



МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖИ РЕСПУБЛИКИ
КРЫМ
ГБОУВО РК «КРЫМСКИЙ ИНЖЕНЕРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ ФЕВЗИ ЯКУБОВА»

Факультет экономики, менеджмента и информационных технологий
Кафедра прикладной информатики

Направление подготовки 09.03.03 Прикладная информатика
Профиль «Прикладная информатика в информационной сфере»

КУРСОВОЙ ПРОЕКТ

по дисциплине «Программирование и поддержка веб-приложений»

НА ТЕМУ

**Разработка веб-приложения «Система публикации научных работ
и достижений преподавателей»**

Студента IV курса
группы И–2–18
очной формы обучения
Абибулаев Ленур Бахтиярович

— (подпись)

Руководитель:
ст. преп. Абдураманов Зиннур
Шевкетович

— (подпись)

Симферополь – 2022

АННОТАЦИЯ

Абибулаев Л. Б. Разработка веб-приложения «Система публикации научных работ и достижений преподавателей»

В курсовом проекте представлена технология разработки веб-приложения для публикации научных работ и достижений преподавателей. Приложение разработано с использованием языка программирования Python на базе технологий Apache (HTTP Server), JavaScript, MySQL. Это приложение может успешно применяться в любых учебных заведениях, целью которых создать библиотеку научных работ преподавателей .

Ключевые слова: JavaScript, Apache (HTTP Server), MySQL, веб-приложение, Python.

ANNOTATION

Abibulaev L. B. Development of a web application "System of publication of scientific works and achievements of teachers"

The course project presents the technology for developing a web application for the publication of scientific works and achievements of teachers. The application was developed using the Python programming language based on Apache (HTTP Server), JavaScript, MySQL technologies. This application can be successfully used in any educational institutions whose goal is to create a library of scientific works of teachers.

Keywords: JavaScript, Apache (HTTP Server), MySQL, web application, Python.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ГЛАВА I. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ СОЗДАНИЯ WEB-САЙТОВ	6
1.1 Общая характеристика, классификация Web-сайтов.	6
1.2 Этапы создания Web-сайта	9
1.3 Средства разработки Web – сайта	15
1.4 Системный анализ предметной области	18
1.5 Проектирование баз данных	21
Выводы к главе 1	23
ГЛАВА 2. РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ «СИСТЕМА ПУБЛИКАЦИИ НАУЧНЫХ РАБОТ И ДОСТИЖЕНИЙ ПРЕПОДАВАТЕЛЕЙ»	23
2.1. Реализация программного продукта	23
2.2. База данных программного продукта	28
2.3. Описание интерфейса приложения	32
Вывод к главе 2	35
ЗАКЛЮЧЕНИЕ.....	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	37

ВВЕДЕНИЕ

Актуальность темы. WWW - одна из самых распространенных и универсальных технологий в современном Интернете. Эта технология предоставляет пользователю целый ряд различных информационных услуг - информационных, развлекательных, бизнес-ориентированных и т. Д. Однако изначально эта технология была создана, чтобы облегчить ученым объединение и поиск информации. Свои странички в интернете есть у госструктур, разных компаний, школ. Учебное заведение, которое хочет иметь эффективную систему работы с информацией, чтобы создать эффективную систему управления учебным процессом. Чтобы решить эту проблему, вы можете создать сайт, на котором будут публиковаться достижения и научные работы учителей. Это позволит создать библиотеку педагогических работ, которой смогут пользоваться как преподаватели, так и студенты, которым интересна тематика присланных научных работ.

Целью курсовой работы является разработка веб сайта для публикации научных работ и достижений преподавателей.

Для достижения цели были поставлены следующие задачи:

- выполнить проектирование приложения и выбор программных инструментов;
- провести анализ предметной области;

Объектом курсовой работы является процесс разработки веб-сайта для публикации научных работ и достижений преподавателей.

Предметом курсовой работы является веб-сайт для публикации научных работ и достижений преподавателей.

Практическая значимость.

Структура курсовой работы. Курсовая работа бакалавра состоит из введения, двух глав, выводов, заключения, списка использованных источников.

Во введении представлена актуальность темы, цель, задачи, определены предмет и объект исследования, практическая значимость курсовой работы.

В первой главе содержатся теоретические аспекты в области создания Веб-сайтов. Описана хронология создания Web-сайта, проведен системный анализ предметной области, описаны основные этапы проектирования баз данных. Описаны инструменты используемые при разработке Веб-сайта.

Во второй главе приведен процесс реализации приложения, в частности – проектирование базы данных, а также сама разработка сайта.

В заключении приведены важнейшие результаты выполнения курсовой работы.

Общий объем бакалаврской работы составляет _ страниц. Работа содержит _ рисунков, _ листингов. Список использованных источников, включает _ наименований.

ГЛАВА I. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ СОЗДАНИЯ WEB-САЙТОВ

1.1 Общая характеристика, классификация Web-сайтов.

Веб-сайт (от английского «web-site»: «web» – паутина, сеть; «site» – место, местоположение) – это структурированная специальным образом информация, которая размещена на сервере и является открытой пользователям сети как для свободного, так и для авторизируемого или ограниченного доступа.

Совокупность всех сайтов составляет Всемирную паутину, объединяющую в единое целое части информации всего мирового сообщества – объединения планетарного масштаба и базу данных. Протокол HTTP (Hyper Text Transfer Protocol — протокол передачи гипертекста) специально был разработан для того, чтобы обеспечить клиенту прямой доступ к сайтам на серверах.[1]

Сайт является набором определенных файлов – страниц – определенного формата. Страницы сайтов – это файлы с текстом, который содержит команды на языке HTML (HyperText Markup Language – язык разметки гипертекста). Данные файлы при загрузке посетителем на компьютер обрабатываются посредством браузера и выводятся на средство отображения компьютера (монитор, экран смартфона, экран КПК и т.д.) Возможности языка HTML позволяют править, различать его функциональные элементы, создавать гиперссылки, а также включать мультимедийные элементы: изображения, звукозаписи и т.д.

Страницы сайтов не всегда представляют собой простой статичный набор файлов. Иногда они создаются при помощи особой компьютерной программы на сервере – на так называемом движке сайта. Программа может как быть написана специально для отдельного сайта на заказ, так и быть готовым продуктом, предназначенным для использования на сайтах определенного класса. Программы, называемые CMS (система управления

содержимым) обеспечивают владельцу сайта возможность гибкого регулирования вывода и структуризации информации на сайте. Некоторые из движков обеспечивают владельцу сайта возможность гибкой настройки вывода и структурирования информации на веб-сайте.[2]

Существует большое количество признаков, по которым можно произвести классификацию сайтов. Классификация сайтов является необходимой, так как зачастую веб-мастеру приходится сталкиваться с непониманием заказчика того, какой сайт должен получиться в итоге. Поэтому целесообразно выделить базовых признаков каждого вида (типа). Кроме того, у сайтов разного типа различная, специфическая структура.

