

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Разработка социальных и экономических информационных систем

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка информационной системы для резервного копирования MS SQL

Обучающийся

О.А. Автушко

(Инициалы Фамилия)

(личная подпись)

Руководитель

В.В. Дружинкин

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Тема выпускной квалификационной работы «Разработка информационной системы для резервного копирования базы данных MS SQL».

В процессе написания работы разработана информационная система «WeBase», предназначенная для выполнения резервного копирования.

Ключевые слова: информационная система, база данных, резервное копирование, моделирование, проектирование, IDEF0, UML, язык программирования Python, IDE PyCharm, веб-фреймворк Django, css – фреймворк Bootstrap, MS SQL, администратор базы данных, супер администратор.

Во введении обоснована актуальность выполняемой работы, определена главная цель, поставлены задачи, а также обозначены объект и предмет исследования.

В первой главе проведено исследование предметной области резервного копирования, рассмотрены существующие методы. Перечислены готовые программные продукты в данной сфере, составлен список их основных функций. С помощью IDEF0 методологии построена диаграмма «как должно быть». Поставлена задача на разработку интерфейса.

Во второй главе осуществлено логическое моделирование информационной системы. Далее было осуществлено проектирование базы данных. При моделировании использован язык графического описания UML, построены такие диаграммы: диаграмма прецедентов, диаграммы последовательности, диаграммы «сущность-связь». Сформулированы аппаратно-технические требования к системе.

В третьей главе осуществлено физическое проектирование разрабатываемой информационной системы, выбраны архитектура и технологии для разработки интерфейса. Представлен программный код созданного класса для таблицы в базе данных. В специальной среде

разработки PyCharm на языке Python с применением веб-фреймворка Django и Bootstrap, разработан интуитивно-понятный интерфейс с нейтральным дизайном «WeBase». Он предназначен для выполнения функций резервного копирования. Далее в работе описаны модули, из которых состоит система и охарактеризованы связи между ними, представлен программный код функций. Далее проведено успешное ручное тестирование нескольких модулей системы по составленному плану.

В приложении представлены скриншоты разработанных экранных форм приложения и HTML код для нескольких страниц.

В заключении описаны результаты работы и представлены выводы.

Объем выпускной квалификационной работы – 50 страниц, 20 рисунков, 6 таблиц, 4 рисунка в приложении и 4 листинга программного кода.

Содержание

Введение	6
1 Анализ предметной области резервного копирования	8
1.1 Понятие и методы резервного копирования	8
1.2 Обзор технологий резервного копирования MS SQL	12
1.3 Анализ существующих разработок в сфере резервного копирования СУБД MS SQL.....	14
1.3.1 Определение критериев анализа существующих разработок	14
1.3.2 Сравнительная характеристика разработок	14
1.4 Постановка задачи на разработку интерфейса	18
1.5 Разработка модели бизнес-процесса «как должно быть»	19
2 Логическое моделирование и проектирование веб-приложения	21
2.1 Выбор технологии для выполнения логического моделирования приложения.....	21
2.2 Логическая модель веб-приложения и ее описание	22
2.3 Проектирование базы данных веб-приложения	25
2.3.1 Разработка концептуальной модели данных веб-приложения	25
2.3.2 Обоснование вида логической модели веб-приложения.....	25
2.3.3 Разработка логической модели данных веб-приложения	26
2.4 Требования к аппаратно-программному обеспечению веб-приложения	27
3 Физическое проектирование веб-приложения	29
3.1 Выбор архитектуры веб-приложения.....	29
3.2 Выбор технологии для разработки веб-приложения.....	30
3.3 Разработка физической модели данных веб-приложения.....	31
3.4 Разработка веб-приложения.....	33

3.4.1	Схема взаимосвязи модулей приложения	33
3.4.2	Описание модулей веб-приложения с примерами кода функций	34
3.5	Описание функциональности веб-приложения	37
3.6	Тестирование разработанного веб-приложения	37
3.6.1	Выбор метода тестирования веб-приложения	37
	Заключение	41
	Список используемой литературы	42
	Приложение А Скриншоты экранных форм разработанного веб-приложения «WeBase»	45
	Приложение Б Листинги кода страниц HTML	47

Введение

Информация – ценнейший актив для любого современного предприятия. Совершенствование информационного взаимодействия является важнейшим вектором развития. Данный процесс влечет за собой рост объема используемой информации. В итоге растет опасность её утери или повреждения. Сохранение важной информации – предназначение функции резервного копирования. Для оптимизации данного процесса существует несколько способов:

- резервное копирование по расписанию;
- сжатие и шифрование резервных копий;
- тестирование резервных копий на предмет целостности.

