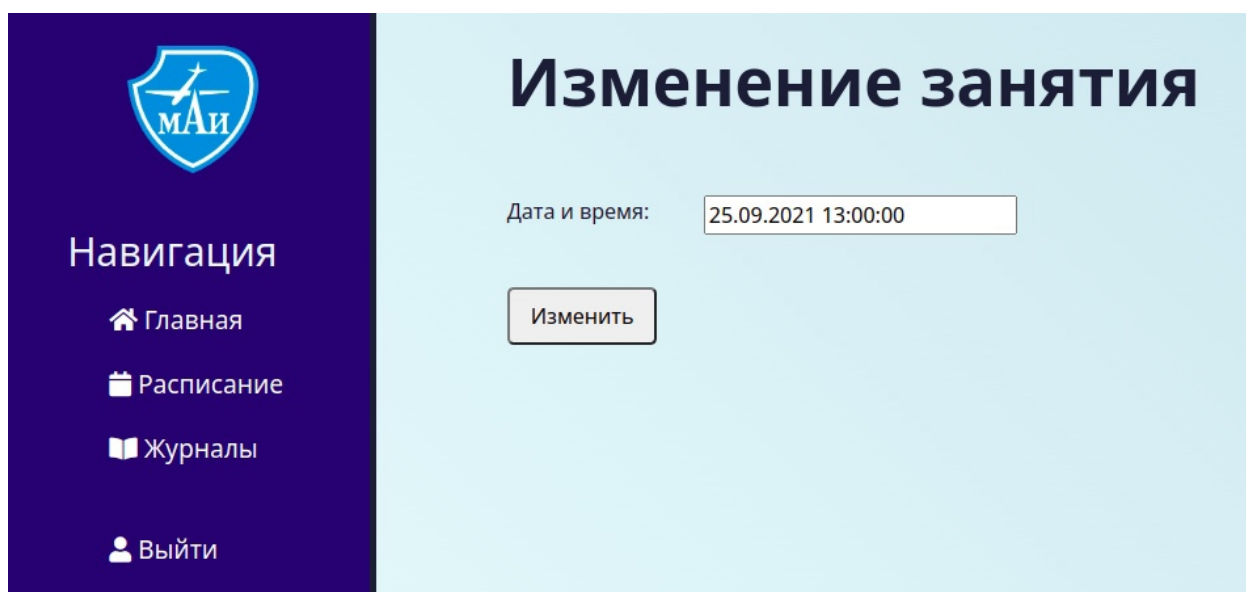


Рисунок 2.8 — Изменение занятия в расписании

При необходимости полностью удалить занятие из расписания преподаватель может сделать и это, как видно на рисунке 2.9.

