

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных Технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ П.П. Дьячук
подпись инициалы, фамилия
« ____ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – «Информационные системы и технологии»

Разработка АИС фитнес-центра «Лето»

Руководитель	_____	<u>ст. препод. ИС</u>	<u>Ю. В. Шмагрис</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия
Консультант	_____	<u>доцент, к.т.н.</u>	<u>И. А. Легалов</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия
Выпускник	_____		<u>С.А. Юматова</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	<u>ст. препод. ИС</u>	<u>Ю.В. Шмагрис</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных Технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ П.П. Дьячук

подпись инициалы, фамилия

« _____ » _____ 2019 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работе

Студенту Юматовой Светлане Александровне

Группа: КИ15-13Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Разработка АИС фитнес центра «Лето»»

Утверждена приказом по университету № 7237/с от 24.05.2019 г.

Руководитель ВКР Ю.В. Шмагрис, старший преподаватель кафедры ИС, консультант ВКР: И.А. Легалов, доцент, к.т.н., кафедры ИС

Исходные данные для ВКР: Требования к разрабатываемой системе, рекомендации руководителя, учебные пособия.

Перечень разделов для ВКР: Введение, Аналитический обзор литературных источников, предметная область «Фитнес центр» и требования к ИС, практическая часть, сценарии работы с ИС «Фитнес центр», заключение, список использованных источников

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2016

Руководитель ВКР

подпись, дата

Ю.В. Шмагрис

инициалы и фамилия

Консультант

подпись, дата

И. А. Легалов

инициалы, фамилия

Задание принял к исполнению

подпись, дата

С.А. Юматова

инициалы и фамилия студента

« ____ » _____ 2019 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка АИС фитнес-центра «Лето»» содержит 47 страниц, 18 рисунков, 10 таблиц и 16 использованных источников.

Актуальность.

В наше время каждая компания, в особенности ориентированная на работу с клиентом, должна иметь сайт с удобным интерфейсом. Наличие веб-приложения с такими функциями как запись клиента на тренировку и управление базой данных на стороне администратора позволяет автоматизировать работу фитнес-центра и существенно сэкономить время, как клиента, так и администратора.

Целью бакалаврской работы является проектирование и разработка программных компонент веб-ориентированной информационной системы (ИС) компании фитнес центра.

Основные задачи:

- анализ предметной области;
- моделирование базы данных;
- разработка сайта.

СОДЕРЖАНИЕ

1	Аналитический обзор литературных источников	6
1.1	Основные понятия систем баз даны	6
1.2	База данных	8
1.3	Архитектура системы баз данных	8
1.4	Анализ систем управления базами данных (СУБД)	11
1.4.1	Oracle 12c	12
1.4.2	MySQL	13
1.4.3	Microsoft SQL сервер	15
1.4.4	PostgreSQL	16
1.4.5	ЛИНТЕР	17
1.4.6	РЕД База Данных	18
1.5	Анализ языков программирования для создания сайта	19
1.5.1	Python	20
1.5.2	SQL	21
1.5.3	JavaScript	22
1.5.4	PHP	23
2	Предметная область «Фитнес центр» и требования к ИС	26
3	Практическая часть	27
3.1	Даталогическое проектирование БД фитнес центра	27
3.2	Разработка сайта «Фитнес центр»	34
4	Сценарии работы с ИС «Фитнес центр»	38
	ЗАКЛЮЧЕНИЕ	45
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	46

ВВЕДЕНИЕ

В современном обществе наблюдается значительный рост информационных потребностей, внедрение информационного обеспечения во все сферы человеческой деятельности, а также в различных организациях. Ни одна организация как коммерческая, так и государственная не обходится без информационных технологий.

Для удобства управления современным предприятием используют различные информационные системы, которые предназначены для получения, хранения, поиска и обработки информации. Для хранения специальным образом организованной информации используются базы данных. Они позволяют структурировать, хранить и извлекать информацию оптимальным для пользователя способом.

Целью работы является проектирование и разработка программных компонент веб-ориентированной информационной системы (ИС) компании фитнес центра. Основные ее функциональные возможности связаны с планированием тренировок и отслеживанием их проведения. В системе реализуется функционал для администратора и клиента.

В соответствии с поставленной целью были сформулированы следующие задачи:

- изучить ИС фитнес центров для формирования списка ключевых функций;
- формирование информационно-логической модели для проектирования БД;
- создание БД в СУБД MySQL;
- определение модулей PHP и их функций как на стороне клиента, так и на сервере;
- реализация программных модулей и их отладка на различных тестовых наборах данных, как корректных, так и нет;

– проверка функционирования информационной системы в целом и внесение необходимых изменений.

Объектом бакалаврской работы являются актуальные веб-технологии, базирующиеся на использование сервера баз данных (MySQL) и web-сервера (Apache).

Предметом исследования являются средства разработки приложений - языки программирования HTML, CSS, PHP, SQL и JavaScript, а также системы управления базами данных и веб-приложений.

1 Аналитический обзор литературных источников

1.1 Основные понятия систем баз даны

Система баз данных – это компьютеризированная система хранения структурированных данных, основной целью которой является хранить информацию и предоставлять ее по требованию.

По сравнению с традиционным методом ведения учета у системы баз данных есть некоторые преимущества:

- компактность;
- скорость;
- актуальность.

Система баз данных включает в себя четыре основных компонента: данные, аппаратное обеспечение, программное обеспечение (в том числе систему управления базами данных (СУБД)) и пользователи.

А также системы баз данных бывают однопользовательские и многопользовательские. В однопользовательской системе к базе данных одновременно доступ может получить только один пользователь, а в многопользовательской системе к базе данных доступ могут получать сразу несколько пользователей [1].

В общем случае данные в базе данных являются интегрированными и разделяемыми. Понятие интегрированности данных подразумевает под собой возможность представить базу данных разными файлами данных, которые:

- могут быть объединены между собой, допустим, база данных реализованная в нескольких файлах данных, информация в которых дополняет одна другую.
- полностью или частично выключают избыточность информации.

В таком случае структура файлов построена таким образом, что не обязательно задавать дополнительную уточняющую информацию в одном файле, которая уже есть и представлена в другом файле.

А понятие разделяемости данных подразумевает под собой возможность использования отдельных элементов, которые хранятся в базе данных несколькими различными пользователями. То есть каждый пользователь имеет возможность воспользоваться одним и тем же элементом данных на свое усмотрение.

Аппаратное обеспечение системы включает в себя следующее:

- тома вторичной (внешней) памяти, которые используются для хранения информации, а также соответствующие устройства ввода-вывода, контроллеры устройств, каналы ввода-вывода и т.д.
- аппаратный процессор в сочетании с основной (первичной) памятью, предназначенные для поддержания работы программного обеспечения системы баз данных.

Между физической базой данных и пользователями системы располагается уровень программного обеспечения, который называется менеджер базы данных или система управления базы данных (СУБД). Основной задачей СУБД является предоставить пользователю возможность работать с базой данных, при этом не вникая в детали на уровне аппаратного обеспечения. А также в СУБД обрабатываются все запросы пользователя на доступ к базе данных.

Пользователей базы данных можно поделить на три большие взаимосвязанные группы:

- прикладные программисты. Их задача заключается в написании прикладных программ, использующих базу данных.
- конечные пользователи. Они работают с системой баз данных непосредственно через рабочую станцию или терминалы.
- администраторы базы данных. Их задача заключается в обеспечении необходимой технической поддержки базы данных. То есть они реализовывают создание базы данных, а также формируют технический контроль.

1.2 База данных

База данных – это упорядоченная совокупность данных, которая предназначена для хранения, изменения, накопления, а также обработки с помощью ЭВМ.

Сущность – это важный объект, данные которого должны быть сохранены в базу данных. Между сущностями существуют связи, которые объединяют основные сущности. В реляционных базах данных и основные сущности и связи между ними представляются с помощью таблиц. Связь можно понимать как связь особого типа.

Сущности (связи) имеют некоторые свойства, соответствующие тем данным о них, которые мы желаем записать. В общем случае свойства могут быть как простыми, так и сложными, причём настолько, насколько это потребуется [2].

Независимость может быть реализована на двух уровнях: физическом и логическом. Независимость баз данных может быть определена как иммунитет прикладных программ к изменениям способа хранения данных и используемых методов доступа. Среди прочего для независимости данных требуется строгое разделение между моделью данных и её реализацией.

Системы баз данных обычно поддерживают транзакции или логические единицы работы. Основное преимущество транзакций заключается в том, что они гарантируют атомарность выполняемых действий, несмотря на возможные сбои системы, имевшие место до завершения выполнения транзакции.

1.3 Архитектура системы баз данных

Архитектура системы баз данных включает три уровня: внутренний, внешний и концептуальный. Внутренний уровень (также называемый физическим) наиболее близок к физическому хранилищу информации, т.е.

связан со способами хранения информации на физических устройствах. Внешний уровень (также называемый пользовательским логическим) наиболее близок к пользователям, т.е. связан со способами представления данных с отдельными пользователями. Концептуальный уровень (также называемый общим логическим или просто логическим) является “промежуточным” уровнем между первым и вторым [3].