В первую очередь, как правило, выделяют сайты коммерческие и некоммерческие. Конечной целью функционирования коммерческого сайта всегда является прибыль, в то время как некоммерческие сайты выполняют, в основном, информационную функцию. Коммерческим можно назвать не только сайт, предназначенный для осуществления продаж либо предоставления услуг, но и промо-сайты, popularизирующие бренд, товар или фирму. Некоммерческие сайты чаще всего создаются общественными организациями, политическими партиями, государственными организациями. Также некоммерческими сайтами можно назвать ресурсы, предназначенные для обмена мнениями и организации общения людей (форумы, блоги, сообщества).

Основные виды сайтов:

Блог (с английского *blog*, от *web log* – Internet-журнал, Internet-дневник, онлайн-дневник) – сайт, основным содержанием которого являются записи (посты), регулярно добавляемые и содержащие мультимедийные элементы.

Форум (в древнеримском градостроительстве – центральная городская площадь, на которой велись переговоры, заключались сделки, проходила общественная жизнь) – организует общение пользователей, информация на веб-форуме структурирована, предполагается наличие общей тематики и различных тем для обсуждения.

Wiki-ресурс – справочный ресурс, наполнение которого осуществляется пользователями. Кроме того, пользователи могут изменять структуру и содержимое wiki-ресурсов.

Flash-сайт – веб-ресурс, созданный с использованием платформы Adobe Flash. На сайтах такого вида могут быть реализованы нестандартные способы представления информации.

Доска объявлений – ресурс, на котором организуется прием и размещение объявлений от пользователей.

Визитка – сайт, содержащий только необходимую информацию о фирме, предпринимателе, краткое описание деятельности и контакты. Иногда на сайтах подобного вида размещается портфолио работ.

Сервис – сайты, предоставляющие некоторую статистику; позволяющие анализировать определенные показатели; почтовые и поисковые сервисы.

Портал (от англ. Portal – главный вход) – веб-сайт, предоставляющий пользователю разнообразные интернет-сервисы, работающие в пределах этого сайта. Порталы являются крупными сайтами и предполагают наличие обширной аудитории.

Интернет-магазин – сайт, продающий товары онлайн. По сути, он является совокупностью программ, позволяющих покупателю удаленно выбрать товар из каталога, а также оформить заказ на него.

Биржа – ресурс играет роль посредника между заказчиком и клиентом. Тематика сайта-биржи может быть разнообразной. Примером такого сайта можно назвать биржу статей ADVEGO.

Простой сайт – сайт, не относящийся к вышеперечисленным. К примеру, сайт члена известной музыкальной группы, имеющий несколько разделов и имеющий, в основном, информационную функцию.[3]

1.2 Этапы создания Web-сайта .

Независимо от того, какой сайт вы задумали – одностраничный или на 15-20 страниц – необходимо понимать, что он требует вложения времени, сил и финансов. Ведь это своего рода недвижимость – ваша частная собственность, строительством и обустройством которой заведуете вы лично. И ваша обязанность убедиться в том, что сайт получится как можно качественнее и удобнее для клиентов.

Основные этапы создания веб-сайта:

1. Планирование
2. Аналитика
3. Стратегия
4. Создание технического задания
5. Прототипирование
6. Создание макетов дизайна
7. Верстка
8. Программирование
9. Базовое наполнение
10. Тестирование
11. Техническая поддержка

Планирование сайта

Слишком много людей сразу начинают «забивать гвозди», не тратя времени на создание надлежащего плана. Но суть в том, что какую бы

работу вы не выполняли необходимо две трети времени потратить на планирование и только одну треть – на осуществление задуманного. Экономия времени на этапе планирования проекта приводит к непредвиденным ошибкам, из-за которых в результате приходится переделывать часть работы.[4]

1. Аналитика

Начать стоит с определения целевой аудитории и анализа конкурентов. Владелец сайта важно понимать свою целевую аудиторию, её потребности и боли. Тогда он сможет адаптировать свой сайт под их интересы и получить из этого собственную выгоду. Стоит найти ответы на вопросы:

- для кого предназначен веб-ресурс?
- каковы предпочтения ваших потенциальных клиентов?
- какая информация будет для них полезной и интересной?
- какими коммуникационными каналами они пользуются?

Пересмотрите своих прямых конкурентов, изучите их решения, найдите преимущества и недостатки. А потом используйте это в своих целях. Определите свои преимущества, свою изюминку, которая будет вас отличать и показывать в выгодном свете.

2. Стратегия

Определите цель вашего сайта, чего вы хотите достичь с помощью него и какие инструменты вам в этом помогут. Цели у бизнес-ресурса могут быть абсолютно разные:

- продавать (для e-commerce);
- привлекать клиентов;
- презентовать продукт или линейку товаров;
- рассказать об услуге;

- собирать лиды (контактные данные);
- рассказать о вашей компании;
- выстроить долгосрочные отношения с клиентами;
- информировать ЦА;
- повысить охват и узнаваемость бренда, используя digital-каналы.

Один сайт может подходить сразу для нескольких целей, существуют также сложные индивидуальные проекты, которые выполняют каждую из перечисленных задач. Очень важно правильно определить, какой именно результат нужен и на основе этой информации формировать остальные этапы работы над созданием сайта.

3. Создание технического задания

Крайне важный шаг. Погрешности здесь недопустимы, так как они могут проявиться на любом этапе и в итоге отбросить проект далеко назад. В ТЗ прописываются основные требования проекта, его возможности и способ работы, учитывая анализ ЦА и конкурентов, цели и стратегию. По сути, два предыдущих этапа проводят ради того, чтобы составить качественное ТЗ и передать в работу специалистам.

Основные требования к техзаданию:

Детальность. Все, что должен сделать специалист, необходимо указать в ТЗ. Даже если это обязательная часть работы, даже если разработчик делал это сотни раз и не нуждается в напоминании. Каждое требование, каждый аспект работ следует прописывать.

Отсутствие неточностей. Любым неоднозначным формулировкам или субъективизму не место в техзадании. Это инструкция, в которой говорится, что и как сделать, а понятия: «хорошо», «красиво», «цепляет» у всех разные.