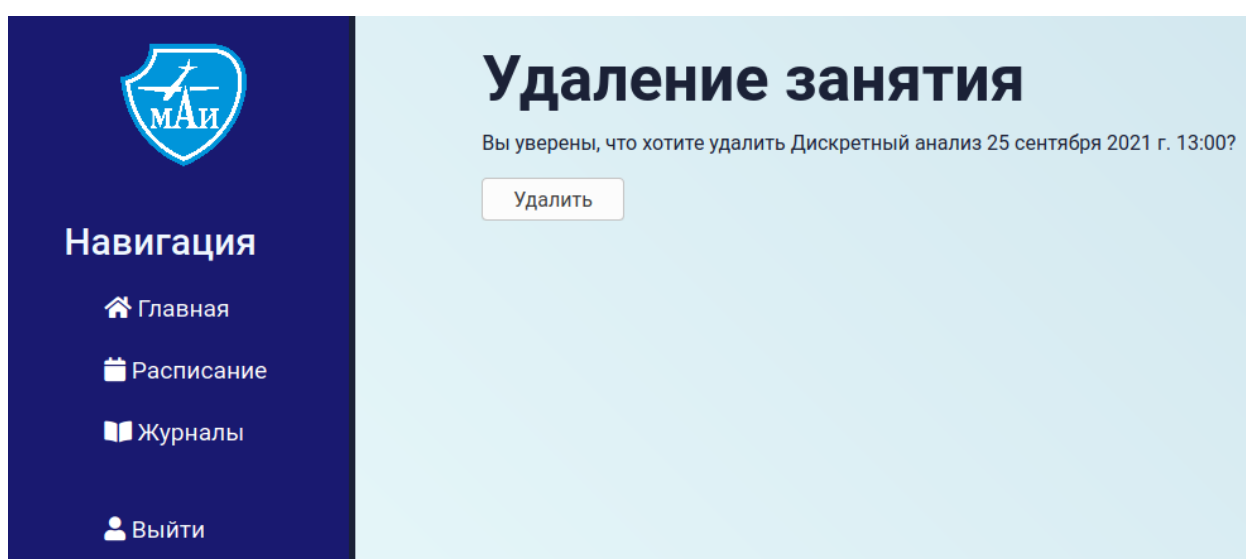


Рисунок 2.9 — Удаление занятия из расписания

Также следует обратить внимание на то, что преподаватель может добавить ссылку на дистанционное занятие, как видно на рисунке 2.10.

Добавление занятия

Дата и время: 28.01.22 18:15

Аудитория: 7-401

Занятие: Дискретный анализ, M80-401Б-18

Ссылка: <https://us05web.zoom.us/j/8>

Добавить

Рисунок 2.10 — Добавление ссылки на дистанционное занятие