Если внешний уровень связан с индивидуальными представлениями пользователей, то концептуальный уровень связан с обобщённым представлением пользователей. Иначе говоря может существовать несколько внешних представлений, каждое из которых состоит из более или менее абстрактного представления определённой части базы данных, и только одно концептуальное представление, состоящее из абстрактного представления базы данных в целом.

Архитектура базы данных, кроме элементов самих трёх уровней включает определённые отображения. Отображение “концептуальный-внутренний” устанавливает соответствие между концептуальным представлением и хранимой базой данных, т.е. описывает как концептуальные записи и поля представлены на внутреннем уровне. При изменении структуры хранимой базы данных отображение “концептуальный-внутренний” также изменится, причём таким образом, чтобы концептуальная схема осталась неизменной. Отображение “внешний-концептуальный” определяет соответствие между некоторым внешним представлением и концептуальным представлением [4].

Пользователи, т.е. конечные пользователи и прикладные программисты, работающие на внешнем уровне, взаимодействуют с данными с помощью подязыка, который включает по крайней мере два компонента: язык определения данных (ЯОД) и язык манипулирования данными (ЯМД).

Система управления базой данных (СУБД) представляет собой программное обеспечение, которое управляет всем доступом к базе данных.

Функции СУБД:

- определение данных;
- обработка данных;
- оптимизация и выполнение;
- защита и сохранение целостности данных;
- восстановление данных и поддержка параллельности;
- словарь данных.

На высоком уровне систему баз данных можно рассматривать как систему с очень простой структурой, состоящей из двух частей – сервера (внутреннего компонента или машины баз данных) и набора клиентов (внешнего компонента или внешнего интерфейса).

Сервер – это сама СУБД. Он поддерживает все основные функции СУБД и предоставляет полную поддержку внешнего, концептуального и внутреннего уровня.

Клиенты – это различные приложения, которые выполняются поверх СУБД: как приложения написанные пользователями, так и встроенные приложения, предоставляемые поставщиком СУБД или некоторыми сторонними поставщиками программного обеспечения.

Распределённая обработка – это возможность соединить разные машины в коммуникационную сеть для организации совместного решения одной задачи обработки данных на нескольких машинах сети. Взаимодействие между различными машинами осуществляется с помощью специального программного обеспечения, предназначенного для управления сетью. В общем случае каждый сервер может обслуживать много клиентов, а каждый клиент может работать со многими серверами. Если система обеспечивает полную прозрачность доступа, то в таком случае мы имеем настоящую распределённую систему баз данных [5].

Каталог (словарь данных) – это набор системных переменных-отношений, содержащих метаданные о различных элементах, важных для

системы. Пользователи могут опрашивать каталог теми же методами, которые они применяют для доступа к собственным данным.

1.4 Анализ систем управления базами данных (СУБД)

На начальном этапе работы у меня возникла необходимость выбрать одну из множества СУБД, которые подходят для выполнения поставленных задач. Чтобы выбрать средства разработки я провела сравнительный анализ.

Несмотря на то, что все системы управления базами данных выполняют одну и ту же основную задачу (т.е. дают возможность пользователям создавать, редактировать и получать доступ к информации, хранящейся в базах данных), сам процесс выполнения этой задачи варьируется в широких пределах. Кроме того, функции и возможности каждой СУБД могут существенно отличаться. Различные СУБД документированы по-разному: более или менее тщательно. По-разному предоставляется и техническая поддержка.

При сравнении различных популярных баз данных, следует учитывать, удобна ли для пользователя и масштабируема ли данная конкретная СУБД, а также убедиться, что она будет хорошо интегрироваться с другими продуктами, которые уже используются. Кроме того, во время выбора следует принять во внимание стоимость системы и поддержки, предоставляемой разработчиком [6].

Если речь идёт о выборе СУБД для предприятия, то следует принять во внимание возможность СУБД «расти» вместе с развитием организации. Малому бизнесу могут потребоваться только базовые функции и возможности, а также небольшое количество информации, размещаемой в БД. Но требования могут существенно расти с течением времени, а переход на другую СУБД может стать проблемой.

Существует несколько популярных СУБД, как платных, так и бесплатных, которые можно рекомендовать для применения в организации.

1.4.1 Oracle 12c

Корпорация Oracle предлагает одноимённый продукт, с которого обычно начинается рассмотрение вариантов популярных СУБД. Первая версия Oracle была создана в конце 70-х годов, имея на данный момент блестящую репутацию. Кроме того, существует несколько версий этого продукта для удовлетворения потребностей конкретной организации.

Актуальная версия Oracle на данный момент - 12c, она предназначена для облачных сред и может быть размещена на одном или нескольких серверах, что позволяет управлять базами данных, которые содержат множество записей. Некоторые из функций новой версии Oracle включают в себя grid framework и использования как физических, так и логических структур [7].

В таком случае, физическое управление данными не влияет на доступ к логическим структурам. Более того, безопасность в этой версии доведена до высочайшего уровня, так как каждая транзакция изолирована от других.

Достоинства:

- самые свежие инновации и впечатляющий функционал уже внедрены в этом продукте, поскольку компания Oracle стремится держать планку даже на фоне других разработчиков СУБД;
- СУБД от Oracle является крайне надёжной, фактически это эталон надёжности среди подобных систем.

Недостатки:

- стоимость Oracle может оказаться непомерно высокой, особенно для небольших организаций;

- система может потребовать значительных ресурсов уже сразу после установки, поэтому возможно потребуется модернизировать оборудование для внедрения Oracle;
- идеально подходит для крупных организаций, которые работают с огромными базами данных и разнообразными функциями [8].

1.4.2 MySQL

MySQL - одна из самых популярных баз данных для веб-приложений. Фактически, является стандартом de facto для веб-серверов, которые работают под управлением операционной системы Linux. MySQL - это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала, который может стать и достоинством и недостатком - в зависимости от области внедрения.

Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Эта СУБД позволяет выбирать различные движки для системы хранения, которые позволяют менять функционал инструмента и выполнять обработку данных, хранящихся в различных типах таблиц. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие

полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. Она также имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных. Система невероятно надежна и не стремится подчинить себе все доступные аппаратные ресурсы [9].

Достоинства:

- Распространяется бесплатно;
- прекрасно документирована;
- предлагает много функций, даже в бесплатной версии;
- пакет MySQL включен в стандартные репозитории наиболее распространённых дистрибутивов операционной системы Linux, что позволяет устанавливать её элементарно;
- поддерживает набор пользовательских интерфейсов;
- может работать с другими базами данных, включая DB2 и Oracle.

Недостатки:

- придётся потратить много времени и усилий, чтобы заставить MySQL выполнять несложные задачи, хотя другие системы делают это автоматически, например: создавать инкрементные резервные копии;
- отсутствует встроенная поддержка XML или OLAP;
- для бесплатной версии доступна только платная поддержка;
- идеально подходит для: организаций, которым требуется надежный инструмент управления базами данных, но бесплатный.

1.4.3 Microsoft SQL сервер

Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL сервер 2016, Microsoft адаптировала продукт для операционной системы Linux, а на Windows-платформе он работал изначально.

Одной из уникальных особенностей версии 2016 года является temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные [10].

Достоинства:

- продукт очень прост в использовании;
- текущая версия работает быстро и стабильно;
- движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов;
- вы сможете получить доступ к визуализации на мобильных устройствах;
- он очень хорошо взаимодействует с другими продуктами Microsoft.

Недостатки:

- цена для юридических лиц оказывается неприемлемой для большей части организаций;
- даже при тщательной настройке производительности корпорация SQL Server способен занять все доступные ресурсы;

- сообщается о проблемах с использованием службы интеграции для импорта файлов;
- идеально подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

1.4.4 PostgreSQL

PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов. Это была одна из первых разработанных систем управления базами данных, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использован на большинстве основных платформ, включая Linux. Прекрасно справляется с задачами импорта информации из других типов баз данных с помощью собственного инструментария.

Движок БД может быть размещен в ряде сред, в том числе виртуальных, физических и облачных. Самая свежая версия, PostgreSQL 9.5, предлагает обработку больших объемов данных и увеличение числа одновременно работающих пользователей. Безопасность была улучшена благодаря поддержке DBMS_SESSION.

Достоинства:

- является масштабируемым и способен обрабатывать терабайты данных;
- поддерживает формат json;
- существует множество predefined функций;
- доступен ряд интерфейсов [11].

Недостатки:

- документация туманна, поэтому, возможно, ответы на некоторые вопросы придется искать в интернете;
- конфигурация может смутить неподготовленного пользователя.

- скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения;
- идеально подходит для организаций с ограниченным бюджетом, но квалифицированными специалистами, когда требуется возможность выбрать свой интерфейс и использовать json.

1.4.5 ЛИНТЕР

Линтер — российская СУБД, реализующая стандарт SQL:2003 (за исключением нескаллярных типов данных и объектно-ориентированных возможностей) и поддерживающая большинство операционных систем, в том числе семейство Windows, различные версии UNIX, ОС реального времени (включая QNX).