Понятный язык. Поправка: понятный для специалиста. Это значит, что специалист не должен расшифровывать и переспрашивать, что именно от него хотят.

4. Прототипирование

Для наглядности создается прототип с подробным расположением элементов и навигацией. Последняя требует особенного внимания. Слишком много сайтов создаются без учета удобства навигации. Почему так много людей любят продукты Apple? Потому что они интуитивно понятны. Пользователь при первом же контакте с продуктом понимает, как им пользоваться.

Дизайнер в первую очередь обеспечивает удобство интерфейса. Все элементы на странице должны быть расположены таким образом, чтобы пользователь без труда мог найти и воспользоваться ими. При помощи расположения информационных блоков дизайнер управляет вниманием посетителя, демонстрирует ему то, что он должен или хочет увидеть в данный момент.[5]

5. Создание макетов дизайна

Этап разработки дизайна сайта отвечает за визуальную составляющую ресурса:

- цвет;
- форма;
- шрифт;
- анимация.

Дизайнер придерживается фирменного стиля заказчика, вписывает логотип, делает сайт полностью узнаваемым и соответствующим компании.

Следовать трендам хорошо, но не делайте дизайн слишком “модным”, он должен быть современным и сохранять свою актуальность хотя бы два года. Позже, когда ресурс заслужит доверие поисковых ботов, посетителей и клиентов, вы сможете изменить его внешний вид.

6. Верстка

После завершения работы дизайнер отправляет макет на утверждение заказчику и только потом передает на адаптивную верстку. Она подразумевает перевод эскизов в HTML, т. е. адаптацию изображения сайта под любой браузер и устройство (ПК, планшет, смартфон и прочие гаджеты).

Это как раз тот случай, когда «семь раз отмерь, один отрежь». Лучше 7 раз прорисовывать дизайн сайта и один – его сверстать в HTML, нежели отправить на верстку первый же вариант, а потом бесконечно вносить правки и переверстывать. И тут должен быть внимательным прежде всего заказчик. Именно он утверждает концепцию сайта. Чем активнее заказчик участвует в обсуждении дизайна, тем вероятнее он получит ресурс, соответствующий его ожиданиям.

7. Программирование

Зачем нужно программирование? Вроде ж уже дизайнеры всё прорисовали, верстальщики перевели эскизы в HTML – адаптировали макет сайта под Всемирную сеть. Что дальше-то? А дальше на сцену выходит программист, который оживляет картинки. Тут лучше объяснить на примере кнопки призыва. Сделал её дизайнер прямоугольной и зеленого цвета, поместил в неё зазывающий лозунг (допустим, «Забронировать билеты сейчас»). Все это хорошо. Но ведь нужно сделать кнопку активной! Чтобы посетитель сразу попадал на заветную страницу бронирования билета, а не бился в истерику, видя, что от кликания мышки чуда не происходит.

В общем, программист – это тот, кто все сделанные дизайнером и собранные верстальщиком атрибуты сайта прописывает на определенном языке программирования (PHP, Java, JavaScript, Python или технологии ASP.NET).

Иногда этап верстки и программирования выполняют одновременно. Если, конечно, позволяют масштабы проекта.

8. Базовое наполнение

Сайт без информационного наполнения, как скворечник без птиц. Этап заполнения предполагает перевод контента в необходимый формат и размещение на страницах ресурса. Материалы готовятся заранее. Их могут писать как копирайтеры по заранее составленному техническому заданию (ТЗ), так и владелец сайта или любой человек, знающий специфику текстов и владеющий искусством слова. Но на этом работа не прекращается. Контент требует стабильного обновления. Иначе поисковые боты при ранжировании будут обходить сайт. Немаловажный фактор – заинтересованность клиентов. Их важно постоянно подогревать актуальной информацией.

9. Тестирование

Хоть после каждого этапа разработки ресурс проверяют на предмет «работает – не работает», без финального тестирования не обойтись. А вдруг где-то затерялась битая ссылка, некорректно отображается информация или всплывающее окно не выскакивает на экран? Недоработки во время тестирования обнаруживаются самые разные и команда их сразу устраняет. Как правило, для тестирования и демонстрации сайт размещают на специальной площадке. Позднее, когда же все погрешности устранены, проект со всеми файлами переносят на его постоянное место прописки в Интернете – хостинг. Но и тут ещё не всё. После переноса ресурс снова тестируют.

10. Техническая поддержка

При передаче готового проекта специалисты обучают работе с админпанелью, а также с сервисами сбора статистики. После этого владелец может полностью самостоятельно публиковать новый контент, создавать страницы, размещать товары, собирать данные для аналитики поведенческих показателей и пользоваться остальным функционалом ресурса. Если возникают какие-либо вопросы, компания-разработчик всегда готова помочь разобраться. Это же касается и багов, если вдруг они появились после релиза сайта - команда техподдержки их тут же устраним.

К основным этапам может добавляться еще раскрутка сайтов на этапе разработки, что поможет сайту быстрее выйти в топ и стабильно держать позиции, а также подготовит его к дальнейшему продвижению. [6]

1.3 Средства разработки Web – сайта

В нашем информационном обществе интернет занимает существенное место. С его помощью можно относительно быстро найти практически любую нужную нам информацию. Общество прогрессирует, поэтому прогрессируют и его запросы. Современному человеку необходимо постоянно получать информацию. Но, т.к. информация порой слишком разнообразна, то приходится часть из нее отсеивать и собирать только «нужную» информацию. Телевидение и радио давно не могут эту проблему решить. И тогда на помощь нам может прийти интернет, т.к. в нем можно сортировать информацию.

Веб -страницы - это фактически основа известного нам интернета. Поэтому вопрос о средствах разработки Web - страниц является одним из наиболее важным.[7]

Существует два способа технической разработки Web-страниц. Первый из них – это открыть какой-либо текстовый файл и набрать в нем HTML код. Разумеется , здесь же можно использовать CSS, Javascript,

DHTML, Active-X , PHP и другие. Таким способом пользовались до появления специального ПО.

Второй способ - программные средства, которые имеют специальные встроенные компоненты для конвертирования данных, созданных в этих продуктах, в HTML формат. Например, программы широко используемого пакета MS Office, начиная с версии 97, позволяет сохранить документ в формате HTML. При этом сохраняются, насколько это возможно, особенности форматирования символов и абзацев, изображения, таблицы, списки и так далее. Многие современные графические редакторы предоставляют возможность сохранить коллекцию картинок в виде HTML-документа. Надо отметить, что при таком конвертировании исходный текст HTML-документа получается крайне избыточным и нуждается в коррекции.

