

## **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 оптимизированную систему управления учебным процессом, в современных реалиях, когда завести свой сайт и наделить его рядом необходимых функций проще простого, но нет необходимых знаний для этого, может воспользоваться услугами готовых сервисов. Существует огромное количество сайтов ориентированных на публикацию научных работ, но их масштабы слишком велики для частного использования отдельным учебным заведениям, что может помешать пользователю найти нужную ему научную публикацию. Чтобы решить эту проблему, можно создать сайт, на котором будут публиковаться достижения и научные работы преподавателей, а храниться публикации будут в частной базе данных того или иного учебного заведения. Эта система позволит создать библиотеку педагогических работ, которой смогут пользоваться как преподаватели, так и студенты, которым интересна тематика опубликованных их преподавателями научных работ. Опираясь на вышеуказанное можно с уверенностью сказать что создание системы публикации достижений и научных работ преподавателей имеет большую актуальность в современных реалиях.

Целью бакалаврской 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 бакалаврской работы** является процесс разработки веб-сайта для публикации научных работ и достижений преподавателей.

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

**Практическая значимость** полученных результатов заключается в разработке веб-приложения, предназначенного для использования в учебных заведениях. Предоставляемый функционал позволяет преподавателям публиковать свои научные работы

**Структура бакалаврской работы.** Курсовой проект состоит из введения, двух глав, выводов, заключения, списка использованных источников, приложения.

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 Веб-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 курсового проекта.

Общий объем курсового проекта составляет 45 страниц. Работа содержит 21 рисунок, 1 листинг. Список использованных источников, включает 26 наименований.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	1
ГЛАВА 1. ОСНОВНЫЕ ОСОБЕННОСТИ СОЗДАНИЯ СИСТЕМ ПУБЛИКАЦИИ ДОСТИЖЕНИЙ И НАУЧНЫХ РАБОТ .....	4
1.1 Особенности публикации научных работ .....	4
1.2. Сравнительный анализ систем публикации научных работ .....	5
Выводы к главе 1 .....	7
Глава 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИЛОЖЕНИЯ ПУБЛИКАЦИИ ДОСТИЖЕНИЙ И НАУЧНЫХ РАБОТ ПРЕПОДАВАТЕЛЕЙ.....	8
2.1. Проектирование с использованием диаграмм .....	8
2.2. Анализ и выбор стека технологий для разработки системы .....	11
2.3. Описание используемых инструментариев.....	19
Выводы к главе 2 .....	23
ГЛАВА 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПУБЛИКАЦИИ ДОСТИЖЕНИЙ И НАУЧНЫХ РАБОТ ПРЕПОДАВАТЕЛЕЙ. ....	24
3.1. Управление базой данных.....	24
3.2. Реализация клиентской части приложения .....	29
3.3. Реализация серверной части приложения .....	35
Выводы к главе 3 .....	36
ЗАКЛЮЧЕНИЕ .....	37
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	<b>Error! Bookmark not defined.</b>
Приложение А .....	<b>Error! Bookmark not defined.</b>

# **ГЛАВА 1. ОСНОВНЫЕ ОСОБЕННОСТИ СОЗДАНИЯ СИСТЕМ ПУБЛИКАЦИИ ДОСТИЖЕНИЙ И НАУЧНЫХ РАБОТ**

## **1.1 Особенности публикации научных работ**

Научные исследования бесплодны, пока они не будут распространены и оценены другими исследователями. Публикация теоретической работы или лабораторного исследования является основным способом определения научного приоритета. Ценность научной работы может быть выявлена сразу после публикации или оценена спустя десятилетия.

### **Виды научных работ.**

Существует 7 основных видов научных работ. К ним относятся:

1. Доклад.
2. Реферат.
3. Научная статья.
4. Депонированная статья.
5. Сборник научных трудов.
6. Учебник.
7. Монография.

В России для получения ученых степеней необходимо публиковать научные работы. В соответствии с государственными нормами к кандидатской диссертации прилагается 2-3 публикации. Для получения возможности защищать докторскую диссертацию нужно передать информацию о 10-15 научных работах. При защите диссертации полученные патенты на модели, изобретения, промышленные образцы приравниваются к научным публикациям. Диссертационные комиссии проверяют каждую публикацию отдельно, в зависимости от авторитетности печатного, электронного журнала, сборника научных статей. [1]

### **Требования к научным работам.**

Научная публикация также должна соответствовать условиям публикации. В зависимости от отрасли науки и вида публикации в работе должны быть отражены:

1. Изучение теоретической проблемы или практического вопроса;
2. формулировка нового подхода;
3. теоретическое или практическое подтверждение идеи, метода исследований, технологии;
4. выводы о практическом использовании исследования или теоретической актуальности. [2]

### **Ценность публикаций.**

Ученые, которые занимаются математикой, литературой или физикой, начинают свою работу еще во время обучения. У студентов нет проблем с написанием теоретической статьи или подведением практических результатов. Научная работа студента проходит внутреннее рецензирование у наставников, рассматривается на семинарах и конференциях. Утверждение научно-методическим советом вуза. [3] В случае успешного прохождения всех этих этапов, подтверждающих важность работы студентов и их ценности для общества, статья рекомендуется к публикации. Высокая квалификация ученых позволяет им оценивать статьи, диссертации, монографии как в официальных рецензиях на публикации и доклады по исследованиям. Высокую оценку за научный материал получают доказательства и развития научных взглядов известных ученых. Общие статьи Эйнштейна о работах М. Планка, Э-Бэкингема и П. Ланжевена могут служить примером такой оценки. Но не всякая научная работа может быть признана международной. По теории игр, математик Д. Нэш получил Нобелевской премии (1994 г.) за один абзацем статьи по теории игр.[4]

### **1.2. Сравнительный анализ систем публикации научных работ**

В статье необходимо проанализировать текущую ситуацию в области информационных систем, сравнить эти системы и выбрать наиболее подходящие для публикации научных работ.

Научная библиотека eLIBRAR.RU - крупнейшая в России электронная база данных научных публикаций, обладающая широкими возможностями для поиска и анализа научной информации. В библиотеку включен Российский

индекс научного цитирования (Ринц) - бесплатный общедоступный инструмент для измерения издательской деятельности ученых и организаций, созданный приказом Министерства образования и науки Российской Федерации. С помощью компании «Научной электронной библиотеки» была создана, а затем и поддерживается компания «eLIBRARY.ru», которая является разработчиком и владельцем сайта. [5]

Сервис eLIBRAR.RU был создан по инициативе Российского фонда фундаментальных исследований в 1999 году для предоставления российских ученых электронного доступа ко всем мировым научным журналам. Сотрудничая с русскоязычными изданиями, eLIBRAR.RU является ведущей электронной библиотекой научной периодики на русском языке в мире и сейчас является одной из ведущих электронных библиотек научных журналов мира на русском языке. [6]

Посетители портала ELIBRARIA.RU могут просмотреть рефераты и полные тексты более 38 миллионов научных публикаций, патентов и статей в 5600 российских научно-технических журналах. Число пользователей институциональных организаций (организаций) составляет более 2800. На данный момент в системе зарегистрировано 1,7 млн индивидуальных посетителей из 125 стран мира. Из библиотеки ежедневно выкладывают более 12 млн статей и просматривают более 90 миллионов рефератов.