К особенностям можно отнести защиту данных: 2 класс защиты данных от несанкционированного доступа и 2 уровень контроля отсутствия недекларированных возможностей. Мандатный контроль доступа к данным на уровне таблиц, столбцов записей и отдельных полей записей. Управление доступом к рабочим станциям и устройствам хранения информации. Контроль доступа к СУБД по расписанию. Управление протоколированием операций над БД (аудит). Аутентификация пользователей через LDAP, Kerberos, средствами операционной системы. Хеширование паролей по алгоритму FIPS 180-2 SHA-224.

18 марта 2016 года по решению Экспертного совета по российскому программному обеспечению при Минкомсвязи России СУБД ЛИНТЕР включена в единый реестр российских программ для электронных вычислительных машин и баз данных (реестр российского ПО) [12].

Репликация асинхронная (в том числе и двунаправленная), возможна репликация с другими БД через ODBC.

Имеет утилиты конвертации, работающие через ODBC и ADO.NET. Конвертор из DBF-формата. Конвертор модели данных (из ERwin в ЛИНТЕР).

Достоинства:

- российская разработка;
- она поддерживает SQL:2003;
- облегчается конвертация при переходе с других СБУД;
- рекомендована "Единым реестром российских программ".

Недостатки:

- падение эффективности в случае высокой динамики изменений;
- идеально подходит для: отечественных организаций, которые работают с конфиденциальными и персональными данными.

1.4.6 РЕД База Данных

«РЕД База Данных» — российская СУБД, работает на всех основных платформах и ОС (Windows, Linux, BSD Unix, IBM AIX, HP-UX, Sun Solaris и т.д.). Система модульная. Имеет открытый исходный код.

Возможность «горячего» резервного копирования и инкрементного резервного копирования. Сертифицирована ФСТЭК России. Соответствует отечественным требованиям по защите информации. Может использоваться при создании информационных систем до класса защищенности 1Г включительно и при создании информационных систем персональных данных до 1 класса включительно. Полное соответствие принципам атомарности, непротиворечивости, изоляции, долговечности (ACID).

Имеются модули сопряжения практически для всех используемых сред разработки (драйверы ODBC, JDBC, C/C++, C#, Java, Delphi, PHP, Python, Perl, VB, и т.д.), результатов тестов этих модулей и гарантия стабильной работы.

Возможность работы во «встроенном» в ПО (embedded) локальном режиме в виде библиотеки DLL без отдельной установки и настройки СУБД, в т.ч. поддержка встраивания в виртуальную машину Java.

Достоинства:

- российская разработка;
- соответствует отечественным требованиям по защите информации;
- высокое быстродействие, сравнимое с лидерами рынка;
- возможность хранения базы данных в одном отдельном файле;
- идеально подходит для: отечественных организаций (включая оборонные), которые работают с конфиденциальными и персональными данными.

Недостатки:

- нет.

Вывод: проанализировав достоинства и недостатки наиболее популярных систем управления базами данных, я выбрала MySQL, так как уже изучала её основы, и данная СУБД обеспечивает всем необходимым функционалом для решения поставленных в моей бакалаврской работе задач.

1.5 Анализ языков программирования для создания сайта

После выбора средств разработки БД, встал вопрос о том, на каком языке программирования лучше написать сайт, который бы обеспечивал все необходимые функции по работе с моей БД как на стороне администратора, так и на стороне клиента.

Никто точно не скажет, какой язык программирования лучше всего подходит для конкретной цели. Это будет зависеть от множества факторов и в различных условиях тот или иной язык программирования будет иметь свои преимущества и недостатки.

Основными сравнительными критериями являются:

- специфичность;
- универсальность;
- эффективность;
- производительность.

В некоторых случаях очень важно оптимизировать производительность вашего кода, тем более при работе с большими объемами особо важных данных. Однако скомпилированные языки обычно намного быстрее, чем интерпретируемые. Аналогично, статически типизированные языки значительно более отказоустойчивы, чем динамически типизированные. Таким образом, единственным компромиссом является снижение производительности.

В некоторой степени, каждый из представленных ниже языков программирования обладает одним параметром в каждой из двух групп: универсальность - специфичность; производительность - удобство.

Учитывая эти основные принципы, давайте рассмотрим некоторые из наиболее популярных языков программирования, которые используются для работы с данными.

1.5.1 Python

В 1991 году Гвидо ван Россум представил язык программирования Python. С тех пор этот язык стал чрезвычайно популярным языком программирования общего назначения и широко используется в сообществе специалистов по данным.

Преимущества:

- Python – это очень популярный, широко используемый язык программирования общего назначения. Он имеет обширный набор специально разработанных модулей и широко используется разработчиками. Многие онлайн-сервисы предоставляют API для Python;

- Python очень прост в изучении. Низкий порог вхождения делает его идеальным первым языком для тех, кто занимается программированием;
- такие программные пакеты как pandas, scikit-learn и Tensorflow, делают Python надежным вариантом для современных приложений в области машинного обучения
- большое количество библиотек на все случаи (HTTP и FTP протоколы, работа с сетью, работа с изображениями и т.д.) [13].

Недостатки:

- типобезопасность. Python – это динамически типизированный язык, а это значит, что вы должны быть осторожными при работе с ним. Ошибки несоответствия типов (например, передача строки (string) в качестве аргумента методу, который ожидает целое число (integer)) могут время от времени случаться.

Например, в случае если имеются конкретные цели статистического анализа и анализа данных, то обширный набор пакетов языка R дает ему преимущество перед Python. Кроме того, существуют более быстрые и безопасные альтернативы Python среди языков программирования.

Python является хорошим вариантом для целей науки о данных (data science), и это утверждение справедливо как для начального, так и для продвинутого уровней работы в данной области. Большая часть науки о данных сосредоточена вокруг процесса ETL (извлечение-преобразование-загрузка). Эта особенность делает Python идеально подходящим для таких целей языком программирования. Библиотеки, такие как Tensorflow от Google, делают Python очень интересным языком для работы в области машинного обучения.

1.5.2 SQL

SQL («язык структурированных запросов») определяет, управляет и запрашивает реляционные базы данных. Язык появился в 1974 году и с тех

пор претерпел множество видоизменений, но основные его принципы остаются неизменными.

Преимущества:

- очень эффективен при работе с запросами, обновлениями, а также при обработке реляционных баз данных;
- SQL очень часто используется в различных приложениях, так что знакомство с ним может очень пригодиться. Модули, такие как SQLAlchemy, упрощают интеграцию SQL с другими языками.

Недостатки:

- синтаксис SQL может показаться достаточно сложной задачей для тех, кто привык к императивному программированию.
- существует множество различных вариаций SQL, таких как PostgreSQL, SQLite, MariaDB. Все они достаточно разные, поэтому ни о какой совместимости не может быть и речи.

SQL более полезен в качестве языка для обработки данных, чем в качестве передового аналитического инструмента. Тем не менее, так много процессов в области науки о данных зависит от ETL, а долговечность и эффективность SQL лишний раз свидетельствуют о том, что такой язык программирования должен знать каждый специалист по данным (data scientist) [14].

1.5.3 JavaScript

JavaScript – язык программирования, являющийся прототипно-ориентированным. Он отражает язык ECMAScript, чьим прототипом изначально и являлся. Первая вариация появилась ещё в 1995 году и с тех пор постоянно совершенствовалась, пока не пришла к нынешнему виду.

Базовой особенностью этого языка отмечается то, что на него повлияли другие (Python, Java и др.) языки программирования с целью придания максимального комфорта JavaScript и лёгкости в понимании его теми

пользователями, которые не имеют соответствующего образования и глубинных знаний – не программистами. JavaScript – официально зарегистрированная торговая марка компании Oracle [15].

Преимущества:

- ни один современный браузер не обходится без поддержки JavaScript;
- с использованием написанных на JavaScript плагинов и скриптов справится даже не специалист;
- полезные функциональные настройки;
- постоянно совершенствующийся язык – сейчас разрабатывается бета-вариация проекта, JavaScript2;
- взаимодействие с приложением может осуществляется даже через текстовые редакторы – Microsoft Office и Open Office;
- перспектива использования языка в процессе обучения программированию и информатике.

Недостатки:

- пониженный уровень безопасности ввиду повсеместного и свободного доступа к исходным кодам популярных скриптов;
- множество мелких раздражающих ошибок на каждом этапе работы. Большая часть из них легко исправляется, но их наличие позволяет считать этот язык менее профессиональным, сравнительно с другими;
- повсеместное распространение. Своеобразным недостатком можно считать тот факт, что часть активно используемых программ (особенно приложений) перестанут существовать при отсутствии языка, поскольку целиком базируются на нем.

1.5.4 PHP

PHP (Hypertext PreProcessor, препроцессор гипертекста) – язык программирования, исполняемый на стороне веб-сервера, спроектированный

Расмусом Лердорфом (Rasmus Lerdorf) в качестве инструмента создания динамических и интерактивных веб-сайтов. Этот язык оказался достаточно гибким и мощным, поэтому приобрёл большую популярность и используется в проектах любого масштаба: от простого блога до крупнейших веб-приложений в Интернете.