Третий способ – использовать некие программные средства для разработки Web-страниц. Этот способ разумеется легче, т.к. можно «видеть» то, что ты создаешь. Хотя, по-моему, для начинающего web-мастера было бы полезным сначала научиться писать код « по - старинке », т.е. познать HTML, DHTML, Active-X, javascript и CSS, SQL , PHP на собственной шкуре. Примером такого программного обеспечения может служить Adobe Dreamweaver.

Современные требования, предъявляемые к web-редакторам, включают в себя:

Поддержку каскадных таблиц стилей.

Использование современных скриптовых языков, таких как JavaScript и т.д.

Генерацию Dynamic HTML для различных браузеров.

Средства наглядного дизайна: вставка изображений, таблиц и фреймов.

Динамическое отображение создаваемой страницы в браузере.

Шаблоны WWW-страниц или специальные программы - "мастера" по их созданию.

Средства по управлению web-сайтом (создание файловой структуры сайта, отслеживание связей и проверка ссылок между страницами, публикация сайта в Интернет).

Существуют две разновидности web-редакторов: визуальные и не визуальные.

Не визуальные редакторы работают с "чистым" HTML-кодом. Они предоставляют возможность быстрого ввода тэгов HTML с использованием инструментальных панелей, вставки изображений и таблиц с помощью специальных диалоговых окон, работы с файловым менеджером для выбора папок и файлов, подсветки синтаксиса, проверки правильности получившегося кода и ссылок, использования справки по тегам и их атрибутам. Не визуальные web-редакторы часто называют HTML-редакторами. Они прекрасно подходят как для опытных web-дизайнеров, знающих HTML и прекрасно представляющих, как должна выглядеть web-страница в результате тех или иных изменений в ее коде, так и для тех, кто только начинает осваивать процесс создания сайта.[8]

Визуальные редакторы позволяют работать с самой web-страницей "как она есть", в режиме WYSIWYG (What You See Is What You Get - То, что ты видишь, то и получаешь). Редактирование и форматирование текстов, вставка рисунков, таблиц, гиперссылок и других элементов происходит как в обычном текстовом редакторе, а сама программа формирует (генерирует) соответствующий HTML-код. Кроме средств визуального редактирования эти web-редакторы одновременно предоставляют доступ к получившемуся HTML-коду. Визуальные редакторы — отличная стартовая площадка для начинающих web-

дизайнеров, плохо знакомых с HTML. Но и опытные дизайнеры часто пользуются визуальными редакторами, чтобы быстро "сверстать" web-страничку, а потом довести ее до ума вручную.

Выбор редактора, который будет использоваться для создания HTML-документов, зависит исключительно от понятия удобства и личных пристрастий каждого автора. На сегодняшний день разработано большое число web-редакторов, которые можно разбить на три группы.

Свободно распространяемые (бесплатные). В основном это не визуальные html-редакторы, разработанные небольшой группой авторов. Визуальные редакторы этой группы имеют ограниченный набор средств визуального редактирования. Характеризуются небольшим объемом занимаемой памяти, а некоторые из них даже не требуют специальной установки.

Условно-бесплатные. Web-редакторы этой группы характеризуются невысокой ценой.

Платные. В эту группу входят не визуальные и визуальные редакторы, разработанные крупными фирмами. Отвечают всем требованиям, предъявляемым к современным web-редакторам. Накладывают определенные ограничения на ресурсы компьютера. Обычно имеются демо-версии продукта, которые можно установить бесплатно и работать с ними ограниченное время.[9]

Вообще при разработке сайта можно выделить два основных этапа- это создание дизайна сайта и создание кода, которые упоминались ранее в этапах создания web-сайта.

1.4 Системный анализ предметной области

Разрабатываемый сайт предназначен для публикации научно-исследовательских работ преподавателей. Но для того чтобы правильно организовать работу базы данных, в которой у будут храниться файлы

данных а так, же организовать наполнение сайте всеми необходимыми инструментами, нужно провести системный анализ предметной области. Необходимо ознакомиться со всеми аспектами создания и типами научно-исследовательских работ.

Научная работа – это работа научного характера, связанная с научным поиском, проведением исследований, экспериментами в целях расширения имеющихся и получения новых знаний, проверки научных гипотез, установления закономерностей, научных обобщений и обоснований. В результате этой работы автор должен показать умение планировать свою деятельность, проявлять инициативу, придерживаться поставленного исследовательского вопроса, анализировать ход своей работы и оценивать полученные результаты, применять специализированную терминологию, отражать результаты своего исследования. [10]

Все научные статьи условно можно разделить на шесть основных типов, среди которых:

Научно-теоретические.

Обзорные.

Научно-практические.

Аналитические.

Научно-публицистические.

Научно-исследовательские.

Давайте же более детально познакомимся с особенностями каждого из представленных видов работ:

К научно-теоретическим, относятся работы, в которых автор сконцентрирован на детальном изучении конкретных проблем. Он изучает их на протяжении времени, систематизирует, пытается найти общие черты, выявляет закономерности и находит объяснение таким событиям и особенностям.

Обзорные статьи, это материалы, в которых автор высказывает собственное мнение по поводу работ собственных коллег по какой-то конкретной теме. Он может высказываться касательно определенных подходов и экспериментов, сравнивая подходы разных ученых.

Научно-практические работы являются одними из самых распространенных. В них автор описывает проведенные эксперименты, детально рассказывая обо всех нюансах и особенностях процессов, выделяя их пользу и смысл для раскрытия конкретной проблемы или даже направления в целом.

К аналитическим работам можно отнести глубокий и комплексный анализ какой-то конкретной узкой проблемы или задачи. Автор изучает массу материалов, собирает их воедино и с помощью экспериментальных методов выделяет определенные схожие черты и закономерности. Стоит отметить, что сами аналитические статьи могут быть оформлены только в научно-деловом стиле, а построение задающих тон предложений должно сначала выдвигать предположение «если» и затем раскрывать последствия «то».[11]

Научно-исследовательские статьи очень похожи на аналитические, однако их характеризуют последовательные, логически-цельные высказывания, которые имеют под собой весомые основания. В таких статьях не используется схема построения повествования «если» «то».