В настоящее время более 4500 российских научных журналов доступны бесплатно. Доступ к другим публикациям можно оформить подпиской или заказать отдельные публикации. [7]

Киберленинка поддерживает распространение знаний в формате открытого доступа. Это означает, что вы можете получить доступ к научным публикациям онлайн и получать их права на основании лицензий Creative Commons Attribution (CC-BY) Это ключевая идея для создания открытой научной инфраструктуры в России. [8]

Научные статьи, опубликованные в научных журналах России и ближнего зарубежья. Библиотека дополняется научными статьями из

журналов России или зарубежных стран для публикации результатов диссертационных исследований на сайте ВАК РФ. Представленные в библиотеке научные тексты бесплатно публикуются на сайте библиотеки, а также доступны для поиска как с помощью популярных поисковых систем, так и через систему полнотекстового научного поиска с поддержкой русской морфологии на сайте библиотеки. У пользователей библиотек есть возможность смотреть на экран планшета, мобильного телефона и других современных мобильных устройств. Благодаря этой услуге, издатели могут получить доступ к DOI. В сервисе содержится около 2 750 000 научных статей. [9]

Киберленинка - электронная библиотека открытого доступа, основная задача которой - популяризация науки и научных исследований, общественный контроль качества публикаций в области междисциплинарных наук. Современный институт научного рецензирования, современный научный журнал для исследования проблем российской науке. Киберленинка базируется на парадигме открытой науки.

Анализ существующих сервисов позволил нам решить задачу внедрения системы публикации научных статей, которая позволяет делиться всеми научными публикациями и не может использоваться человеком с незнанием всех сложностей при написании научной статьи. [10]

### **Выводы к главе 1**

В первой главе выпускной квалификационной работы были рассмотрены основы системы публикации научных статей, определены ее основные виды научных публикаций, учтены требования к публикации. Затем был проведен анализ систем публикации научных статей на рынке и их сравнение. Определены необходимые требования и функции к системе публикации научных работ и достижений преподавателей.

## **Глава 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИЛОЖЕНИЯ ПУБЛИКАЦИИ ДОСТИЖЕНИЙ И НАУЧНЫХ РАБОТ ПРЕПОДАВАТЕЛЕЙ**

### **2.1. Проектирование с использованием диаграмм**

UML расшифровывается как Unified Modeling Language, а диаграммы UML используются для визуализации того, как работает система и как с ней взаимодействуют другие системы и пользователи. Термин «система» может означать веб-сайт, приложение, рабочий процесс и т. д.

UML является общим языком, открытым стандартом. Он использует графическую нотацию для создания абстрактной модели системы и называется моделью MSKSUP. Uml был разработан для определения, визуализации, проектирования или документирования программных систем. Не является язык программирования, но генерация кода возможна с помощью моделей UML. [11]

Для решения проблем в вычислительной технике, науке об алгоритмах и данных существуют различные парадигмы и модели. На сегодняшний день существует четыре категории таких моделей: императивные, функциональные, декларативные и объектно-ориентированные. На объектно-ориентированных языках представлены алгоритмы, определяющие объекты и их взаимодействия друг с другом. Существующие на данный момент объекты являются реальными и требуют определенных действий. В качестве объектов могут выступать предметы быта, мебель и даже люди. Объектно-ориентированные языки являются основными в области программирования, потому что они моделируют объекты в реальном мире. Объёмный формат UML включает в себя несколько разновидностей объектоориентированного дизайна - объектно-ориентированный дизайн, методы объекта для моделирования и программное обеспечение. В UML, сочетающем в себе преимущества всех трех подходов, пользователи могут использовать удобную и непротиворечивую методологию для схематизации и документирования различных аспектов моделирования программ. [12]



Язык нотаций UML используется для реализации следующих целей:

1. иерархического описывания сложных систем путём выделения моделей;
2. формализацию функциональных требований к системе с помощью аппарата вариантов использования;
3. детализацию требований к системе путем постройки UML диаграмм деятельности и сценариев;
4. выделения классов данных и построение концептуальной модели данных в виде диаграмм классов;
5. выделение классов, описывающих пользовательский интерфейс, и создание схемы навигации экранов;
6. описание процессов общности объектов при выполнении системных функций;
7. описание поведения объектов в виде диаграмм деятельности и состояний;
8. аналитический обзор программных компонентов и их взаимодействия через интерфейсы;
9. аналитический обзор физической архитектуры системы.

В результате анализа был выделен преподаватель – User. Ему доступны следующие функции системы (рис. 2.1.).

1. Создать либо изменить личные данные;
2. Посмотреть данные преподавателей:
  - Email, номер телефона;
  - Должность;
  - Количество работ;
  - Место работы(ВУЗ; факультет; кафедра);
3. Добавлять либо удалять научные публикации;
4. Добавлять либо изменять данные научных работ;

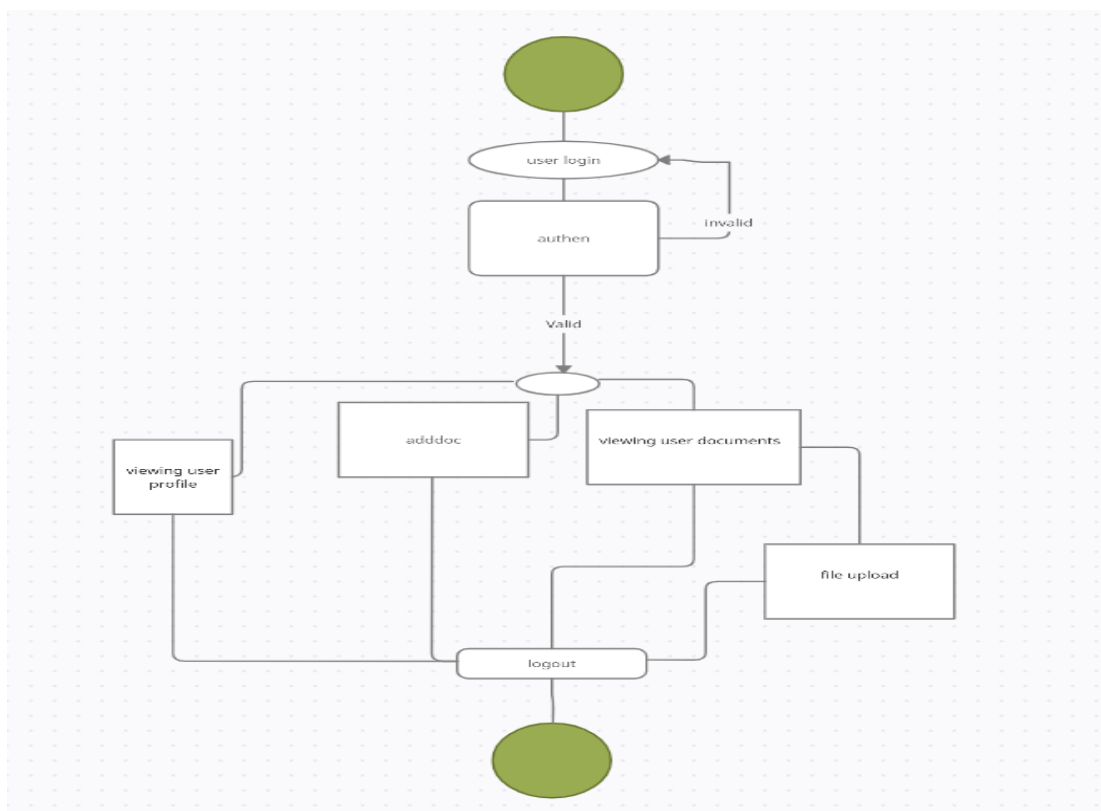


Рис.2.1 Диаграмма деятельности

На рис. 2.1 выделены основные функции преподавателя. Как видно из схемы, у преподавателя достаточно вариантов использования возможностей приложения.