Код, написанный на PHP, направлен на выполнение двух задач:

- html-часть отвечает за внешний вид и отображение информации;
- php-часть, интегрированная в html, обеспечивает возможности интерактивности и динамику.

При этом подобные программные коды и, соответственно, составленные с их помощью проекты, являются легкими, эффективными, гибкими, многофункциональными, удобными в администрировании, редактировании, обслуживании [16].

На сегодняшний день подавляющее большинство сайтов, сервисов и приложений, а также такие популярные платформы как Joomla, Drupal, WordPress, 1С-Bitrix и UMI.CMS написаны именно на PHP-языке.

Преимущества:

- простота освоения, простой синтаксис;
- отличная совместимость и переносимость – php-коды работают одинаково хорошо с разными платформами;
- набор текста кода и его редактирование можно осуществлять в любом текстовом или html-редакторе;
- высокая гибкость, емкость и функциональность. PHP-программу можно составлять отдельно от разработки веб-страницы, без привязки, после чего совместить. Это существенно упрощает жизнь дизайнеров и программистов;
- многозадачность и широкие возможности – создание любых веб-приложений, блогов, гостевых книг, интернет-магазинов, сайтов, работа с редиректами, заголовками, pdf-документами, базами данных, электронной почтой и пр.

- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка.

Недостатки:

- не подходит для создания десктопных приложений или системных компонентов;
- имеет слабые средства для работы с исключениями;
- глобальные параметры конфигурации влияют на базовый синтаксис языка, что затрудняет настройку сервера и разворачивание приложений;
- объекты передаются по значению, что смущает многих программистов, привыкших к передаче объектов по ссылке, как это делается в большинстве других языков;
- веб-приложения, написанные на PHP, зачастую имеют проблемы с безопасностью.

Вывод: на основе сравнительного анализа различных языков программирования для работы с данными, я сделала выбор в пользу PHP. Данный язык ориентирован на создание веб-сайтов, совместим с MySQL и реализует те функции, которые я ставила своей целью.

2 Предметная область «Фитнес центр» и требования к ИС

Фитнес центр предоставляет услугу по обеспечению всем необходимым для занятий на тренажерах и другие мероприятия для укрепления здоровья. Желających тренироваться в фитнес центре много, а количество тренажеров и зон тренировок не так много, поэтому нужна информационная поддержка в организации проведения занятий. Запись на посещение фитнес центра клиентом производится заранее, обычно по телефону, что не всегда удобно для клиентов из-за некруглосуточного режима работы, а администраторы должны постоянно быть на связи и производить различные манипуляции с соответствующими записями, при этом возрастает вероятность ошибки.

Создаваемая информационная система должна обеспечить возможность записи на занятия и процедуры через сайт, база по тренажерам, тренерам, функции администрирования базы. Вход в систему можно сделать по номеру клубной карты. Клиенты смогут просматривать и записываться на нужные занятия. Администратор сможет добавлять тренеров, тренажеры и клиентов в базу.

В результате обзора фитнес центров г. Красноярска и г. Железногорска были выявлены следующие закономерности:

- существуют фитнес центры как с широким спектром предлагаемых услуг, таких как тренажерный зал, групповые занятия, SPA-процедуры, спортивное питание, сауны и др., так и узконаправленные организации, специализирующиеся на каком-то конкретном направлении;
- наиболее популярные фитнес центры имеют веб-сайт;
- многие веб-сайты обладают сравнительно узким функционалом без возможности онлайн записи на получение услуг.

3 Практическая часть

3.1 Даталогическое проектирование БД фитнес центра

Любые информационные системы включают подсистему хранения данных, которые обеспечивают сохранение состояний объектов бизнес-процессов и конфигурацию самой системы. Обычно подсистема хранения и обработки данных базируется на использовании какой-либо СУБД. Наиболее распространенными являются реляционные СУБД, где данные представляются в виде связанных таблиц. Будем использовать реляционную модель базы данных.

Проектирование реляционной базы данных означает, что необходимо определиться с тем, какие будут таблицы, и хранение каких данных о широких категориях объектов они будут обеспечивать, при этом одной записи будет соответствовать один объект, представленный как структура данных, состоящая из полей (атрибутов).

Проектирование базы данных должно обеспечить такое состояние объектов, чтобы не возникало противоречий и других аномалий (неадекватного отражения объектов), т.е. при этом определяются ограничения.

Можно выделить два подхода к проектированию реляционной базы данных. Во-первых, можно создать одну большую таблицу всех данных, которые могут быть сохранены в базе данных, так называемое универсальное отношение и затем, производя ряд преобразований нормализации получить несколько отношений (таблиц), удовлетворяющих, например, третьей нормальной форме (3НФ). Во-вторых, можно использовать анализ отношений между сущностями предметной области, при этом обычно строят диаграммы «сущность-связь», в процессе анализа приходят к определенному набору сущностей и отношений между ними, которые потом будут

представлять таблицы, в которых атрибуты будут представлены полями, также выделяют ключевые поля (первичные и внешние).

Использовать подход на основе универсального отношения обычно целесообразно при создании небольших баз данных, а второй, подход используют на практике наиболее часто. Вторым подходом используют сущности и отношения, вытекающие из постановки задачи, что облегчает процесс перехода от его требований к объектам базы данных.

Исходя из описания предметной области, можно выделить ряд сущностей: тренер (или тот, кто проводит процедуры), зона тренировок (или место процедуры), тренажер (оборудование), программа тренировок, клиент план занятий (расписание) и занятие (запись на занятие), размещения тренажеров по зонам, дополнительно администратор.

Каждой сущности присущ определенный набор атрибутов:

- тренер (фамилия, имя отчество, дата рождения, телефон, фотография);
- зона (название);
- тренажер (название);
- размещения тренажеров (тренажер, зона)
- программа (название);
- клиент (фамилия, имя отчество, дата рождения, телефон, номер клубной карты, код доступа);
- план (дата и время начала занятия, длительность в минутах, вид программы, зона, тренер);
- занятие (план, клиент, отметка о проведении).

Отношения между сущностями представим на диаграмме «сущность-связь». На диаграмме подчеркнуты атрибуты сущности, которые совместно могут однозначно идентифицировать ее объект, причем такие ключи могут быть как простыми, так и составными.

Для обеспечения связей между сущностями введем атрибут ID, который будем являться первичным ключом, а в связанных таблицах будем

использовать внешние ключи, имена которых будут заканчиваться на “_ID”. Так же переименуем таблицы и атрибуты на англоязычные и установим для них подходящие типы данных и иные ограничения. Таким образом, перейдем к даталогическому проектированию БД, которая будет реализована средствами СУБД MySQL.

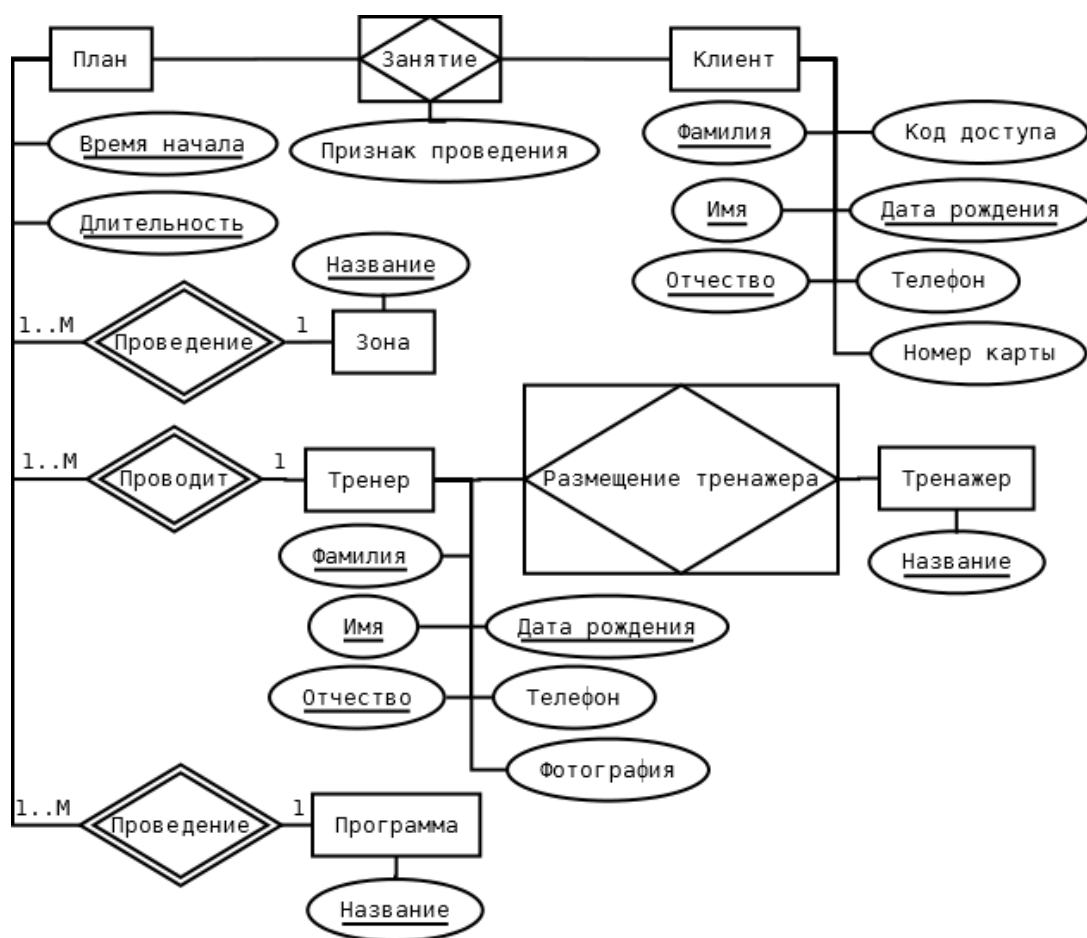


Рисунок 1 – Диаграмма «сущность-связь» БД «Фитнес центр»

Таблица 1 представляет собой справочник названий программ (видов) занятий.