Научно-публицистические работы объединяют в себе исследование и публицистический стиль. В них может присутствовать эмоциональный

авторский слог, а в процессе написания автор должен быть нацелен на широкую, далеко не всегда научную аудиторию, а значит, и использовать простые и понятные структуры, обходясь без большого объема специализированных терминов.[12]

Написание научной работы делится на этапы:

1. Выбор направления исследования
2. Выбор темы исследования
3. Формулирование гипотезы
4. Планирование этапов работы
5. Сбор данных о предмете исследования
6. Проведение исследования
7. Оценка полученных результатов
8. Оформление работы

1.5 Проектирование баз данных

Проектирование баз данных — процесс создания схемы базы данных и определения необходимых ограничений целостности. Основные этапы проектирования баз данных

Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель

является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

- описание информационных объектов или понятий предметной области и связей между ними.
- описание ограничений целостности, то есть требований к допустимым значениям данных и к связям между ними.[13]

Логическое (дatalogическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов

управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.[14]

Выводы к главе 1

В первой главе курсового проекта было дано общее описание и классификация веб-сайтов.

Кроме того, были рассмотрены основные этапы разработки сайта, а именно: анализ целевой аудитории и конкурентов, постановка целей, разработка технического задания, дизайн и прототипирование, разработка дизайн-макетов, верстка и интеграция с CMS, верстка основных модулей, интеграция с учетными системами и сервисами, наполнение SEO-контентом, тестирование и запуск, поддержка и сопровождение.

Также был проведен анализ предметной области. В конце главы был выполнен дизайн базы данных проекта.

ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «СИСТЕМА ПУБЛИКАЦИИ НАУЧНЫХ РАБОТ И ДОСТИЖЕНИЙ ПРЕПОДАВАТЕЛЕЙ»

2.1. Реализация программного продукта

Для реализации веб-службы необходимо установить среду разработки Visual Studio Code. Следующим шагом будет установка базовой библиотеки Flask для языка программирования Python. На рис. 2.1 показана базовая структура проекта.

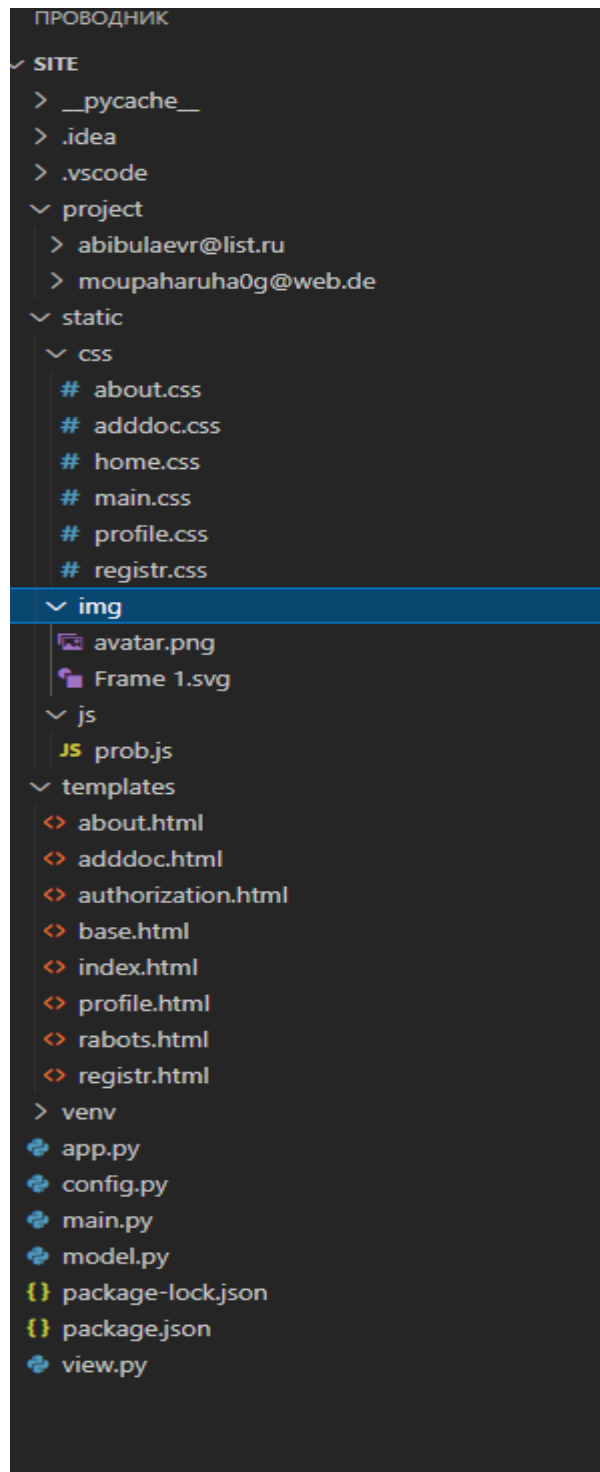


Рис. 2.1. Структура проекта

Данная структура показывает, что проект состоит из скриптов на языке программирования Python, на которых написана задняя часть проекта. Кроме того, проект содержит HTML-файлы, стили CSS и JS-скрипты на стороне клиента. Рассмотрим каждый из серверных скриптов:

app.py – основной файл проекта в нём происходит инициализация всех основных переменных.

config.py – хранит настройки сервера

main.py – служит для запуска проекта

model.py – здесь лежат классы выступающие в качестве таблиц для базы данных

view.py – отвечает за маршрутизацию по страницам проекта, здесь же хранятся все функции, например: регистрация, авторизация, выгрузка данных в базу данных и т. д.

В листинге 2.1 файла view.py вызывается функция index(). Данная функция предназначена в первую очередь для отображения файла “base.html” – главной страницы проекта. Также внутри этой функции реализованы методы для авторизации и регистрации пользователей.

Листинг 2.1

Маршрутизация

```
@app.route('/')  
  
def index():  
  
    return render_template("base.html", current_user=current_user)
```

В листинге 2.2 реализован метод удаления сеанса входа пользователя.

Листинг 2.2.

Функция очистки сессии и возврат на главную страницу.

```
@app.route("/logout")  
  
@login_required  
  
def logout():
```

```
logout_user()
```

```
return redirect("/")
```

В листинге 2.3 файла adddoc.html реализовано добавление научных работ.

Листинг 2.3.