На этапе разработки проекта составляется диаграмма классов предметной области, которая должна отражать важность отношений классов.

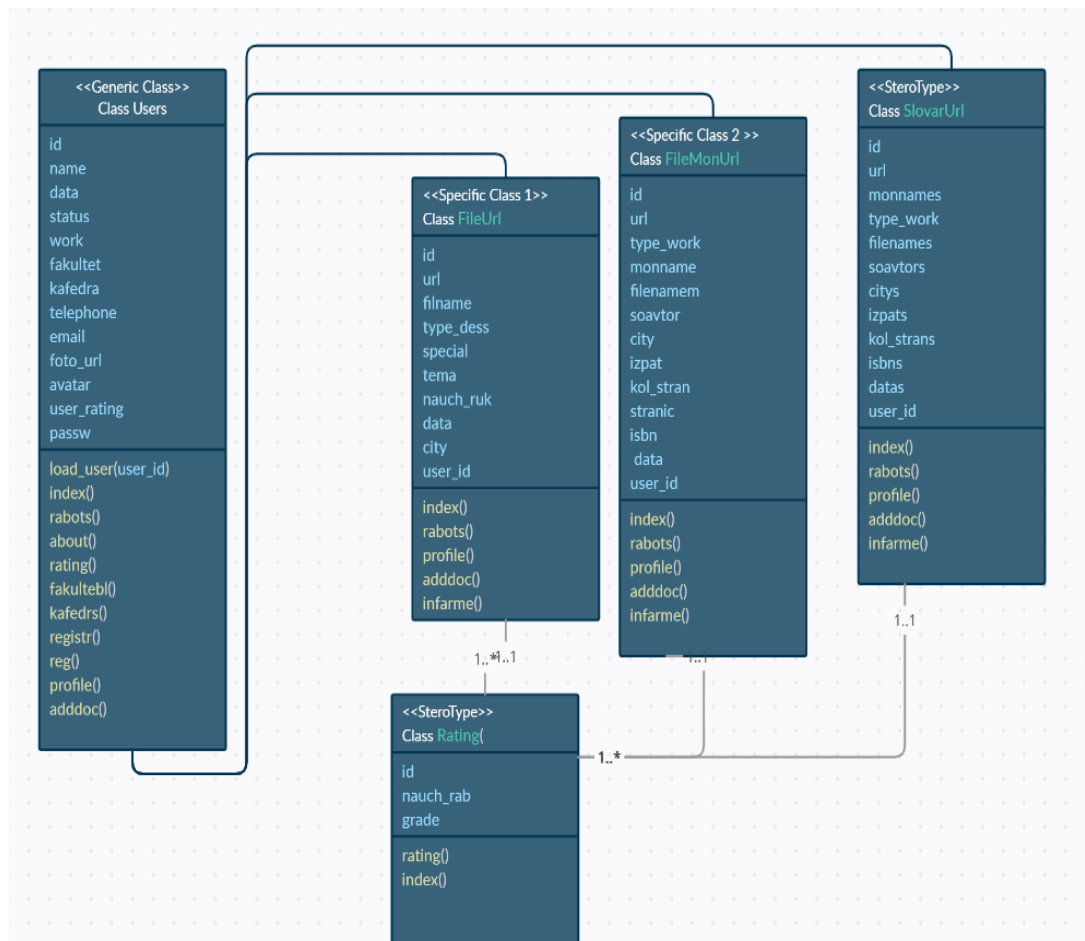


Рис. 2.2. Диаграмма классов

В данном разделе, была представлена большая часть классов и связь между ними.

## 2.2. Анализ и выбор стека технологий для разработки системы

Для начала работы необходимо выбрать язык программирования. UML является общим языком, открытым стандартом. Он использует графическую нотацию для создания абстрактной модели системы и называется моделью MSKSUP. Uml был разработан для определения, визуализации, проектирования или документирования программных систем. Не является язык программирования, но генерация кода возможна с помощью моделей UML. [13]

Для решения проблем в вычислительной технике, науке об алгоритмах и данных существуют различные парадигмы и модели. На сегодняшний день существует четыре категории таких моделей: императивные, функциональные, декларативные и объектно-ориентированные. На объектно-

ориентированных языках представлены алгоритмы, определяющие объекты и их взаимодействия друг с другом. Существующие на данный момент объекты являются реальными и требуют определенных действий. В качестве объектов могут выступать предметы быта, мебель и даже люди. Объектно-ориентированные языки являются основными в области программирования, потому что они моделируют объекты в реальном мире.

Объёмный формат UML включает в себя несколько разновидностей объектно-ориентированного дизайна - объектно-ориентированный дизайн, методы объекта для моделирования и программное обеспечение. В UML, сочетающем в себе преимущества всех трех подходов, пользователи могут использовать удобную и непротиворечивую методологию для схематизации и документирования различных аспектов моделирования программ. [14]

Проектирование приложений, программного обеспечения и приложений включает программное обеспечение для работы людей в течение дня. В частности, интернет-браузер, используемый пользователем для просмотра этого веб-сайта, считается программой. В данной статье представлен анализ наиболее популярных языков программирования для веб-разработки.

1. Высокий уровень (наиболее часто) / низкий уровень
2. Декларативный / императивный / процедурный
3. Универсальный / для предметной области
4. Объектно-ориентированный / параллельный
5. Командный / Скомпилированный / Язык сценария
6. Набор ответов

## **Java**

Проектирование приложений, программного обеспечения и приложений включает программное обеспечение для работы людей в течение дня. В частности, интернет-браузер, используемый пользователем для просмотра этого веб-сайта, считается программой. На этой странице представлен обзор самых распространенных языков программирования для веб-разработки. Java

- это объектно ориентированный язык разработки, созданной компанией Sun Microsystems. Дата выпуска — 23 мая 1995 года.

В байт-код, который затем проходит виртуальная машина Java (JVM), транслируется программа Java. Jvm -это программная система, которая обрабатывает байт-коды и инструктирует оборудование как интерпретатор. Отличительной особенностью этой реализации является отсутствие байт-кода от операционной системы и оборудования, что позволяет запускать Java приложения на любом устройстве, для которого есть JVM.[15]

Наивысшей особенностью технологии Java является гибкая система безопасности, которая полностью контролирует выполнение программ виртуальной машиной. Любые действия, превышающие разрешение программы (Например: попытка получить несанкционированный доступ к данным или подключение другого компьютера), будут немедленно завершены.

Иногда в качестве недостатка концепции виртуальной машины называют тот факт, что запуск байт-кода с помощью компьютерной программы может снизить производительность программ и алгоритмов на языке Java. Системные программы, написаны на Java имеют репутацию более медленных и занимают больше оперативной памяти по сравнению с программами C. Однако в сравнении со стандартными языками веб-разработки производительность Java обычно выше. [16]

## **RNR**

Основу этого языка составляют сценарии, написанные на языке RNR. Язык скриптов используется для написания сценариев и программ, автоматизирующих определенные задачи (в противном случае пользователь выполнял бы их вручную). Встроенные скрипты позволяют избежать ошибок, экономят время пользователя и меняют программу без опасения, что все остальное перестанет работать. Основываясь на большинстве языков, скриптовые языки не нуждаются в переводе и используются только для небольших рутинных задач. [17]

На данный момент PHP используется в основном для веб-разработки и используется в основном для backend-разработки, т.е. разработки серверной части страницы. Он часто используется для написания динамических страниц и небольших веб-программ. Вот несколько примеров его использования:

1. работа с базами данных;
2. создание системы комментариев;
3. разработка форумов;
4. разработка поиска по сайту;
5. создание страницы для регистрации и входа в систему.