Таблица 1 – Структура таблицы «program»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	Int unsigned	+	+	+	AUTO_INCREMENT
Название	Name_prog	varchar(255)		+	+	

Таблица 2 представляет собой справочник названий зон тренировок и мест проведения процедур.

Таблица 2 – Структура таблицы «zone»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	Int unsigned	+	+	+	AUTO_INCREMENT
Название	name_zone	varchar(255)		+	+	

Таблица 3 представляет собой справочник названий тренажеров и другого оборудования.

Таблица 3 – Структура таблицы «machine»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	Int unsigned	+	+	+	AUTO_INCREMENT
Название	name_mach	varchar(255)		+	+	

Таблица 4 представляет собой данные о размещении оборудования и тренажеров по зонам проведения процедур и тренировок.

Таблица 4 – Структура таблицы «mach_zone»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	Int unsigned	+	+	+	AUTO_INCREMENT
Id зоны	zone_id	Int unsigned		+	+	FK: `zone`(`id`)
Id тренажера	mach_id	Int unsigned		+		FK: `machine`(`id`)

Таблица 5 представляет собой данные о тренерах и другом персонале.

Таблица 5 – Структура таблицы «coach»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	Int unsigned	+	+	+	AUTO_INCREMENT
Фамилия	Fam	char(30)		+	+	
Имя	Im	char(30)		+		
Отчество	Otch	char(30)		+		
Дата рождения	dr	date		+		
Фото	photo	char(255)				
Телефон	tel	char(11)			+	

Таблица 6 представляет собой данные о клиентах.

Таблица 6 – Структура таблицы «client»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	Int unsigned	+	+	+	AUTO_INCREMENT
Фамилия	fam	char(30)		+	+	
Имя	im	char(30)		+		
Отчество	otch	char(30)		+		
Дата рождения	dr	date		+		
Номер карты	num_card	char(6)		+	+	
Код доступа, шифрованный двойным md5	access_code	varchar(32)		+		
Телефон	tel	char(11)		+	+	

Таблица 7 представляет собой данные о планируемых тренировках и процедурах.

Таблица 7 – Структура таблицы «plan»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	int unsigned	+	+	+	AUTO_INCREMENT
Дата и время начала	time_begin	datetime		+	+	
Id вида программы	program_id	int unsigned		+		FK: `program`(`id`)
Id зоны занятия	zone_id	int unsigned		+		FK: `zone`(`id`)
Id тренера	coach_id	int unsigned		+		FK: `coach`(`id`)
Число мест	cnt_places	int unsigned		+		
Длительность в минутах	minutes	int unsigned		+		

Таблица 8 представляет собой данные о записи клиентов на тренировки.

Таблица 8 – Структура таблицы «activity»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор	id	int unsigned	+	+	+	AUTO_INCREMENT
Id элемента плана	plan_id	datetime		+	+	FK: `plan`(`id`)
Id клиента	client_id	int unsigned		+		FK: `client`(`id`)
Признак получения услуги	is_get	tinyint(1)				

Следующие таблицы можно рассматривать как дополнение для обеспечения разграничения доступа и контроля доступа.

Таблица 9 представляет собой данные о клиенте, который наделен правом входа как администратор.

Таблица 9 – Структура таблицы «admins»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор клиента	client_id	int unsigned	+	+	+	FK: `client`(`id`)

Таблица 10 представляет собой данные о клиенте, который зарегистрирован.

Таблица 10 – Структура таблицы «authorized»

Описание	Имя поля	Тип данных	PK	Not Null	Unique	FK и триггеры и др.
Идентификатор клиента	client_id	int unsigned	+	+	+	FK: `client`(`id`)
Хеш	hash	varchar(32)				
IP-адрес клиента	ip	int		+		
Время входа	time_login	timestamp				

Можно обобщить сделанное и представить это в виде схемы данных, изображенной на рисунке 2.

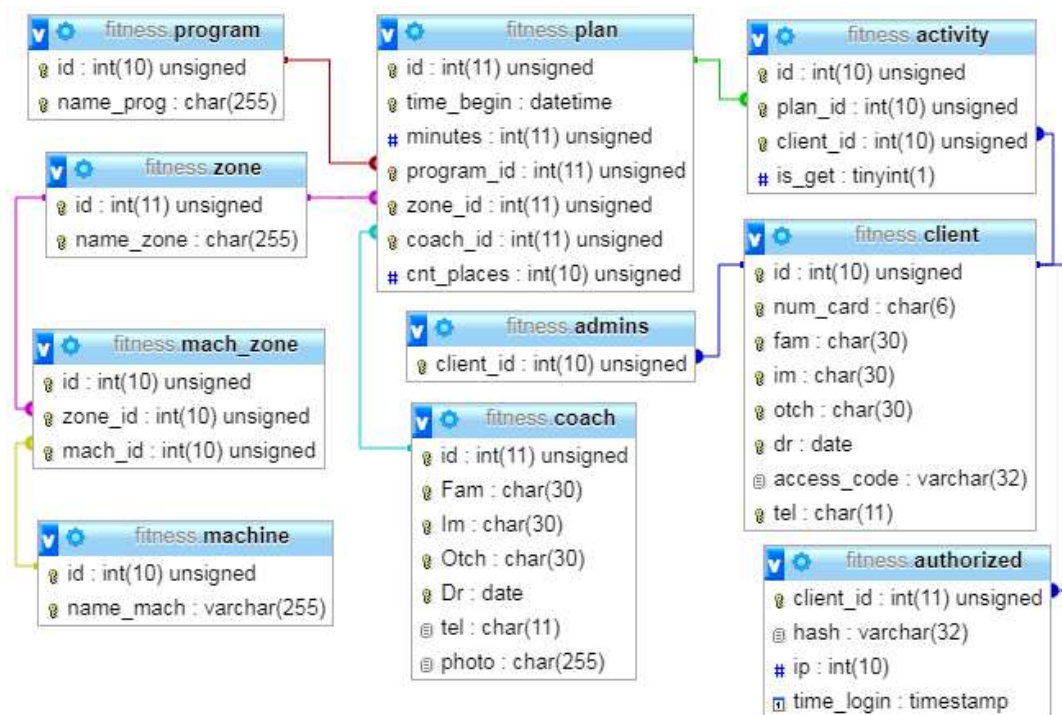


Рисунок 2 – Схема данных

3.2 Разработка сайта «Фитнес центр»

Разработка сайта в основном велась на языке PHP, а для подключения к СУБД MySQL использовался модуль mysqli. PHP был использован как средство генерации html контента по запросам от сервера Apache, получаемых в свою очередь от web-клиента (браузера).

Для ограничения доступа к некоторым особо важным файлам, которые обычно включаются в другие скрипты посредством функции require() был создан файл «.htaccess» в директории «inc».

Другие директории, в том числе корневая, так же были настроены для недопущения скачивания и просмотра скриптов.

Подключение к БД конфигурируется в файле «inc/config.php», там есть её небольшой массив с кодами ошибок при вводе в систему.

Само же подключение к БД осуществляется программой «inc/connect.php», которая выполняет некоторые настройки подключения и его успешность.

Поскольку доступ для пользователей должен быть разграничен, то была реализована авторизация с учетом, что у пользователя могут быть две роли, такие как администратор и клиент.

Для обеспечения управления доступом были разработаны:

- форма входа («inc/login_form.php»);
- форма выхода («inc/exit_form.php»);
- скрипт проверки («inc/check.php»).

Управление доступом использует cookies, задавая такие параметры как id, hash и is_admin. Эти параметры задают id клиента, hash для сверки и признак входа администратора, в структуре БД предусмотрено их сохранение и ещё сохраняется IP. Скрипт «inc/check.php» выполняет их серию и в случае несоответствия закрывает доступ и предлагает форму входа. Пользователь может сам выйти из системы для этого на каждой странице после успешного входа предлагается форма выхода.

После успешного входа администратора (это клиент, чей id есть в таблице admins) скрипт «inc/check.php» перенаправляет на скрипт «admin.php», который выводит страницу с возможными действиями по работе с таблицами БД.