Добавление научных работ.

```
<h1>Добавление научной работы</h1>
```

```
<div class="container">
```

```
<h1>Защита диссертаций</h1>
```

```
<h1><a class="zash" href="#des">Защита кандидатской  
диссертации</a></h1>
```

```
<div id="des" >
```

```
<form method="post" enctype="multipart/form-data">
```

```
<input class="inputnamedok" name="title_project" type="text"  
id="floatingInput" ><br>
```

```
<label class="opisinput"  
for="title_project">специальность(00.00.00);Тема("");Научный  
руководитель(должность ФИО);Город;Дата(00.00.00)</label><br>
```

```
<input type="file" name="file"><br>
```

```
<button type="submit" class="dobdocbtn">Загрузить </button>
```

```
</form>
```

```
</div>
```

```
<h1><a class="zas" href="#dess">Защита докторской  
диссертации</a></h1>
```

```

<div id="dess" >

    <input class="inputnamedok" name="kandid" type="text"><br>

    <label class="opisinput"
for="kandid">специальность(00.00.00);Тема("");Научный
руководитель(должность ФИО);Город;Дата(00.00.00)</label><br>

    <input type="file" accept=".pdf"><br>

    <button type="submit" class="dobdocbtn">Загрузить </button>

</div>

<h1> <a>Научные публикации</a></h1>

<h1><a class="zas" href="#desss">Научные публикации</a></h1>

<div id="desss" >

    <input class="inputnamedok" name="kandid" type="text"><br>

    <label class="opisinput" for="kandid">Название;Соавторы(должность
ФИО);Название издания и номер;Издательство;Город;Количество
страниц;Дата(00.00.00)</label><br>

    <input type="file" accept=".pdf"><br>

    <button type="submit" class="dobdocbtn">Загрузить </button>

</div>

</div>

```

На следующем этапе реализуется страница добавление документов пователями. В листинге 2.4 показан код отображающий форму добавления научных работ, а также сам функционал загрузки.

Листинг 2.4.

Файл view.py метод отправки заявок.

```

@app.route('/adddoc',methods=['POST',"GET"])

@login_required

def adddoc():

    if request.method=="POST":

        create_project(current_user.email,request.form['title_project'])

        file=request.files['file']

        filename=secure_filename(file.filename)

url='project/'+str(current_user.email)+'/'+request.form['title_project']

user_url=FileUrl(url=url,filename=filename,user_id=current_user.id)

        db.session.add(user_url)

        db.session.commit()

file.save(os.path.join('project/'+str(current_user.email)+'/'+request.form['title_p
roject'], filename))

        return render_template("adddoc.html",name=current_user.name)

```

2.2. База данных программного продукта

База данных разрабатываемого веб-приложения создавалась локально в СУБД MySQL с использованием графического веб-интерфейса phpMyAdmin (рисунок 2.2 «Создание базы данных»).

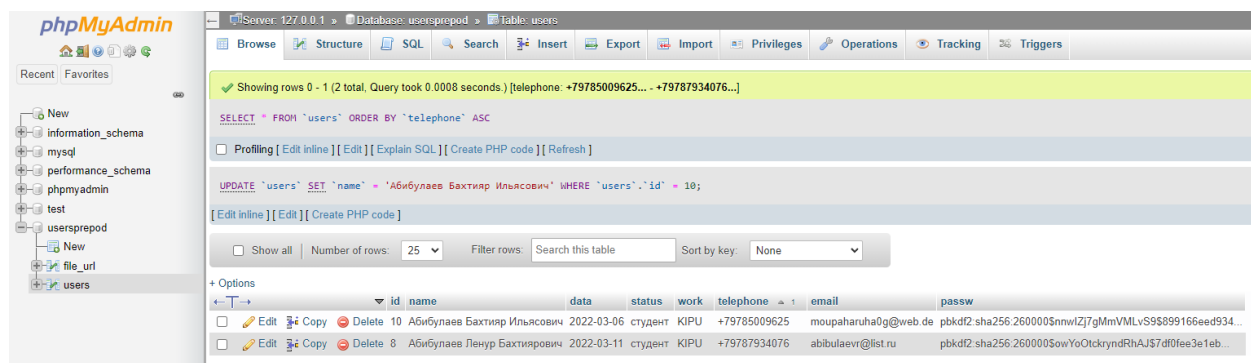


Рис. 2.2 База данных проекта

Далее необходимо задать наименование создаваемой базы данных, выбрать из предложенного списка кодировку и нажать кнопку «Создать».

В разработанном проекте в файле app.py реализована инициализация базы данных (рис. 2.3).

```
from flask import Flask
from config import Configuration
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager
import os

app = Flask(__name__)
app.config.from_object(Configuration)

db = SQLAlchemy(app)

login_manager=LoginManager(app)

def create_project(name, project_title):
    os.makedirs('project/'+name+'/'+project_title)
```

Рис. 2.3. Файл app.py

Для полноценной реализации проекта необходимо было создать следующие таблицы в базе данных (рис 2.4):

- users
- file_url

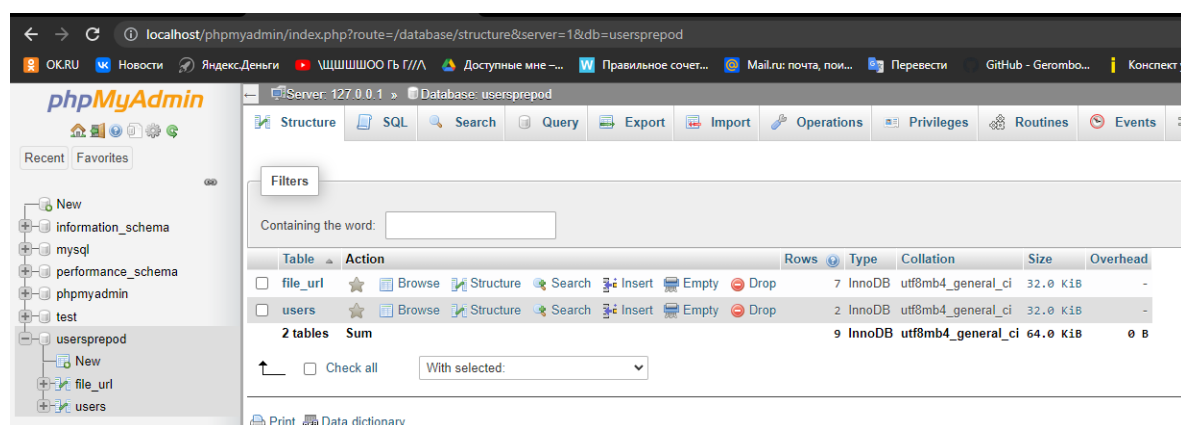


Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> file_url		7	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> users		2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
2 tables Sum		9	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Рис. 2.4. Таблицы базы данных.

На рисунке 2.5 представлена таблица, хранящая в себе данные о пользователях.