Затем это будет отображаться как гиперссылка, кликнув по которой, пользователь сможет попасть на дистанционное занятие, организованное преподавателем.

Также преподавателю доступен список журналов групп, как можно заметить по рисунку 2.11.

Журналы

Дискретный анализ

- [M80-407Б-18](#)
- [M80-401Б-18](#)

Информационный поиск

- [M80-306Б-19](#)

Рисунок 2.11 — Список доступных журналов групп

Преподаватель может выбрать журнал группы, где можно ознакомиться

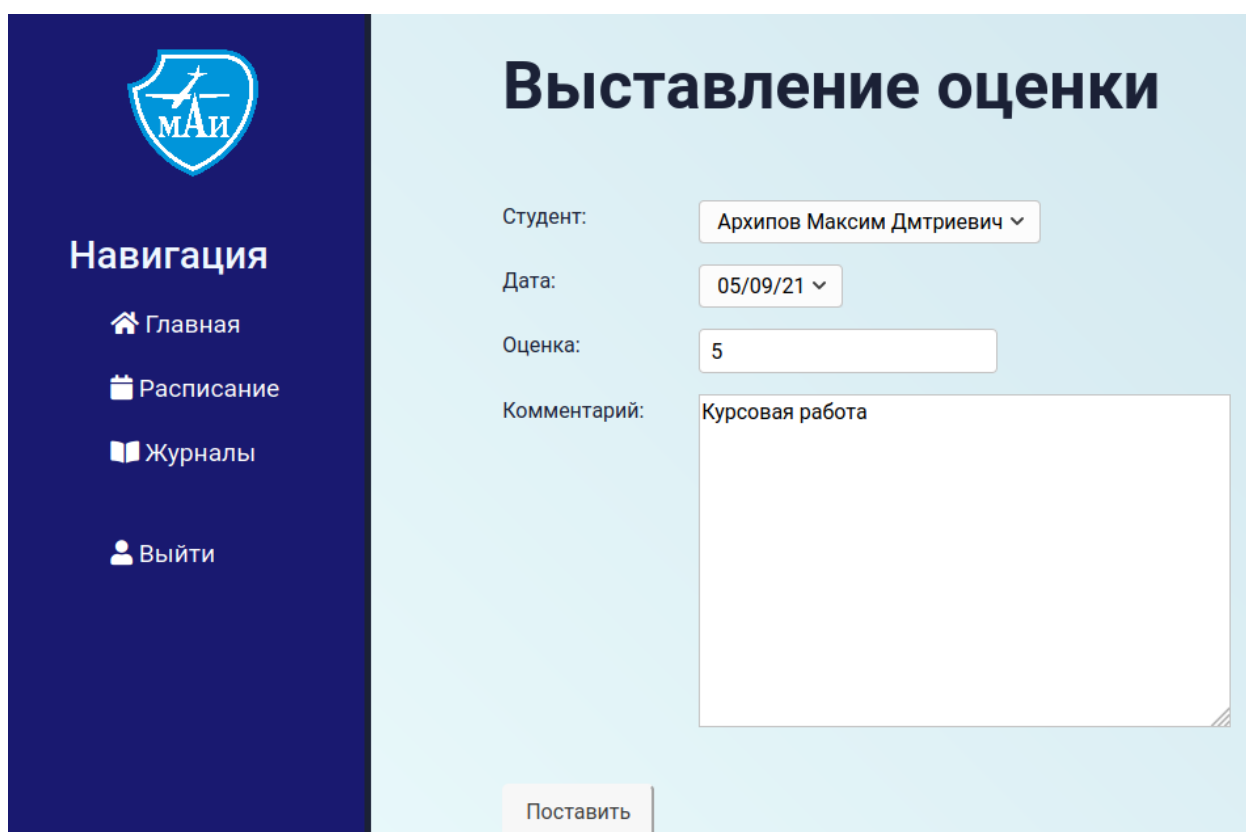
со списком и оценками студентов. Журнал показан на рисунке 2.12.