Тема оптимизации резервного копирования актуальна как никогда. Всё потому что обеспечение надежности систем и безопасности данных являются важнейшими задачами для современного предприятия. Необходимо иметь возможность быстрого и эффективного восстановления утерянных или поврежденных данных.

Объектом исследования является процесс разработки информационной системы для резервного копирования базы данных MS SQL.

Предмет исследования – проектирование и разработка информационной системы для резервного копирования базы данных MS SQL.

Цель исследования – изучение методов резервного копирования и разработка информационной системы с функционалом для резервного копирования. Для достижения изложенных целей потребуется решение таких задач:

- изучение и анализ предметной области резервного копирования;
- обзор существующих программных продуктов;
- построение логической и физической моделей разрабатываемой ИС для управления базой данных и реализации резервного копирования;

- разработка функциональной ИС для осуществления резервного копирования;
- тестирование разработанной ИС.

Методы исследования:

- объектно-ориентированный анализ и проектирование;
- методы оптимизации резервного копирования;
- CASE-технологии для анализа и проектирования;

Практическая значимость этого дипломного проекта состоит в том, что созданный информационный инструмент может оказаться полезным на действующем предприятии. Он позволит оптимизировать процесс резервного копирования данных, уменьшив время выполнения этой операции и снизив вероятность ошибок.

Структура выпускной квалификационной работы: введение, три главы, заключение, список использованных источников и приложение.

В первой главе исследовательской работы была проанализирована предметная область резервного копирования, сравнительный анализ существующих разработок. Была поставлена задача на разработку.

Во второй главе осуществлено логическое моделирование ИС, проектирование базы данных и изложены аппаратно-программные требования к системе.

В третьей главе проведено физическое моделирование, разработка и тестирование ИС.

В приложении представлены скриншоты готового интерфейса программы «WeBase» и HTML-код нескольких страниц приложения.

1 Анализ предметной области резервного копирования

1.1 Понятие и методы резервного копирования

Резервное копирование - это процедура создания точной копии данных на носителе или в файловой системе компьютера. Главная цель данной процедуры – защита от потери или повреждения важных данных и оперативное их восстановление в случае непредвиденных событий. «Резервное копирование данных очень важная операция, которую должен производить любой пользователь через определенный промежуток времени» [4].

Процесс копирования может быть выполнен как на локальном компьютере (на жесткий диск, съемный носитель), так и на удаленном сервере или облаке. В зависимости от типа данных, резервное копирование может включать в себя копирование файлов, баз данных, конфигураций программного обеспечения и других важных данных.

Своевременное создание резервных копий позволит вовремя восстановить данные в случае их потери или повреждения. Данная процедура поможет защитить данные от потери и упростит процесс их восстановления [1].

На текущий момент времени существует несколько подходов к организации резервного копирования данных. Ниже представлено краткое описание традиционных и более новых вариантов выполнения данной процедуры.

«База данных - совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации» [6].

Полное резервное копирование (Full Backup) показано на рисунке 1. Следуя этому методу, создается полная копия всех данных на определенный

момент времени. Логично использовать перед глобальными изменениями в системе. Главный минус – необходим большой объем памяти [19].

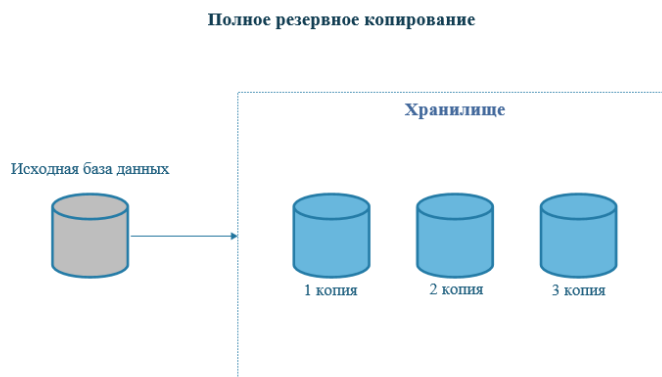


Рисунок 1 - Схема полного резервного копирования

Инкрементальное резервное копирование (Incremental Backup) представлено на рисунке 2. Данный метод предполагает формирование копии только тех данных, которые были изменены с момента последнего резервного копирования. Является более удобным, чем полное копирование, поскольку допускает использование меньшего объема памяти и требует меньше времени [13].