После успешного входа клиента, не администратора скрипт «inc/check.php» перенаправляет на скрипт «client.php», который предоставляет возможность просмотра планируемых тренировок и записи или отмены их, а так же просмотра уже состоявшихся тренировок.

При формировании неотображаемой части страниц сайта используется скрипт «inc/header.php».

При формировании практически каждой страницы для вывода пользователю используется скрипт «inc/firstline.php», он включается непосредственно после тега <body> и выполняет формирование вывода верхней части страницы, которая включает картинку-ссылку, определяет вывод одной из форм входа или выхода, и показывает заголовок страницы.

Администратор имеет доступ к любой из таблиц БД и может просматривать их содержимое и вносить изменения, добавлять записи. Во многом эти действия схожи, поэтому была реализована возможность работы, через автоматизацию генерации соответствующих страниц исходя из параметров таблиц. Так для добавления или изменения записи (в случае события ON DUPLICATE KEY) были разработаны скрипт «act_add.php» для обработки формы, заданной в «inc/add_form.php». Правда пришлось сделать особые обработки для некоторых из полей. Эти скрипты используют сохранение состояния в сессии для восстановления данных в форме, в случае неудачного ввода данных, при этом настройка допустимости ввода значения производится на стороне сервера MySQL через механизм триггеров. Скрипт «act_add.php» включается require() в сценарии из директории add, имена которых соответствуют именам таблиц в БД.

Отображение данных и удаление отдельных записей администратором реализуется скриптом «inc/view_form.php», который для некоторой таблицы,

в зависимости от ее структуры, наличия комментариев к полям генерирует соответствующую табличную форму. Этот скрипт включается `require()` в сценарии из директории `view`, имена которых соответствуют именам таблиц в БД.

Для удобства работы с полями, представляющими внешние ключи, в случае генерации соответствующих форм на страницах был реализован скрипт `«inc/fks.php»`, который содержит ассоциативный массив для получения запросов по имени поля внешнего ключа.

Клиент не администратор работает с выведенным для него расписанием с учетом текущей даты и времени, это реализуется скриптом `«timesheet.php»`. Этот сценарий формирует табличную форму, в которой можно зарегистрироваться на тренировку, если ещё не было регистрации клиентом, посмотреть параметры тренировки (число мест, число уже записавшихся, зона тренировок, тренер и др.). Если регистрация клиентом уже была, то можно отменить ее. В случае если регистрация произошла, и при этом нет свободных мест, то клиент ставится в очередь и при отмене регистрации, стоящими в очереди перед ним, клиент будет продвигаться в очереди. Сценарий `«timesheet.php»` предоставляет возможность перехода на страницу просмотра фотографии тренера `«view_coach.php»`, фото тренеров хранятся в директории `coach`, а имена соответствующих файлов в поле БД `«coach.photo»`. Так же, сценарий `«timesheet.php»` предоставляет возможность перехода на страницу просмотра списка оборудования в соответствующей зоне тренировок (`«view_mach.php»`).

Клиент может просматривать тренировки, на которых он уже побывал, это реализуется сценарием `«past_training.php»`.

После успешной авторизации клиент или администратор может перейти на страницу `«index.php»` по ссылке вверху страницы в виде картинки, и для учета их функциональных возможностей были разработаны сценарии для включения через `require()`, соответственно `«inc/pg_menu_client.php»` и `«inc/pg_menu_admin.php»`.

В целом файловая структура представлена на рисунке ниже.

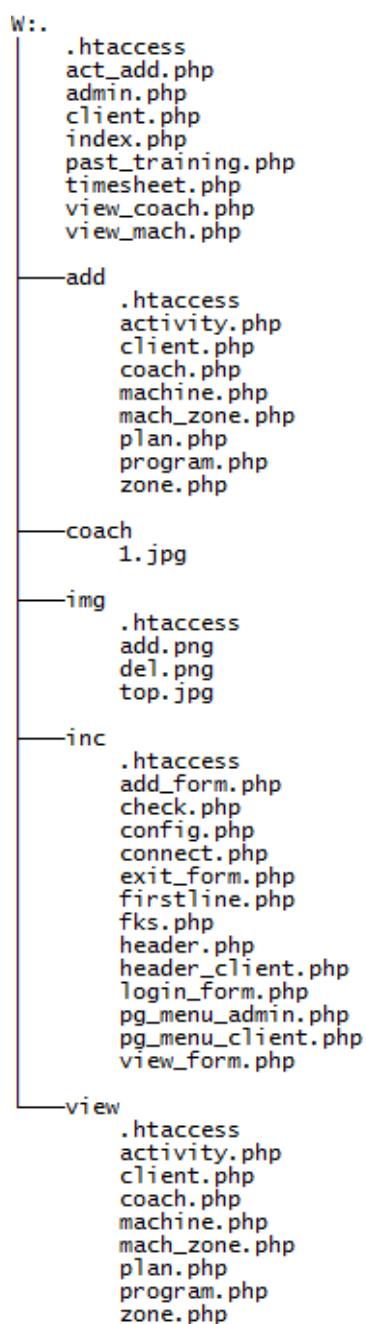


Рисунок 3 – Файловая структура

4 Сценарии работы с ИС «Фитнес центр»

Главная страница сайта представлена на рисунке 4.

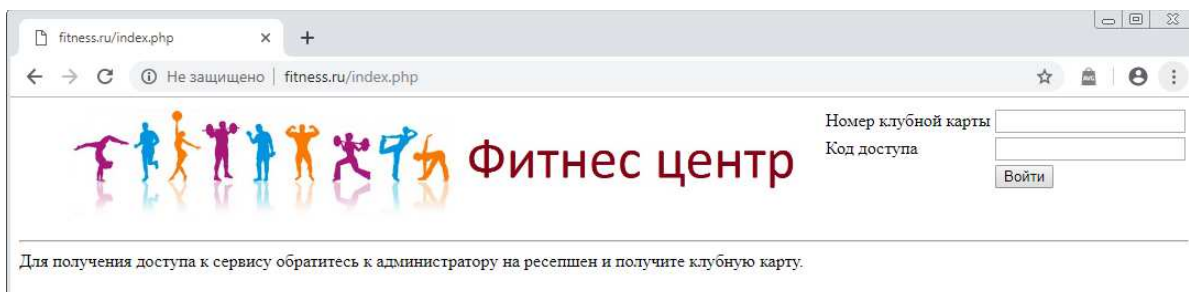


Рисунок 4 – Главная страница сайта

Ввод номера клубной карты admin, вход как администратора и кода доступа 123 представлен на рисунке 5.

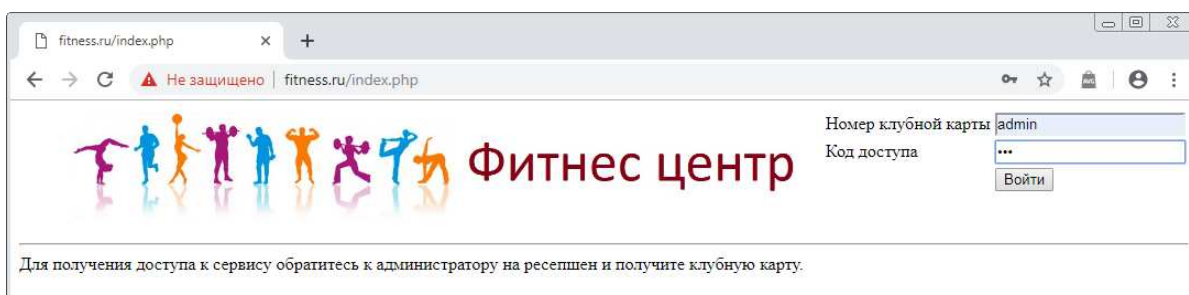


Рисунок 5 – Ввод кода доступа администратора

Результат перенаправления на страницу администратора представлен на рисунке 6.

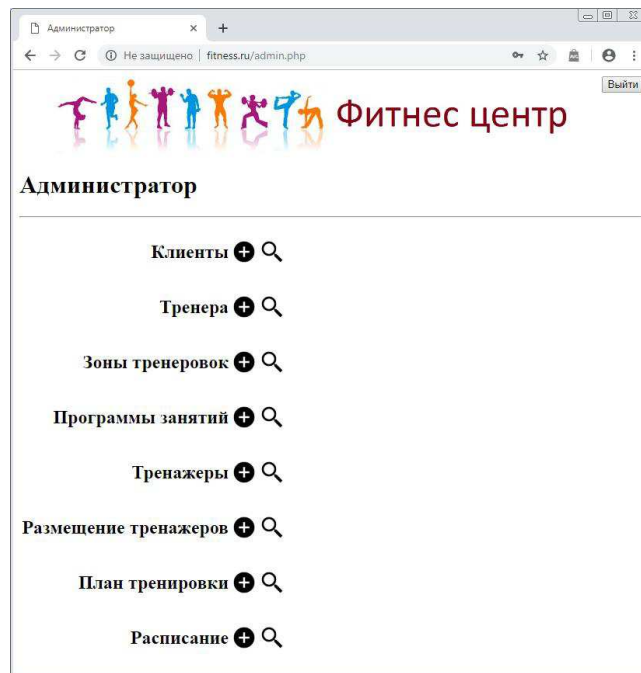


Рисунок 6 – Результат перенаправления на страницу администратора

На рисунке 7 представлен просмотр клиентов.