+ Options

id

name

data

status

work

telephone

email

passw

Edit

Copy

Delete

10

Абибулаев Бахтияр Ильясович

2022-03-06

студент

KIPU

+79785009625

moupaharuha0g@web.de

pbkdf2.sha256:260000\$nnwIZj7gMmVMLvS9\$899166eed934...

Edit

Copy

Delete

8

Абибулаев Ленур Бахтиярович

2022-03-11

студент

KIPU

+79787934076

abibulaevr@list.ru

pbkdf2.sha256:260000\$owYoOtcryndRhAJ\$7df0fee3e1eb...

Check all

With selected:

Edit

Copy

Delete

Export

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Query results operations

Рис. 2.5. Таблица users.

На рисунке 2.6 представлена таблица, хранящая в себе данные об добавленных научных работах.

☐ Show all
 |
 Number of rows:

25

 Filter rows:

Search this table

 Sort by key:

None

+ Options

				id	url	filename	user_id
<input type="checkbox"/>				1	project/abibulaevr@list.ru/ahsdgahsdgasfffffgfd	Zvonaryova_D_A_1.docx	8
<input type="checkbox"/>				2	project/abibulaevr@list.ru/ddd	ffrr.pdf	8
<input type="checkbox"/>				3	project/moupaharuha0g@web.de/09,03,03:jdfskodfs;Ju...	1.pdf	10
<input type="checkbox"/>				4	project/abibulaevr@list.ru/лдьмдлътъмдчл	Zvonaryova_D_A_1_7.docx	8
<input type="checkbox"/>				5	project/abibulaevr@list.ru/чтсмлчтямжлдчятимтсчди...	Zvonaryova_D_A_1_7.docx	8
<input type="checkbox"/>				6	project/abibulaevr@list.ru/09,03,03:иваомлмолчтсмл...	Zvonaryova_D_A_1_5.docx	8
<input type="checkbox"/>				7	project/abibulaevr@list.ru/счясясчяч;авыощдвым;	Zvonaryova_D_A_1_4.docx	8

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export

Рис. 2.6. Таблица file_url.

Для реализации всех таблиц базы данных создан отдельный скрипт model.py, который нужен не только для выполнения самих таблиц. Этот файл использовался для выполнения всей работы с базой данных: от создания таблиц до запросов к самой базе данных (листинг 2.5).

Листинг 2.5.

Файл model.py.

```
from cProfile import label

from turtle import back

from app import db

from flask_login import UserMixin
```

```

class Users(db.Model, UserMixin):

    id = db.Column(db.Integer, primary_key=True)

    name = db.Column(db.String(200), nullable=False)

    data = db.Column(db.String(200), nullable=False)

    status = db.Column(db.String(200), nullable=False)

    work = db.Column(db.String(200), nullable=False)

    telephone = db.Column(db.String(200), nullable=False)

    email = db.Column(db.String(120), nullable=False, unique=True)

    passw = db.Column(db.String(250), nullable=False)

    url=db.relationship("FileUrl")

    def __init__(self, *args, **kwargs):

        super(Users, self).__init__(*args, **kwargs)

    def __repr__(self):

        return '<Users { }>'.format(self.name)

```

```

class FileUrl(db.Model):

    id = db.Column(db.Integer, primary_key=True)

    url= db.Column(db.String(200), nullable=False)

    filname=db.Column(db.String(300), nullable=False)

    user_id = db.Column(db.Integer, db.ForeignKey('users.id'))

    def __init__(self, *args, **kwargs):

        super(FileUrl, self).__init__(*args, **kwargs)

    def __repr__(self):

        return '<FileUrl { }>'.format(self.url)

```

2.3. Описание интерфейса приложения

В ходе разработки основной части веб-приложения был спроектирован веб-сервис, обладающий базой данных для хранения информации о преподавателях, и загруженных ими научных работах.

При посещении сервиса пользователя встречает главная страница, где он может ознакомиться с сферой деятельности веб-приложения (рис 2.7).

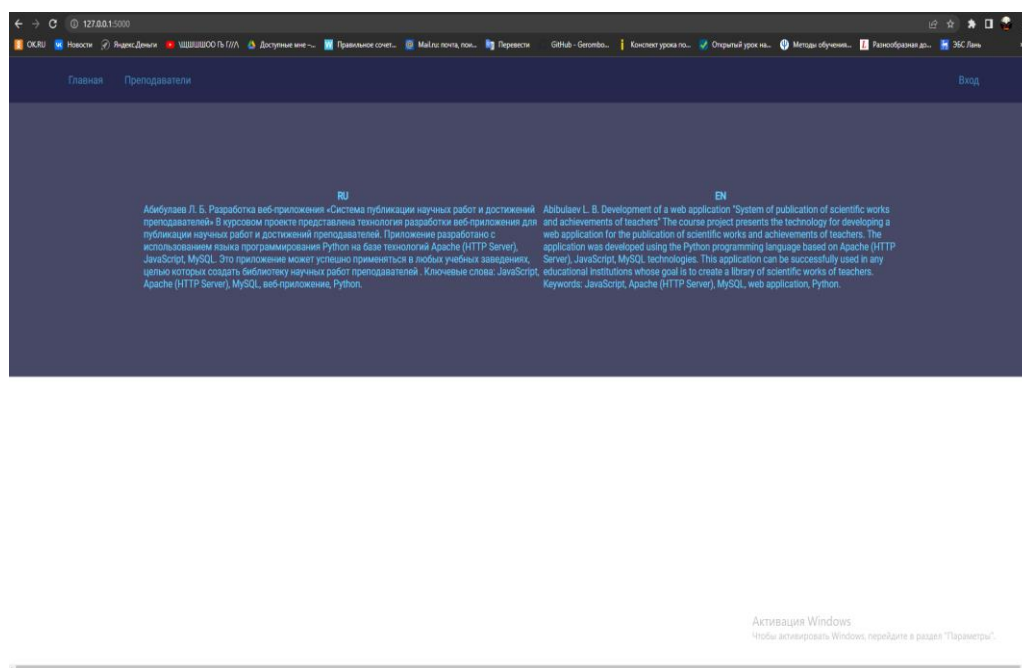


Рис. 2.7. Главная страница проекта.

Далее при желании пользователь может оставить заявку на ремонт своей техники, для этого ему нужно пройти регистрацию (рис 2.8) и авторизацию и (рис 2.9).