## **Плюсы, минусы и особенности PHP**

### **Особенности**

Небольшая динамическая типизация PHP означает, что типы переменных определяются во время выполнения и различные типы могут использоваться вместе. Неявные преобразования выполняются автоматически.

Эта классификация может быть, как плюсов, так и минусов. В результате код проще, гибче и легче читается. С другой стороны, больше вероятность ошибок и вы их найдете после запуска программы.

### **Плюсы**

1. PHP бесплатный, а его код лежит в открытом доступе.
2. Для PHP создано множество библиотек, баз данных и фреймворков.
3. PHP совместим почти со всеми современными серверами.
4. PHP гибкий и даёт много свободы.
5. PHP хорош для новичков, ему легко научиться.
6. Простая и элегантная система классов и объектов.
7. Простой синтаксис.

### **Минусы**

1. Для работы с PHP нужно знать как минимум HTML, а лучше даже CSS. Также полезно знать JavaScript.

2. Сложный и гибкий язык PHP имеет обратную сторону: очень легко написать плохой код, легко сделать ошибку и сложно найти ошибку. Однако, это может затруднить обслуживание кода.

3. Безопасность. Во-первых, тот факт, что код находится в открытом доступе, создает проблему: если кто-то найдет ошибку в коде, он сможет использовать его в своих целях. Во-вторых, из-за простоты языка и большого количества разработчиков код зачастую некачественный и не обеспечивает защиты.

4. Скорость. PHP не самый производительный язык: например, его конкурент Javascript — быстрее.

5. В именах функций стандартной библиотеки нет четкой системы: у некоторых есть аббревиатуры, у некоторых нет (`call_user_func` vs. `create_function`). У некоторых есть символы подчеркивания, у некоторых нет (`isset` или `is_null`) и, например, в именах функций для работы со строками.

6. Иногда есть метка `str`, иногда ее нет. Такая нестабильность затрудняет работу. [18]

## **Python**

Сложный и гибкий язык PHP имеет обратную сторону: очень легко написать плохой код, легко сделать ошибку и сложно найти ошибку. Связано с тем, что это может затруднить обслуживание кода. За последние несколько лет популярность Python выросла в разы и даже превзошла Java. Разработчики машинного обучения, анализа данных и веб-приложений все чаще используют этот язык программирования, поскольку он имеет множество полезных библиотек, простой синтаксис и скорость. [19]

### **Преимущества разработки веб-приложений в Python:**

1. Удобство изучения: Python — самый популярный язык программирования для начинающих. У него есть общий язык, что позволяет писать значительно меньше кода по сравнению с Java или C++. В то же время у него более низкий входной барьер, так как он похож на обычный язык. Это

легко понять. И все же этот факт не означает того, что изучение языка Python в начале станет для вас легким.

2. Многочисленные библиотеки: Python предлагает широкий спектр библиотечных инструментов и пакетов услуг. Это позволит вам получить доступ к заранее написанному коду с ускоренным сроком разработки приложений. Для решения задач математического анализа вы можете использовать Numpy и Pandas для моделирования, Pygal на рисовании и SLALchemy - для вложенных запросов. Также у Python есть хорошие онлайн-фреймворки — Django и Flask. О них мы расскажем в следующем разделе.

3. Быстрая разработка. Программирование: Python занимает меньше времени, чем другие языки программирования и поэтому вы можете реализовать свои идеи намного быстрее! Более того, в процессе работы вы получите обратную связь и легко сможете повторить. Благодаря короткому времени разработки Python, он идеально подходит для стартапов. Поэтому они могут быть более быстрыми в выходе на рынок и получить преимущество.

4. Распространение: Python — один из самых распространенных языковых языков в мире, поэтому он часто пополняется новыми функциями и библиотеками. Python также обладает хорошей информацией и поддержкой сообщества для помощи новым разработчикам запускать свои проекты. [20]

Web-фреймворки представляют собой набор стандартизированных пакетов и модулей, которые предварительно написаны в соответствии с определенными правилами. Поэтому код позволяет разрабатывать веб-приложения, делая этот процесс проще и легче для разработчиков приложений, а программы более надежными. Этими словами, фреймы уже имеют встроенный компонент, облегчающий вам работу над проектом. Результаты анализа фреймворков:

### **Flask**

Это микрофреймворк, который представляет собой минималистичную веб-фреймворк. По сравнению с Django в нем нет таких функций, как механизм онлайн-шаблонов и авторизация учетной записи.



Приятный и простой в использовании Flask. Потому, что в этом случае вы можете добавлять нужные вам расширения и библиотеку при написании кода.

Альтернатива Flask заключается в предоставлении только тех компонентов, которые нужны для создания приложения. Некоторые функциональные функции включают встроенный сервер разработки, отправку запросов Restful и обработку данных HTTP. Flask - это популярный и мощный веб-фреймворк, который используют крупные компании, такие как Netflix, Linkedin и Uber. [21]

### **Django**

Django — это быстрое, безопасное и масштабируемое веб-приложение для Python. Оно предлагает высокоуровневый и открытый исходный код. В Django есть сильная поддержка сообщества и подробная документация.

Это комплексный пакет, который предоставит вам панель управления и интерфейс базы данных после создания приложения. Гарантия на использование продукта. Его содержание включает в себя большое количество функций, поэтому вы можете добавить отдельную библиотеку. Идентификация пользователей, механизм шаблонов и маршрутизации в схеме базы данных - все это примеры возможностей, предлагаемых фреймворком.

Адаптация Django завершена. Он позволяет работать как с минимально-реализуемыми продуктами, так и с достаточно развитыми компаниями. Django, Pinterest, Spotify и Instagram – все эти компании также используют Django.

Затем был проведен анализ и сравнение языка программирования. Было установлено, что язык и фреймворк следует выбирать в соответствии с потребностями и возможностями пользователя.

К работе над диссертацией был применен язык программирования Python из-за его простоты чтения и меньшего количества кода, необходимого для выполнения проекта. Для написания диссертации использовался фреймворк Flask за простоту использования. [22]

## **Angular**

Angular — это фреймворк для динамических веб-приложений, который позволяет вам использовать HTML в качестве языка шаблонов, а затем расширять синтаксис HTML для выражения компонентов приложения. С помощью привязки данных и внедрения зависимостей вы можете удалить большую часть кода, который должны писать.

### **Ключевые факты о Angular:**

Язык шаблона – Typescript, язык программирования компании Microsoft.

Даже если Typescript является основным языком для Angular, приложения также можно писать на таких языках: Dart или JavaScript.

Google поддерживает Angular.

Angular - это решение для создания одностраничных приложений (SPA), то есть решения для одностраничных приложений (SPA). Среди них есть известные социальные сети (Twitter, Instagram и Facebook), а также популярные приложения для социальных сетей (Twitch, YouTube и Facebook).

Angular предлагает платформу MVC на стороне клиента для создания и поддержки динамических приложений с современным уровнем качества.

Приложения, написанные на Angular можно использовать в нескольких браузерах. В Angular автоматически анализируется код JavaScript, который подходит для каждого браузера.