	05/09	25/09	02/10	05/11	27/02
Архипов Максим Дмитриевич	±	±	±	±	±
Бердихин Тимофей Алексеевич	5	4	±	±	±
Васильев Максим Игоревич	±	±	±	±	±
Егорова Анна Владимировна	±	±	±	±	±
Полей-Добронравова Амелия Вадимовна	5	3	3	±	±
Син Денис Дмитриевич	5	±	4	±	±

Рисунок 2.12 — Журнал группы

Разумеется, у преподавателя присутствует возможность выставления оценок, что видно на рисунке 2.13.



Выставление оценки

Студент:

Дата:

Оценка:

Комментарий:

Рисунок 2.13 — Добавление оценки в журнал

Также, если оценку необходимо исправить или удалить, такая возможность тоже присутствует. Это показано на рисунке 2.14.

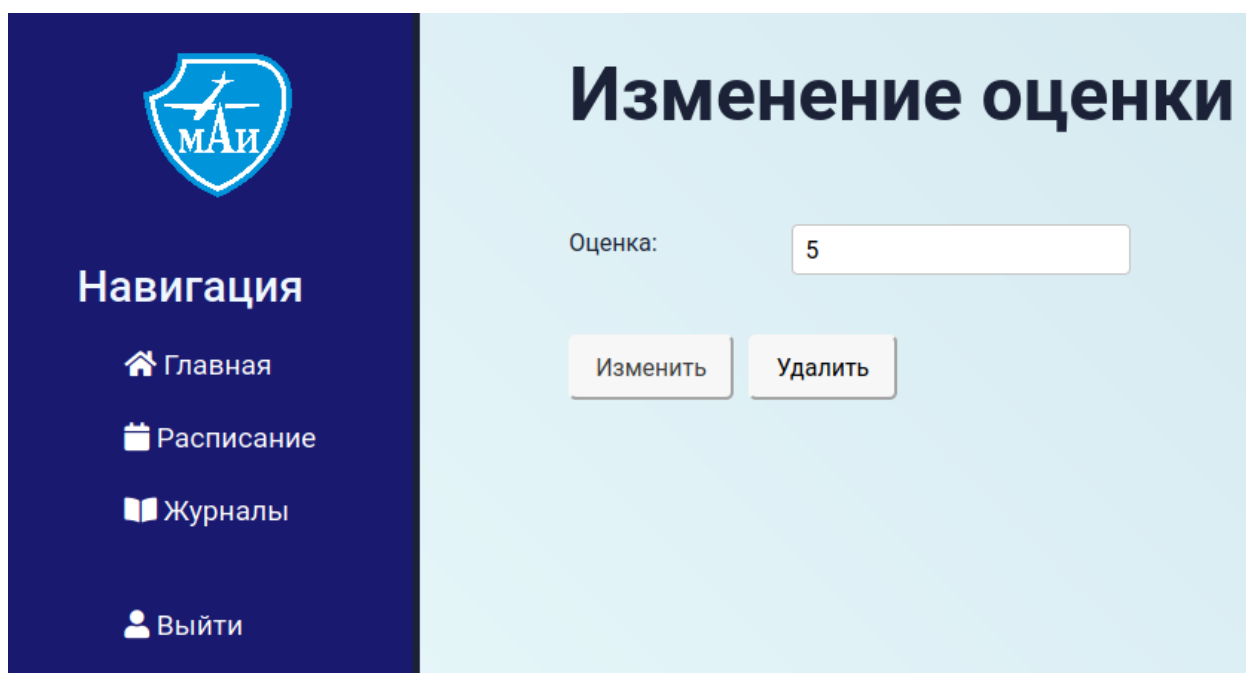


Рисунок 2.14 — Изменение и удаление оценки в журнале

При удалении оценки, чтобы исключить фактор ошибки, у преподавателя запрашивается подтверждение, что можно увидеть на рисунке 2.15.

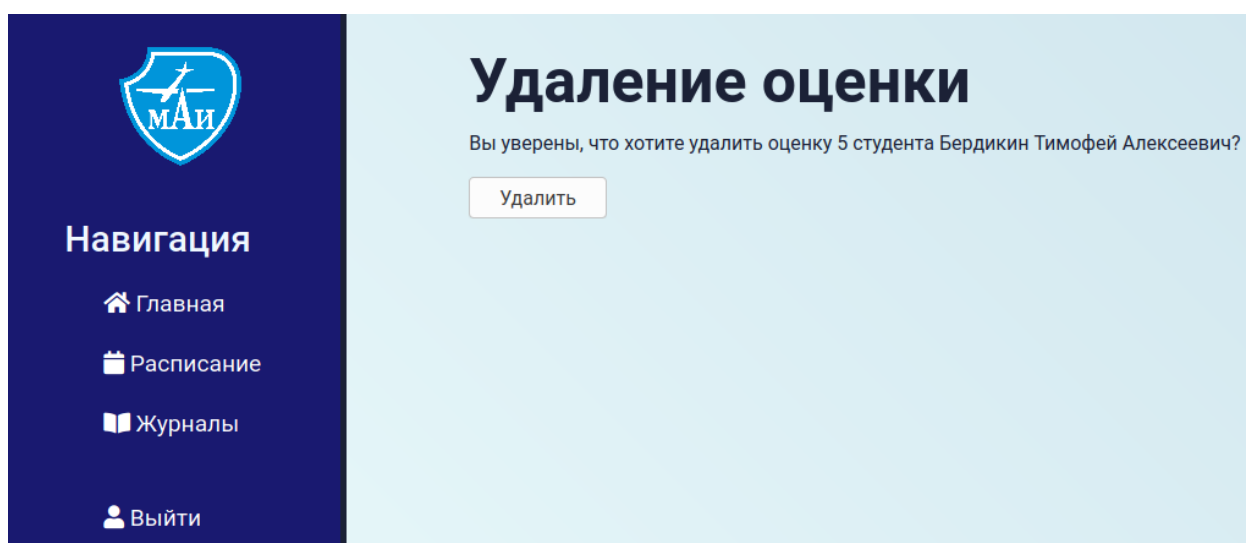


Рисунок 2.15 — Подтверждение удаления оценки в журнале

ЗАКЛЮЧЕНИЕ

Результатом выпускной квалификационной работы бакалавра является прототип веб-сервиса для преподавателей.

Была проведена классификация веб-сервисов по предоставляемой информации, их доступности, цели и технологии создания. Был проведён анализ недостатков веб-сервисов МАИ и предложено решение актуальной проблемы разработки личного кабинета преподавателя.