The screenshot shows a web browser window with a registration form titled "Регистрация". The form contains the following fields: "ФИО" (Full Name), "Дата рождения" (Date of Birth) with a calendar icon, "Должность" (Position), "Место работы" (Place of Work), "Номер телефона" (Phone Number), "Email", and "Пароль" (Password). Below the fields is a blue button labeled "Зарегистрироваться" (Register). At the bottom of the form, there is a link "У вас есть аккаунт? Log in" and a Windows activation watermark.

Рис. 2.8 Форма регистрации пользователей.

The screenshot shows a web browser window with a login form titled "Авторизация". The form contains two fields: "Email" and "Пароль" (Password). Below the fields are two blue buttons: "Log in" and "Sing in".

Рис. 2.9. Форма авторизации пользователей.

На странице «Добавить работу» пользователю необходимо заполнить все поля из формы (рис. 2.10), после данные о научной работе отправятся в базу данных, а прикрепленный документ сохранится в локальной базе данных.

Рис. 2.10. Форма для добавления научной работы

Данная функция доступна только для авторизованных пользователей так, же как и личный кабинет, в котором пользователь может посмотреть загруженные им документы научных работ(рис.2.11)

Рис. 2.11. Личный кабинет.

На странице «Преподаватели»(рис.2.12), можно увидеть таблицу состоящую из всех зарегистрированных пользователей. Нажав на ФИО любого из предложенных преподавателей можно увидеть все загруженные им научные работы(рис.2.13).

ФИО	Место работы	Должность	Номер телефона	Почта
Абибулаев Ленур Бахтиярович	КИУ	студент	+79787934076	abibulaev@list.ru
Абибулаев Бахтияр Ильясович	КИУ	студент	+79785009625	moupaharuha0g@web.de

Рис. 2.12. Список преподавателей.

Главная Преподаватели Мой кабинет Добавить работу	
Наименование документа	Загрузить
Zvonaryova_D_A_1.docx	Скачать
frrr.pdf	Скачать
Zvonaryova_D_A_1_7.docx	Скачать
Zvonaryova_D_A_1_7.docx	Скачать
Zvonaryova_D_A_1_5.docx	Скачать
Zvonaryova_D_A_1_4.docx	Скачать
1.pdf	Скачать
1.pdf	Скачать
Kurovaya_rabota.docx	Скачать
1.pdf	Скачать

Рис. 2.13. Список научных работ.

Вывод к главе 2

Во второй главе описан процесс разработки веб-приложения для публикации научных работ и достижений преподавателей, состоящего из страницы загрузки научных работ, страниц для просмотра зарегистрированных пользователей и их работ. На главной странице отображается текст аннотации курсового проекта.

ЗАКЛЮЧЕНИЕ

Курсовой проект посвящён разработке веб-приложения «Система публикации научных работ и достижений преподавателей». Разработанное приложение обладает широким функционалом, встроенные инструменты позволяют пользователю посмотреть весь список научных работ нужного ему преподавателя. Зарегистрированный пользователь, может добавлять свои научные работы в библиотеку, скачивать их для изучения. Кроме того, сервис обладает понятным интерфейсом для обычного пользователя.

В результате проделанной работы необходимо сделать следующие выводы:

- выполнены проектирование приложения и выбор программных инструментов;

- разработана система публикации научных работ.

В ходе разработки базы, были задействованы все основные средства управления и отображения информации. В дальнейшем планируются улучшение и доработки разработанного приложения. Все поставленные цели были выполнены в полном объеме.

В ходе разработки курсового проекта на тему «Разработка веб-приложения «Система публикации научных работ и достижений преподавателей»» были созданы 4 экрана, а именно: «Главная страница», «Преподаватели», «Личный кабинет», «Добавление работы». Для данного сервиса создана база данных с помощью веб-интерфейса для администрирования «phpMyAdmin». Созданная база данных работает корректно и автоматизирует следующие ключевые моменты работы: загрузка документов, хранение данных о пользователях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. iimag.ru [Электронный ресурс]// iimag.ru - 2020. - Режим доступа:
2. <https://iimag.ru/opredelenie-veb-sajta-vidy-i-klassifikacii-veb-sajtov/>
3. seoklub.ru [Электронный ресурс]// seoklub.ru - 2020. - Режим доступа:
https://seoklub.ru/tipy_saitov.html
4. studfile.net [Электронный ресурс]// studfile.net - 2018. - Режим
доступа: <https://studfile.net/preview/5133326/page:4/>
5. www.site-protect.ru [Электронный ресурс]// www.site-protect.ru -
2018. – Режим доступа:<https://www.site-protect.ru/o-nas/vopros-otvet/sozдание-saitov/poriadok-sroki-razrabotki-saita>
6. redline.by [Электронный ресурс]// redline.by - 2018. - Режим доступа:
<https://redline.by/novosti/etapyi-razrabotki-sajta.html>
7. yasno.mobi [Электронный ресурс]// yasno.mobi - 2018. - Режим
доступа:<https://yasno.mobi/blog/etapy-sozdaniya-sayta-ot-proektirovaniya-do-razrabotki/>
8. sites.google.com [Электронный ресурс]// sites.google.com - 2018. -
Режим доступа:
<https://sites.google.com/site/informtehnologi7943/sajt/sredstva-razrabotki-veb-sajta>
9. studbooks.net [Электронный ресурс]// studbooks.net - 2018. - Режим
доступа:https://studbooks.net/1269107/informatika/vybor_sredstv_razrabotki_sayta
10. www.ntoint.ru [Электронный ресурс]// www.ntoint.ru - 2018. - Режим
доступа:<https://www.ntoint.ru/info/osnovnie-sredstva-sozdaniya-saitov.shtml>
11. integraciya.org [Электронный ресурс]// integraciya.org - 2018. - Режим
доступа: <http://integraciya.org/konkursy/rekomendatsii-po-podgotovke-nauchno-issledovatel'skoj-raboty.php>
12. big-time.pro [Электронный ресурс]// big-time.pro - 2018. - Режим
доступа: <https://big-time.pro/raznovidnosti-nauchnyh-statej-ih-tipy/>

- 13.spb.hse.ru [Электронный ресурс]// spb.hse.ru
<https://spb.hse.ru/humart/lang/arg/art>
- 14.habr.com [Электронный ресурс]// habr.com - 2018. - Режим доступа:
<https://habr.com/ru/post/514364/>
- 15.www.mstu.edu.ru [Электронный ресурс]// www.mstu.edu.ru-2018.
- 16.Режим доступа: <http://www.mstu.edu.ru/study/materials/zelenkov/>
- 17.ch_5_1.html