Хороший и четкий дизайн пользовательского интерфейса.

Простое маршрутирование.

Облачный фреймворк Angular позволяет легко расширять синтаксис HTML и создавать новые компоненты из директив.

Если говорить в целом, Angular является одним из самых популярных фреймворков для создания больших, высокопроизводительных и простых веб-приложений. [23]

Кроме того, Angular предоставляет рендеринг на стороне сервера для ускорения загрузки домашней страницы и улучшения SEO, облегчая поиск динамических страниц. Облачность быстрого рендеринга страниц

значительно расширит возможности веб-приложений следующего поколения, основанных на Angular. С помощью этого фреймворка вы также можете:

1. Легко тестировать код
2. Легко создавать персонализированные объектные модели документа (Document Object Model, DOM).
3. Моделировать данные ограничено для использования небольших моделей данных, что делает код простым и легким для тестирования.

В настоящее время Angular является одним из самых популярных клиентских фрейм. Angular отличается от других фреймворков и требует больше усилий для входа в него. [24]

### **2.3. Описание используемых инструментариев**

Чтобы написать веб-приложение, нужно выбрать конкретный технологический фонд из множества доступных. В качестве стека для разработки веб-сервисов были выбраны следующие технологии:

1. Python
2. Flask
3. MySQL
4. JavaScript

Python - высокоуровневый язык программирования общего назначения, который ориентирован на повышение продуктивности разработчиков, удобочитаемости и качества кода, а также обеспечение переносимости программ, написанных на нем. Язык полностью объектно-ориентирован в том смысле, что все является объектом. Необычной особенностью языка является назначение блоков кодом пробелами. Основной язык минималистский, поэтому на практике редко приходится обращаться к документации. Не известно, что сам язык является интерпретируемым и используется для написания скриптов. В некоторых случаях у программ на языке есть недостатки, которые могут быть связаны с низкой производительностью и большим потреблением памяти при написании кода на других языках, таких как C или C++.

Flask - это микрофреймворк, но слово «микро» не означает что веб-приложение с помощью Flask должно быть помещено в тот же файл кода Python. Хотя это и не запрещено законом. Не хватает функциональности в Flask. Нет, приставка «микро», как и в случае с приставкой «микро», означает, что Flask использует простое и расширяемое ядро. И это означает то, что Flask в этом случае не решает за программиста, что и как делать. Создавая веб-приложение в Flask программист сам решает, какие шаблоны использовать, к какой базе данных подключаться, использовать ли ORM или работать на чистом SL. Flask не содержит больших слоев абстрактной информации и поэтому работает быстро. По сути Flask – это все, что вам нужно, но не более.

Компания Flask поддерживает несколько расширений, которые включают в себя различные функции для интеграции с базой данных и проверки форм. Также возможно добавление различных технологий аутентификации. В бутылке множество параметров конфигурации, которые имеют разумные значения по умолчанию и некоторые предварительные настройки. Согласно соглашению Flask, шаблоны и статические файлы хранятся в подкаталоге исходного дерева Python с именами файлов или их описанием. В качестве механизма для создания алгоритмов используется модуль Jinja2. Для тех, кто только начинает использовать Flask, эти соглашения могут меняться и не должны меняться.

MYSQL – это система управления реляционными базами данных. Его исходный код (СУБДД) открыт. Для того чтобы сделать это, была использована клиент-серверная модель (2.3). С помощью набора программ или полного сервиса, используемого для работы с базами данных на основе реляционной модели, называется СУБД.

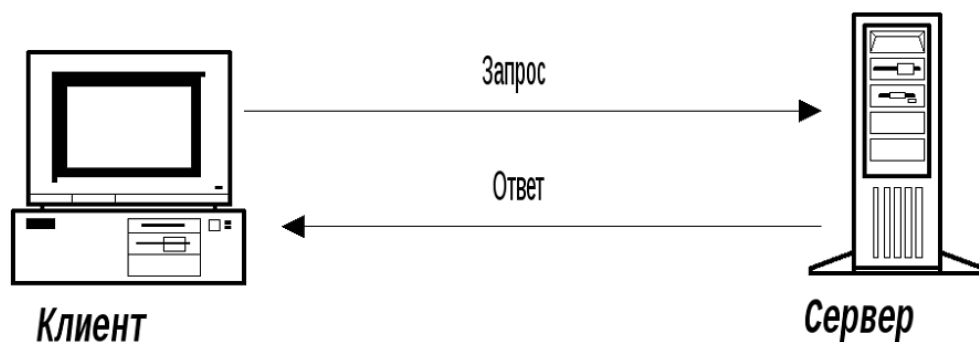


Рис.2.3. Модель «клиент-сервер»

База данных MySQL представляет собой реляционную базу данных. Это название базы данных, созданной в виде взаимосвязанных таблиц. С помощью открытого исходного кода можно изменить его. С помощью программы вы сможете самостоятельно установить её на свой компьютер. Если вы изучите бесплатный код, то сможете настроить его под себя. Для того чтобы получить расширенную поддержку, необходимо приобрести лицензионную версию.

В программе РСУБД компьютеры называют клиентами. Для того, чтобы получить данные, они подключаются к серверу. Сервер MySQL – многопоточный сервер базы данных. Это такая система, которая называется «клиент-сервер».

MySQL был разработан шведской компанией в 1984 году. Система управления была приобретена американской компанией Sun microsystems в 2008 году. В 2010 году его приобрел производитель Oracle.

JavaScript — мультипарадигменный язык программирования, обычно используемый в качестве инструмента для разработки доступа к различным объектам приложения. В рамках веб-разработки без знания этой технологии невозможно создавать современные интерактивные сайты. JS – это то, что оживляет веб сайты с макетом страницы (HTML), и пользовательским интерфейсом (CSS). Посредством этого языка реализуется возможность реакции страницы или ее отдельных элементов на действия посетителей. Сегодня JavaScript является основным языком программирования для всех

современных браузеров. Он полностью совместим с Windows, Livecd, Mac OS и всеми популярными мобильными платформами.[25]

Сейчас мы должны выяснить, как работает JavaScript в плане производительности. Вся деятельность в окне браузера создает события, а программирование на js помогает обрабатывать их определенным образом.

Стандартный алгоритм работы выглядит следующим образом:

1. Пользователь выполнил определенное действие.
2. Браузер определил событие.
3. Активируется JS-код.
4. На странице происходит заданное изменение.

Задача программиста — создать менеджеры для всех событий, на которые сайт должен реагировать при взаимодействии с пользователем. Для наглядности приведем два примера типичных сценариев.

1. Пользователь нажимает левую кнопку мыши.
2. Браузер фиксирует событие onclick.
3. Активируется функция changePhoto.
4. В окне просмотра фотографий изменяется изображение.

Если обработчик не внедрен в код, то алгоритм работы будет примерно следующим:

1. Пользователь совершает клик.
2. Браузер фиксирует событие onkeydown.
3. Для его обработки нет специального JS-кода.
4. После клика ничего не происходит.

Следует отметить, что программирование не всегда привязано к активным действиям пользователя. Например, код JavaScript может срабатывать при полной загрузке страницы или по истечении определенного времени, проведенного на сайте. Эти возможности активно используются для создания всплывающих окон и чатов. [26]

## **Выводы к главе 2**

Представлены здесь все классы, взаимоотношения с которыми были представлены в этом разделе. На этой диаграмме класс «Users» является основным. А все остальные классы наследуют от него методы, строки и методы. При этом некоторые классы здесь не участвуют. Они просто являются частью управления визуальной составляющей. Другие классы включены в список.