Акт	Номер клубной карточки	Фамилия	Имя	Отчество	Дата рождения	Код доступа	Телефон
✗	111111	Лебедев	Иван	Васильевич	2001-01-01	d9b1d7db4cd6e70935368a1efb10e377	89026448888
✗	222222	Макаров	Илья	Александрович	1992-02-02	a02ce9a3fc5def5275b5ca22f0d8f414	89026448885
✗	333333	Кошкин	Валерий	Петрович	1993-03-03	d930807e48a46653a72ccba6f5290bb1	89066598111
✗	444444	Шалимова	Светлана	Георгиевна	1994-04-04	3d2e942726362934909e65c45590106b	89066598114
✗	admin	Иванов	Иван	Иванович	1977-08-07	d9b1d7db4cd6e70935368a1efb10e377	89026448358

Рисунок 7 – Просмотр клиентов

Для перехода назад, можно нажать на верхнюю картинку. Теперь выберем добавление клиента. Результат перенаправления представлен на рисунке 8.

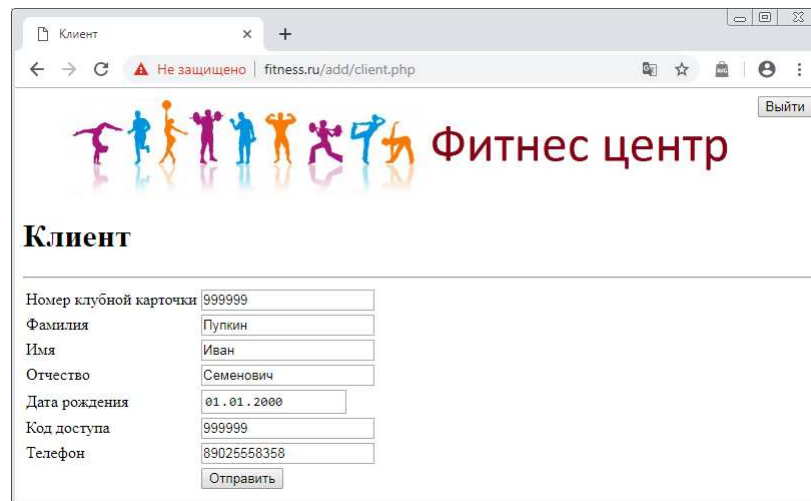


Рисунок 8 – Добавление клиента

Жмем кнопку «Отправить» и в случае успеха получаем уведомление о вставке записи. Результат представлен на рисунке 9.

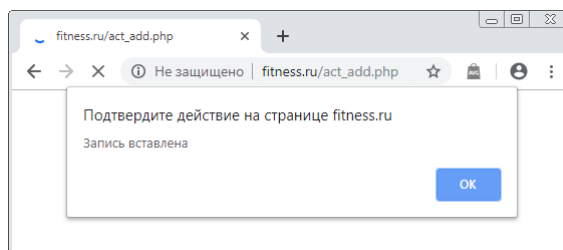


Рисунок 9 – Результат добавления клиента

Можно в просмотре убедиться, что такая запись имеется, и реализуем возможность ее удаления. Работоспособность данной функции проиллюстрирована на рисунках 10, 11.

Act	Номер клубной карточки	Фамилия	Имя	Отчество	Дата рождения	Код доступа	Телефон
X	111111	Лебедев	Иван	Васильевич	2001-01-01	d9b1d7db4cd6e70935368a1efb10e377	89026448888
X	222222	Макаров	Илья	Александрович	1992-02-02	a02cc9a3fc5def5275b5ca22f0d8f414	89026448885
X	333333	Кожкий	Валерий	Петрович	1993-03-03	d930807e48a46653a72ccbaf6f5290bb1	89066598111
X	444444	Шалимова	Светлана	Георгиевна	1994-04-04	3d2e942726362934909c65c45590106b	89066598114
X	999999	Пупкин	Иван	Семенович	2000-01-01	d5961a2f50e3b5422fb6effaf0769fea	89025558358
X	admin	Иванов	Иван	Иванович	1977-08-07	d9b1d7db4cd6e70935368a1efb10e377	89026448358

Рисунок 10 – Список клиентов до удаления записи

После удаления видим, что клиента «Пупкина» нет.

Act	Номер клубной карточки	Фамилия	Имя	Отчество	Дата рождения	Код доступа	Телефон
X	111111	Лебедев	Иван	Васильевич	2001-01-01	d9b1d7db4cd6e70935368a1efb10e377	89026448888
X	222222	Макаров	Илья	Александрович	1992-02-02	a02cc9a3fc5def5275b5ca22f0d8f414	89026448885
X	333333	Кожкий	Валерий	Петрович	1993-03-03	d930807e48a46653a72ccbaf6f5290bb1	89066598111
X	444444	Шалимова	Светлана	Георгиевна	1994-04-04	3d2e942726362934909c65c45590106b	89066598114
X	admin	Иванов	Иван	Иванович	1977-08-07	d9b1d7db4cd6e70935368a1efb10e377	89026448358

Рисунок 11 – Список клиентов после удаления записи

Аналогичные действия можно реализовать и с другими таблицами. Выйдем из системы.

Теперь зайдём в систему как клиент без административных прав с номером карты «11111» и кодом доступа «123». Данные действия изображены на рисунках 12, 13.

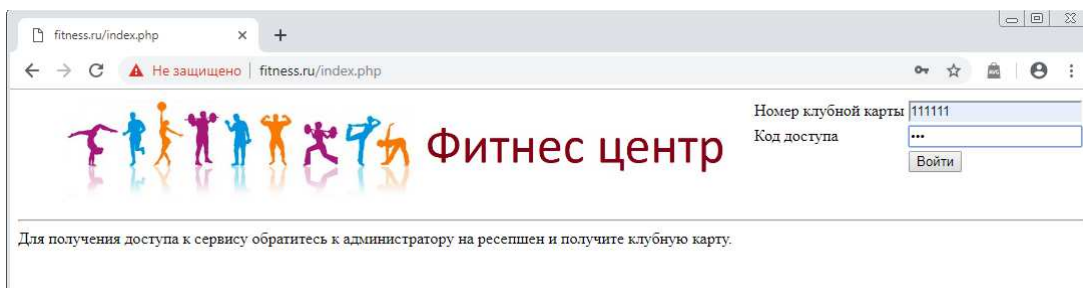


Рисунок 12 – Ввод кода доступа клиента

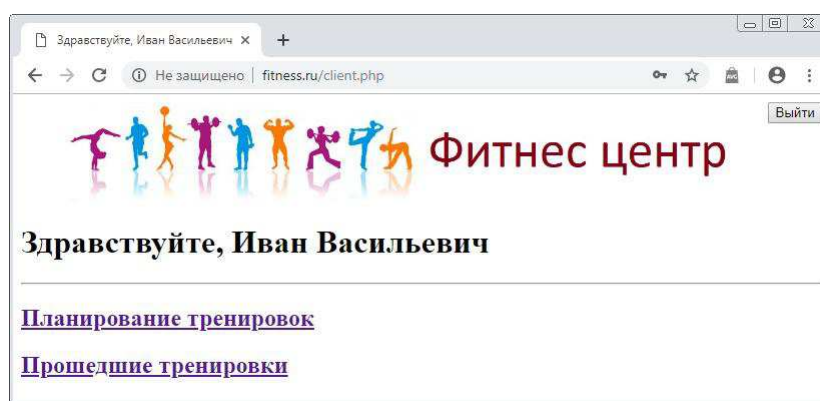


Рисунок 13 – Результат перенаправления на страницу клиента

Посмотрим, на какие тренировки можно записаться и запишемся на одну. Данная функция представлена на рисунке 14.

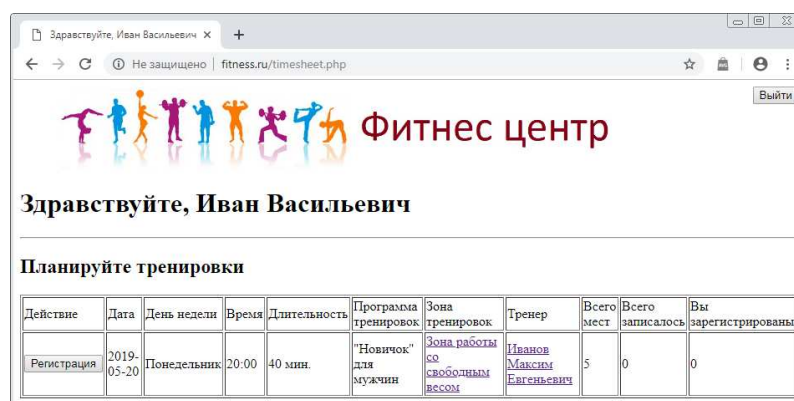


Рисунок 14 – Планирование тренировок клиентом

Можно предварительно просмотреть фотографию тренера и тренажеры в этой зоне. Результат просмотра представлен на рисунках 15, 16.