Была разработана программа, реализующая предложенное решение. Полученный результат показывает эффективность предложенного подхода к личному кабинету. Полученная программа является завершённой, однако имеет высокий потенциал к развитию.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Brooks T. World wide web consortium (W3C) //Encyclopedia of Library and Information Sciences, Third Edition. – CRC Press, 2009. – С. 5695-5699.
2. Holovaty A., Kaplan-Moss J. The Django Administration Site //The Definitive Guide to Django. – Apress, 2008. – С. 83-94.
3. Gardner J. The web server gateway interface (wsgi) //The Definitive Guide to Pylons. – 2009. – С. 369-388.
4. Robbins J. N. Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. – "O'Reilly Media, Inc. 2012.
5. Adithela S. P. et al. Django content management system evaluation and integration with apache airavata //Proceedings of the Practice and Experience on Advanced Research Computing. – 2018. – С. 1-4.
6. Bertocco M. et al. A client-server architecture for distributed measurement systems //IEEE transactions on instrumentation and measurement. – 1998. – Т. 47. – №. 5. – С. 1143-1148.
7. Oluwatosin H. S. Client-server model //IOSRJ Comput. Eng. – 2014. – Т. 16. – №. 1. – С. 2278-8727.
8. Duckett J. HTML & CSS: design and build websites. – Indianapolis, IN : Wiley, 2011. – Т. 15.
9. Банкожитенко Е. В., Власова С. В., Зубанова С. Г. Особенности научно-педагогических подходов при обучении иностранному языку студентов технических факультетов МАИ (НИУ) //Мир науки. Педагогика и психология. – 2017. – Т. 5. – №. 2. – С. 3.
10. Overview of Design Principles // World Wide Web Consortium URL: <https://www.w3.org/2004/10/wcag2-nav/wcag20princ.html> (дата обращения: 31.05.2022).
11. Основные компоненты Web-страницы // Лекции.Орг URL: <https://lektsii.org/6-19817.html> (дата обращения: 29.04.2022).
12. Kitova O. V., Kitov V. A. Anatoly Kitov and Victor Glushkov: Pioneers of Russian Digital Economy and Informatics //IFIP International Conference on the History of Computing. – Springer, Cham, 2018. – С. 99-117.

13. Cohen-Almagor R. Internet history //Moral, ethical, and social dilemmas in the age of technology: Theories and practice. – IGI Global, 2013. – C. 19-39.
14. Zhang J. et al. A Web Site Classification Approach Based On Its Topological Structure //Int. J. Asian Lang. Process. – 2010. – T. 20. – №. 2. – C. 75-86.
15. Bhosale S. T., Patil M. T., Patil M. P. Sqlite: Light database system //Int. J. Comput. Sci. Mob. Comput. – 2015. – T. 44. – №. 4. – C. 882-885.
16. Forcier J., Bissex P., Chun W. J. Python web development with Django. – Addison-Wesley Professional, 2008.
17. Spurlock J. Bootstrap: responsive web development. – "O'Reilly Media, Inc. 2013.
18. Dauzon S., Bendoraitis A., Ravindran A. Django: web development with Python. – Packt Publishing Ltd, 2016.
19. Taneja S., Gupta P. R. Python as a tool for web server application development //JIMS8I-International Journal of Information Communication and Computing Technology. – 2014. – T. 2. – №. 1. – C. 77-83.
20. Duckett J. Web design with HTML, CSS, JavaScript and jQuery set. – IN : Wiley, 2014. – T. 1.
21. Grannell C. The Essential Guide to CSS and HTML Web Design. – Friends of ED, 2007.
22. Erwin P. M. Corporate codes of conduct: The effects of code content and quality on ethical performance //Journal of Business Ethics. – 2011. – T. 99. – №. 4. – C. 535-548.
23. del Pilar Salas-Zárate M. et al. Analyzing best practices on Web development frameworks: The lift approach //Science of Computer Programming. – 2015. – T. 102. – C. 1-19.
24. Plekhanova J. Evaluating web development frameworks: Django, Ruby on Rails and CakePHP //Institute for Business and Information Technology. – 2009. – T. 20. – C. 2009.
25. Rubio D. Django admin Management //Beginning Django. – Apress, Berkeley, CA, 2017. – C. 495-547.
26. Tsoukalos M. Using Django and MongoDB to build a blog //Linux Journal. – 2014. – T. 2014. – №. 238. – C. 3.

27. Garcia-Molina H., Ullman J. D., Widom J. Database system implementation. – Upper Saddle River : Prentice Hall, 2000. – T. 672.
28. Bryon B. django-confit Documentation. – 2015.
29. Sydow D., Rodríguez-Guerra J., Volkamer A. OpenCADD-KLIFS: A Python package to fetch kinase data from the KLIFS database //Journal of Open Source Software. – 2022. – T. 7. – №. 70. – C. 3951.

ПРИЛОЖЕНИЕ А

Разработанный код

Ссылка на хранилище с кодом: <https://github.com/Berdikin/bachelor-diploma>