Разработано описание использования программного обеспечения и инструментов для разработки приложений, даны описания технологий создания приложения. После тщательного анализа был выбран наиболее подходящий вариант развития системы публикации достижений учителей и научных работ.

## **ГЛАВА 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПУБЛИКАЦИИ ДОСТИЖЕНИЙ И НАУЧНЫХ РАБОТ ПРЕПОДАВАТЕЛЕЙ.**

### **3.1. Управление базой данных**

В качестве базы данных для работы был выбран MySQL, простой и удобный программный продукт, который не требует от пользователя особых усилий. Однако MySQL также требует особого подхода при работе с ним. Требования к именам таблиц и столбцов такие же, как и к именам баз данных. Каждая строка имеет определенный тип данных, что ограничивает характер информации в столбце (например, чтобы предотвратить ввод букв в числовое поле). Mysql поддерживает множество видов информации: числовые данные (числовые), строковые, календарные и специальный тип NULL. В процессе разработки веб-приложений SQLAlchemy использует базу данных для работы с базой данных в серверной части. По сути, SQLAlchemy - это программное обеспечение с открытым исходным кодом для работы в базе данных с использованием языка программирования SL. По сути, он реализует программную технологию ORM (Object-Relational Mapping), которая связывает базы данных с концепциями объектно-ориентированных языков. SQLAlchemy предоставляет возможность описания структур базы данных, способов взаимодействия и связей с ними непосредственно в Python. Аппарат SQLAlchemy представлен как набор Python по лицензии MIT, что означает его можно использовать в проприетарных программных системах.

Для подключения базы данных к проекту используется файл app.py который представлен в листинге 3.1.

Листинг 3.1. Фрагмент файла app.py

```
from flask_sqlalchemy import SQLAlchemy  
db = SQLAlchemy(app)
```

После этого нужно в конфигурации указать ссылку на базу данных проекта – листинг 3.2.



## Фрагмент файла config.py

```
class Configuration(object):
    DEBUG = True
    SECRET_KEY = 'slim',
    SQLALCHEMY_TRACK_MODIFICATIONS = False,
    SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:@localhost/usersprepod'
```

Непосредственная работа с таблицами базы данных в SQLAlchemy осуществляется с помощью. Модель — это класс Python, который соответствует таблице в базе данных, а его свойства — это столбцы. Чтобы класс был допустимой моделью, должны быть выполнены следующие требования:

- Наследоваться от декларативного базового класса с помощью вызова функции `declarative_base()`.
- Объявить имя таблицы с помощью атрибута `__tablename__`.
- Объявить как минимум одну колонку, которая должна быть частью первичного ключа.

Создание таблиц в БД показан на картинках 3.1, 3.2, 3.3 файла `model.py`.

```
from cProfile import label
from fileinput import filename
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)
    fakultet=db.Column(db.String(200), nullable=False)
    kafedra=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)
    foto_url=db.Column(db.String(120), nullable=False )
    avatar=db.Column(db.String(120), nullable=False)
    user_rating=db.Column(db.Integer, nullable=False)
    passw = db.Column(db.String(250), nullable=False)
    url=db.relationship("FileUrl")
    urlm=db.relationship("FileMonUrl")
    url=db.relationship("SlovarUrl")
    def __init__(self, *args, **kwargs):
        super(Users, self).__init__(*args, **kwargs)
    def __repr__(self):
```

Рис.3.1. Листинг файла model.py

```

        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)
    filename = db.Column(db.String(300), nullable=False)
    type_dess = db.Column(db.String(300), nullable=False)
    special = db.Column(db.String(300), nullable=False)
    tema = db.Column(db.String(300), nullable=False)
    nachuk_ruk = db.Column(db.String(300), nullable=False)
    data = db.Column(db.String(300), nullable=False)
    city = db.Column(db.String(300), nullable=False)

```

Рис.3.2. Листинг файла model.py

```

    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)
class FileMonUrl(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    url = db.Column(db.String(200), nullable=False)
    type_work = db.Column(db.String(200), nullable=False)
    monname = db.Column(db.String(300), nullable=False)
    filename = db.Column(db.String(300), nullable=False)
    soavtor = db.Column(db.String(300), nullable=False)
    city = db.Column(db.String(300), nullable=False)
    izpat = db.Column(db.String(300), nullable=False)
    kol_stran = db.Column(db.String(300), nullable=False)
    stranic = db.Column(db.String(300), nullable=False)
    isbn = db.Column(db.String(300), nullable=False)
    data = db.Column(db.String(300), nullable=False)
    user_id = db.Column(db.Integer, db.ForeignKey('users.id'))
    def __init__(self, *args, **kwargs):
        super(FileMonUrl, self).__init__(*args, **kwargs)
    def __repr__(self):
        return '<FileMonUrl {}>'.format(self.url)
class SlovarUrl(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    url = db.Column(db.String(200), nullable=False)
    monnames = db.Column(db.String(300), nullable=False)
    type_work = db.Column(db.String(200), nullable=False)
    filenames = db.Column(db.String(300), nullable=False)
    soavtors = db.Column(db.String(300), nullable=False)

```

Рис.3.2. Листинг файла model.py

```

citys=db.Column(db.String(300), nullable=False)
izpats=db.Column(db.String(300), nullable=False)
kol_strans=db.Column(db.String(300), nullable=False)
isbns=db.Column(db.String(300), nullable=False)
datas=db.Column(db.String(300), nullable=False)
user_id = db.Column(db.Integer, db.ForeignKey('users.id'))

```

Рис.3.3. Листинг файла model.py

```

def __init__(self, *args, **kwargs):
    super(SlovarUrl, self).__init__(*args, **kwargs)
def __repr__(self):
    return '<SlovarUrl {}>'.format(self.url)
class Rating(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nauch_rab=db.Column(db.String(200), nullable=False)
    grade=db.Column(db.Integer, nullable=False)
    def __init__(self, *args, **kwargs):
        super(Rating, self).__init__(*args, **kwargs)
    def __repr__(self):
        return '<Rating {}>'.format(self.type lent)

```

Рис.3.4. Листинг файла model.py

После импорта содержимого model.py в базу данных создается несколько таблиц, необходимых для работы веб-приложения.

Таблица Users была создана для хранения данных преподавателем. В таблице Users хранятся такие данные, как: id пользователя

1. ФИО
2. Дата рождения
3. Должность
4. Место работы
5. Факультет
6. Кафедра
7. Номер телефона
8. Электронная почта
9. Рейтинг преподавателя

Для хранения данных о опубликованных диссертациях создана таблица, а именно FileUrl. В этой таблице хранятся такие данные как:

1. Id работы;
2. Адрес хранения работы в БД;
3. Тип работы;
4. Тема работы;
5. Наименование загруженного документа;
6. Специальность данной работы;
7. Дата публикации;
8. Город публикации;
9. ФИО научного руководителя;
10. Id пользователя добавившего работу.
11. Для хранения данных о опубликованных монографиях создана

таблица

FileMonUrl. В этой таблице хранятся такие данные как:

1. Id работы;
2. Адрес хранения работы в БД;
3. Тип работы;
4. Тема работы;
5. Наименование загруженного документа;
6. Издательство;
7. ISBN
8. Дата публикации;
9. Город публикации;
10. Количество страниц
11. Количество страниц написанные преподавателем публикующем

работу

12. ФИО соавтора;
13. Id пользователя добавившего работу.