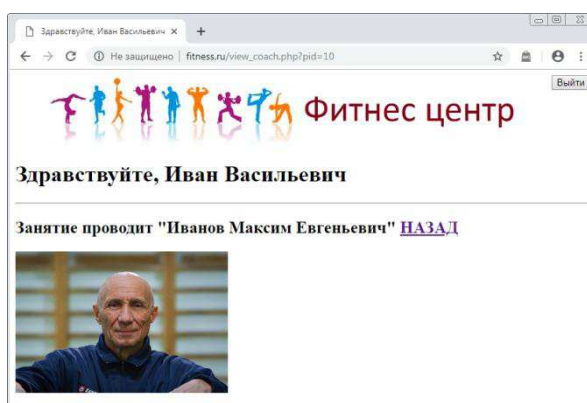


Рисунок 15 – Просмотр тренера

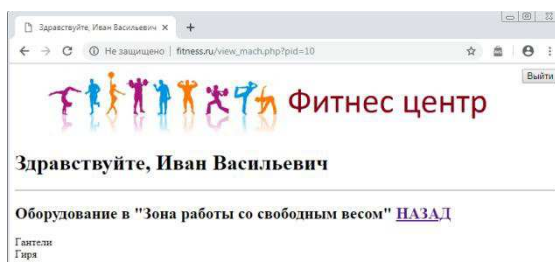


Рисунок 16 – Просмотр оборудования зоны для тренировок

После нажатия кнопки «Регистрация», видим, что состояние расписания изменилось. Результат представлен на рисунке 17. Теперь можно отменить регистрацию, отменим, и получаем прежнее состояние.

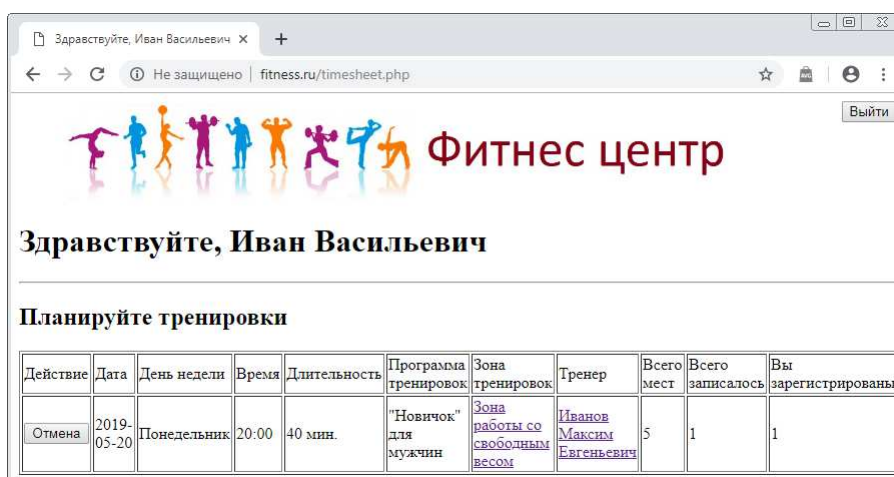


Рисунок 17 – Результат планирования тренировки после регистрации

Посмотрим предыдущие тренировки. Результат просмотра представлен на рисунке 18.

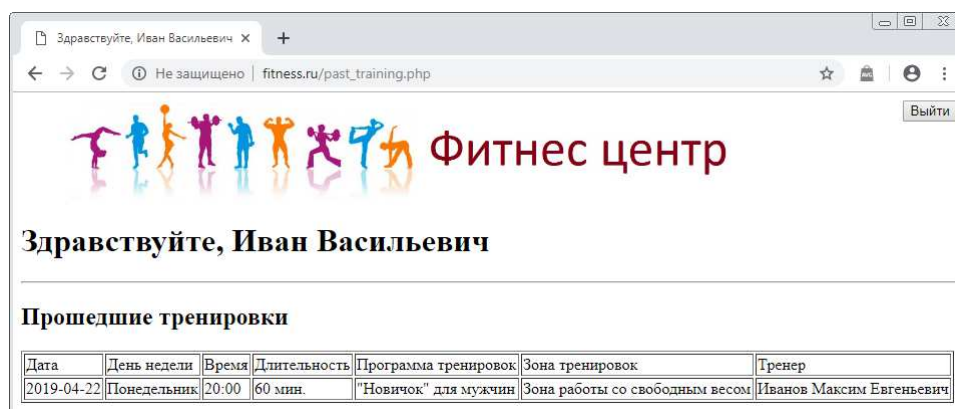


Рисунок 18 – История тренировок

Можно сделать вывод, что система обеспечивает основные сценарии работы с информационной системой.

ЗАКЛЮЧЕНИЕ

Таким образом, система обеспечивает основные функции работы с сайтом фитнес-центра «Лето» как со стороны администратора, так и со стороны клиента.

Результаты работы:

- В качестве базовой технологии выбрано использование серверного языка программирования PHP, сервера баз данных СУБД MySQL;
- Выполнена настройка серверов Apache и MySQL;
- Разработаны объекты БД для функционирования ИС, имеется скрипт для развертывания ее на сервере MySQL;
- Разработаны необходимые программные модули на языке PHP и проверена их работоспособность.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

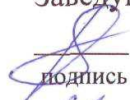
1. Кузнецов С. Д. Основы современных баз данных [Электронный ресурс] / С. Д. Кузнецов // Публикации и обзоры из области интернет-технологий, сетевых технологий, баз данных – Режим доступа: <http://www.citforum.ru/database/osbd/contents.shtml>.
2. Кириллов В. В. Основы проектирования реляционных баз данных [Электронный ресурс] / В. В. Кириллов // Публикации и обзоры из области интернет-технологий, сетевых технологий, баз данных – Режим доступа: <http://www.citforum.ru/database/dbguide/index.shtml>.
3. Пушников А. Ю. Введение в системы управления базами данных [Электронный ресурс] / А. Ю. Пушков // Публикации и обзоры из области интернет-технологий, сетевых технологий, баз данных – Режим доступа: <http://www.citforum.ru/database/dblearn/index.shtml>.
4. Ковязин А. Н. Мир InterBase. Архитектура, администрирование и разработка приложений баз данных в InterBase/Firebird/Yaffil : учебное пособие / А. Н. Ковязин. – Москва : КУДИЦ-ОБРАЗ, 2002. – 432 с.
5. Атре Ш. Структурный подход к организации баз данных : учебное пособие / Ш. Атре. – Москва : Финансы и статистика, 1983. – 317с.
6. Анализ существующих СУБД [Электронный ресурс] : Studbooks – Режим доступа: https://studbooks.net/2145018/informatika/analiz_suschestvuyuschih_subd.
7. Драч В. Oracle 12с [Электронный ресурс] / В. Драч // Сравнение современных СУБД – Режим доступа: <http://drach.pro/blog/hi-tech/item/145-#>.
8. Бурмистров А.В. Недостатки реляционных баз данных : учебное пособие / А.В. Бурмистров. – Москва: Наука, техника и образование, 2015. – 347с.
9. Основы создания баз данных MySQL [Электронный ресурс] : Основы MySQL – Режим доступа: <https://www.internet-technologies.ru/articles/osnovy-sozdaniya-baz-dannyh-mysql.html>.

10. Создание и удаление БД в MS SQL Server [Электронный ресурс] : Основы MySQL – Режим доступа: <https://www.internet-technologies.ru/articles/sozdanie-i-udalenie-bd-v-ms-sql-server.html>.
11. Достоинства PostgreSQL [Электронный ресурс] : StudFiles – Режим доступа: <https://studfiles.net/preview/6338058/page:3/>.
12. СУБД Линтер. Технический обзор [Электронный ресурс] : CITFORUM – Режим доступа: <http://citforum.ru/database/linter/overview/re112.shtml>.
13. Преимущества и недостатки Python [Электронный ресурс] : Язык программирования Python – Режим доступа: <https://lisiynos.github.io/s1/python.html>.
14. Язык SQL [Электронный ресурс] : StudFiles – Режим доступа: <https://studfiles.net/preview/4050050/page:18/>.
15. JavaScript [Электронный ресурс] : Википедия – свободная энциклопедия – Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript>.
16. Особенности и преимущества PHP [Электронный ресурс] : Язык программирования PHP – Режим доступа: https://depix.ru/articles/yazyk_programmirovaniya_php.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и Информационных Технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ




Заведующий кафедрой ИС


подпись инициалы, фамилия
« 21 » 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – «Информационные системы и технологии»

Разработка АИС фитнес-центра «Лето»

Руководитель	 21.06.19 подпись, дата	ст. препод. ИС должность, учёная степень	Ю. В. Шмагрис инициалы, фамилия
Консультант	 21.06.19 подпись, дата	доцент, к.т.н. должность, учёная степень	И. А. Легалов инициалы, фамилия
Выпускник	 21.06.19 подпись, дата		С.А. Юматова инициалы, фамилия
Нормоконтролер	 21.06.19 подпись, дата	ст. препод. ИС должность, учёная степень	Ю.В. Шмагрис инициалы, фамилия

Красноярск 2019