Для хранения данных о опубликованных монографиях создана таблица

SlovarUrl. В этой таблице хранятся такие данные как:

1. Id работы;
2. Адрес хранения работы в БД;
3. Тип работы;
4. Тема работы;
5. Наименование загруженного документа;
6. Издательство;
7. ISBN
8. Дата публикации;
9. Город публикации;
10. Количество страниц
11. ФИО соавтора;
12. Id пользователя добавившего работу.

Таблицы FileUrl, FileMonUrl и SlovarUrl имеют связь с таблицей Users типа “один ко многим ”, так как, у одного пользователя может быть опубликовано большое количество работ.

Для реализации рейтинга преподавателей создана таблица Rating. В ней хранятся данные оценивания работ. Таблице созданы такие поля, как:

1. Id типа работы
2. Наименование научной работы
3. Балл оценивания работы

Существующие таблицы дают возможность реализовать весь необходимый функционал для работы приложения.

### **3.2. Реализация клиентской части приложения**

Для разработки пользовательского интерфейса приложения использовались HTML, CSS и JavaScript.

На основе конкурентного анализа был выявлен ряд требований, которым должен соответствовать сервер разрабатываемого веб-приложения:

- Информация о преподавателях
- Информация о публикациях

- Лента публикаций
- Рейтинг преподавателей
- Информация о факультетах и кафедрах

После перехода в приложение пользователь может просматривать работы, опубликованные учителями. На рис. 3.5 представлена реализация обзора опубликованных работ. Этот сайт неавторизованных пользователей позволяет просмотреть содержание произведений и их описание. Здесь можно увидеть общее количество публикаций по типам. Также можно посмотреть 3 лучших преподавателя по количеству баллов за опубликованные работы. Также может выбрать интересующие его типы работ во вкладках «Диссертации», «Монографии» и т.д.



Рис.3.5.Лента публикаций

Чтобы иметь возможность публиковать свои работы пользователь должен авторизоваться. Для этого нажимаем на кнопку вход и нас переносит на страницу авторизации (Рис.3.6).

**Авторизация**

Email

Пароль

Log in

Sign in

Рис.3.6. Авторизация

Если у него не зарегистрированного аккаунта необходимо нажать на кнопку «SingIn», после чего откроется окно регистрации (Рис.3.7). На данной странице необходимо заполнить все поля и загрузить личное фото.

Регистрация

ФИО

Дата рождения

Место работы

Факультет

Кафедра

Должность

Номер телефона

Email

Пароль

Фото профиля

У вас есть аккаунт? [Log In](#)

Рис.3.7.Регистрация

Если пользователь зарегистрирован и авторизован, то функционал главной страницы немного расширяется. На рисунке 3.8 продемонстрированы изменения главной страницы.

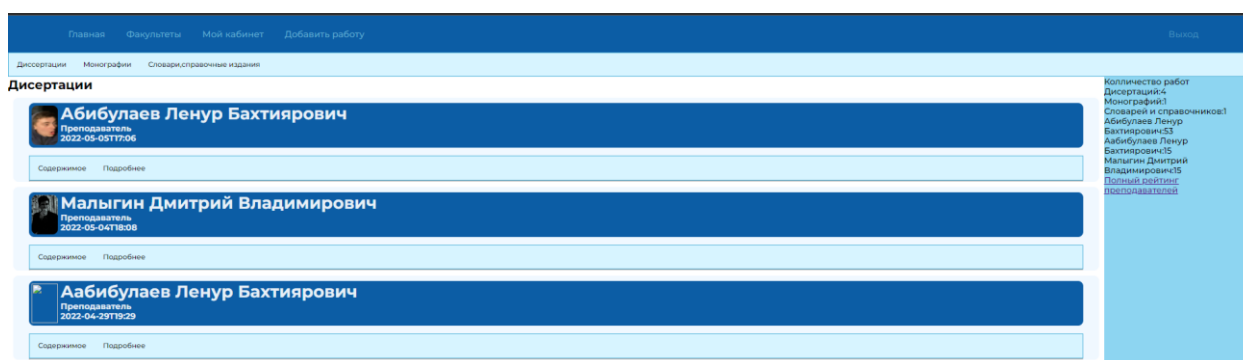


Рис.3.8.Главная страница авторизованного пользователя

После авторизации у нас появилась возможность перейти в личный кабинет и добавить работу.

Так же, меню навигации есть вкладка «Факультеты» нажав на которую нас переносит на страницу существующих факультетов. (Рис.3.9)

Главная	Факультеты	Мой кабинет	Добавить работу	Выход
Наименование факультета				
Факультет экономики, менеджмента и информационных технологий				
Факультет психологии и педагогического образования				
Факультет истории, искусств и крымскотатарского языка и литературы				
Филологический факультет				
Инженерно-технологический факультет				

Рис.3.9.Факультеты

После выбора необходимого факультета откроется страница с выбором кафедры которые есть на данном факультете. (Рис.3.10)

Главная	Факультеты	Мой кабинет	Добавить работу	Выход
Наименование кафедры				
Экономика				
Менеджмент				
Прикладная информатика				
Педагогическое образование				
Образование и педагогические науки				

Рис.3.10.Кафедры

Далее после выбора необходимой кафедры откроется страница с таблицей работающих на ней преподавателей. (Рис.3.11)

Главная	Факультеты	Мой кабинет	Добавить работу	Выход
ФИО	Место работы	Должность	Номер телефона	Почта
<a href="#">Абибулаев Ленур Бактирович</a>	ГБОУВО РК КИПУ имени Феэзи Якубова	Преподаватель	+79785009625	abibulaevn@list.ru
<a href="#">Малыгин Дмитрий Владимирович</a>	ГБОУВО РК КИПУ имени Феэзи Якубова	Преподаватель	+79787934076	abibulaevn@bk.ru
<a href="#">Абибулаев Ленур Бактирович</a>	ГБОУВО РК КИПУ имени Феэзи Якубова	Преподаватель	+79785009625	moupaharuha0g@web.de

Рис.3.11.Преподаватели

Если нажать ФИО какого-либо преподавателя откроется таблица с работами, которые загрузил преподаватель (Рис.3.12). Эти работы можно скачать или посмотреть.



<div>Главная</div> <div>Факультеты</div> <div>Мой кабинет</div> <div>Добавить работу</div> <div>Выход</div>								
<div>Диссертации</div> <div>Монографии</div> <div>Словари, справочные издания</div>								
Диссертации								
Тема	Тип диссертации	Специальность	Научный руководитель	Дата публикации	Город	Загрузить		
мобильное приложение	Докторская	Экономика	халилов	2022-05-05T17:06	симферополь	Скачать		
dsfsdaf	Докторская	Экономика	sdf	2022-05-04T14:06	asdf	Скачать		
Монографии								
Тема	Издательство	Количество страниц	Страницы написанные преподавателем	Соавтор	ISBN	Дата публикации	Город	Загрузить
мобильное приложение	диданкойская газета	15	2-3	Крылов	122-15-14-1886	2022-05-19T17:08	симферополь	Скачать
Словари и справочные издания								
Наименование	Издательство	Количество страниц	Соавтор	ISBN	Дата публикации	Город	Загрузить	

Рис.3.12.Научные работы

Так же у авторизованного пользователя есть возможность зайти в личный кабинет, для этого ему необходимо в меню навигации нажать на вкладку «Мой кабинет». На данной странице у пользователя отображаются его личные данные, а снизу под профилем пользователя есть таблица со всеми загруженными им работами (Рис.3.13).


<a href="#">Главная</a> <a href="#">Факультеты</a> <a href="#">Мой кабинет</a> <a href="#">Добавить работу</a> <a href="#">Выход</a>							
 <div> <b>Малыгин Дмитрий Владимирович</b>  <small>Дата рождения:</small>  1993-02-25  <small>Должность:</small>  Преподаватель  <small>Адрес:</small>  ГБОУ ВО РК КИТУ имени Феликса Якубова  <small>Телефон:</small>  +79787534076  <small>Почта:</small>  malygin@kita.ru </div>							
Наименование документа	Тип диссертации	Специальность	Научный руководитель	Дата публикации	Город	Загрузить	
мобильное приложение	Кандидатская	Педагогическое образование	Крылов	2022-05-04T18:08	симферополь	<a href="#">Скачать</a>	

Рис.3.13.Мой кабинет

Для публикации научной работы пользователю необходимо, в меню навигации нажать на вкладку «Добавить работу». Пользователю откроется страница с набором типов научных работ (Рис.3.14).

Главная Факультеты Мой кабинет Добавить работу Выход

Добавление научной работы

Научно-исследовательская деятельность

Защита диссертации

Монография

Словари, справочные издания

Научные публикации

Научные публикации

Рис.3.14.Добавление работы

Главная Факультеты Мой кабинет Добавить работу Выход

Добавление научной работы

Научно-исследовательская деятельность

Защита диссертации

Выбор

Специальность

Тема

Научный руководитель

Период

Дата

ДД.ММ.ГГГГ

Выбор файла | Файл не выбран

Выложить

Рис.3.15.Публикация диссертации

Научно-исследовательская деятельность

Защита диссертации

Монография

Название

Соавторы

Период

Издательство

Количество страниц

Страницы

ISBN

Дата

ДД.ММ.ГГГГ

Выбор файла | Файл не выбран

Выложить

Рис.3.16. Публикация монографии

Словари, справочные издания

Название

Соавторы

Город

Издательство

Количество страниц

ISBN

Дата

ДД.ММ.ГГГГ -->

Выберите файл | Файл не выбран

Загрузить

Рис.3.17. Публикация словарей, справочных изданий

Пользователь может добавить любую из предложенных научных работ. Для этого ему просто необходимо выбрать и нажать на нужный тип работы. (Рис. 3.15, Рис.3.16, Рис.3.17)

### 3.3. Реализация серверной части приложения

В данном веб - приложении – выделено 3 типа пользователя – это:

1. администратор
2. преподаватель
3. гость

Администратор самый главный и на нем лежит вся ответственность, даже регистрация на сайте проходит через него.

Преподавателю доступна информация о нем самом, а также возможность публиковать свои научные работы.

Гость же только может просматривать публикации, скачивать их, а так же, может ознакомиться с данными о преподавателях.

Для того чтобы попасть на сайт необходимо в браузере в адресной строке ввести адрес сайта «[www.adddoc.ru](http://www.adddoc.ru)».

Администрация базы данных происходит через сервис phpMyAdmin. Вся работа с данными перед выводом их в интерфейс происходит в файле «view.ru» который представлен в приложении А.

Структура проекта показана на рисунке 3.18

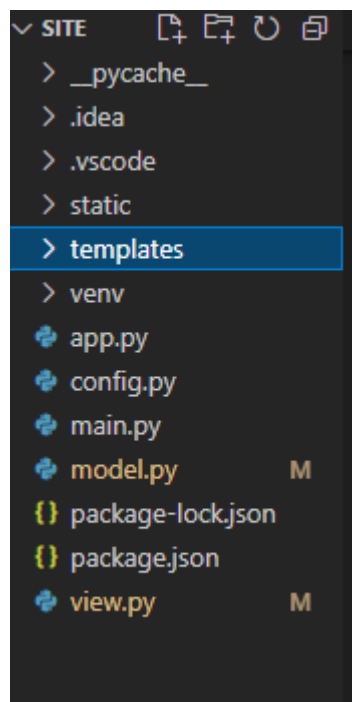


Рис.3.18. Структура проекта

### Выводы к главе 3

В процессе выполнения третьей главы дипломной работы были представлены разработки системы публикации научных работ и достижения преподавателей с использованием уже описанного технологического фонда.

Показана реализация базы данных и ее структура. Представлена реализация приложения с выбранным технологическим фондом. Следует отметить, что основная логика приложения была разработана и описана на примере основных методов, описывающих базовую логику приложения. Также была разработана база данных MySQL и создано веб-приложение «Система публикации научных работ и достижений преподавателей».

## **ЗАКЛЮЧЕНИЕ**

В выпускной квалификационной части выпускника были представлены разработки системы публикации научных работ и достижения преподавателей. Система основана на платформе Python. В дипломной работе были учтены существующие аналогичные приложения и их возможности.

В результате проделанной работы были сделаны следующие выводы:

1. выполнен обзор печатных и электронных литературных источников по заданной теме проанализированы требования к разрабатываемому программному приложению на основе обзора существующих систем публикаций научных работ;
2. выполнено проектирование приложения и были проведен выбор программного инструментария; определены методы и средства для достижения поставленной цели;
3. спроектировано, реализовано и протестировано веб-приложение для публикации научных работ и достижений преподавателей.

В первой главе были изучены концепции публикации научных работ, а также выявлены возможности данного направления. Проведено сравнение систем управления человеческими ресурсами и выявлены основные требования к этой системе.

Во второй главе был разработан план проекта, так называемая структура языка UML. Выполнен анализ и подбор технологического фонда, в качестве основы выбран язык программирования Python. Все технологии, используемые в проекте, были изучены и кратко описаны.

В процессе выполнения третьей главы была показана работа базы данных, доступ к базе данных в проекте осуществляется с помощью SQLAlchemy. Также была разработана клиентская часть приложения, разработанная с использованием JavaScript. И была разработана серверная часть приложения, показана его работоспособность, выведен список файлов веб-приложения.

### **Итоги бакалаврской работы**

В процессе выполнения диссертации автор рассмотрел существующие системы публикации научных статей, рассмотрел их обычные задачи и учел их основные направления на мировом рынке. Кроме того, программное обеспечение системы публикации научных статей спроектировано и реализовано с использованием объектно-ориентированного языка программирования Python, фреймворка Flask, базы данных MySQL и других современных технологий. Разрабатываемое веб-приложение представляет собой простую систему публикации научных статей, в которой планируются следующие изменения и улучшения.

### **Рекомендации по применению результатов работы**

Разработанное веб-приложение представляет собой систему публикации научных работ и реализовано таким образом, чтобы преподаватели могли делиться своими достижениями с другими, служили удобству пользователей, находили необходимую информацию и легко сохраняли работу преподавателей в базе данных системы. Также планируются следующие изменения и улучшения.

Данное приложение рассчитано на длительный период разработки. В ближайшее время планируется следующие изменения и новшества:

1. Добавить возможность комментирования работ другими преподавателями и гостями сайта.
2. Добавить некоторые функции для более удобного поиска нужной информации.
3. Также планируется совершить интеграцию с Google, чтобы подключить Google docs с помощью API для удобства преподавателей в работе со своими документами.
4. Дополнительная система оценивания, для того чтобы гости сайта могли оценить понравившуюся работу.
5. Улучшение интерфейса приложения.
6. Расширение базы данных для других университетов.