Рисунок 2 - Схема инкрементального резервного копирования

Реверсивное инкрементальное копирование (Reverse Incremental Backup) показано на рисунке 3 [16]. Более сложный и более новый метод, который включает в себя такие этапы:

- создается полная копия;
- последующие инкременты внедряются в первоначально созданную полную копию;
- замещенные таким образом блоки данных в полной копии сохраняются как предшествующие ей.

Главными плюсами данного метода являются: более гибкое управление объемом используемой памяти и более высокая безопасность [20].

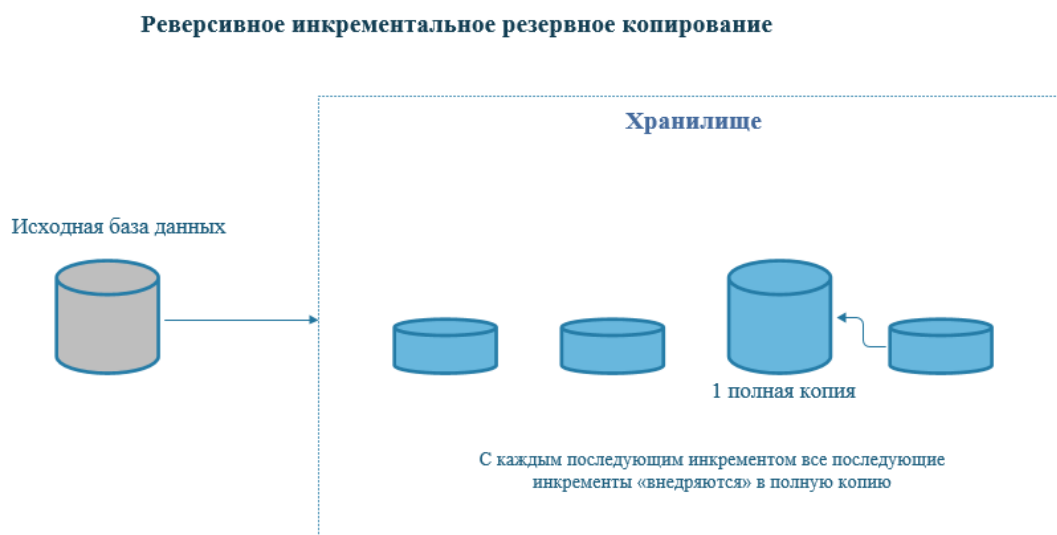


Рисунок 3 - Схема реверсивного инкрементального копирования

Синтетическое полное резервное копирование (Synthetic Full Backup) описано на рисунке 4. Данный метод начинается также с полного резервного копирования. Далее следует серия инкрементных копий, которые в заданный момент времени объединяются с полной копией, которая в результате становится исходной. Главным преимуществом данного метода является снижение нагрузки на сеть и экономия пространства для хранения [21].

Синтетическое полное резервное копирование



Рисунок 4 - Схема синтетического резервного копирования

Дифференциальное резервное копирование (Differential Backup) показано на рисунке 5. Создание полной резервной копии является первоначальным этапом дифференциального копирования. Далее, в отличие от инкрементного копирования, создаются копии изменений с момента первоначальной копии. У данного метода главным плюсом является более высокие надежность и скорость, в сравнении с инкрементным копированием [13].

Дифференциальное резервное копирование



Рисунок 5 - Схема дифференциального резервного копирования

Кроме того, существуют пользовательские методы и стратегии резервного копирования. Например, стратегия «Дед-отец-сын» («Grandfather-Father-Son», GFS). Она предполагает создание резервных копий в несколько шагов:

- создание копии 1 раз в месяц (с сохранением на отдельный носитель);
- создание копии 1 раз в неделю на стандартное хранилище;
- инкрементное или дифференциальное копирование раз в день.

Подобных стратегий существует много. При выборе метода и стратегии для создания резервных копий, необходимо учитывать все плюсы и минусы. Также нужно брать в расчет ресурсы, выделенные для работы системы. Это пропускная способность сети, объем памяти, вычислительные способности оборудования [10]. Далее проведем обзор существующих технологий резервного копирования для конкретной системы - MS SQL.

1.1 Обзор технологий резервного копирования MS SQL

Microsoft SQL Server (MS SQL) – это специальный продукт от компании Microsoft, предназначенный для управления реляционными базами данных с помощью языка запросов – SQL [22]. Внутри программы уже имеются инструменты для резервного копирования (агент SQL Server), но они могут быть сложны для использования. Поэтому важно, чтобы интерфейс программы для резервного копирования был интуитивно понятен [5]. Рассмотрим несколько отдельных программных решений для выполнения данной задачи:

- SQL Backup And FTP удобный инструмент, который позволяет создавать инкрементные резервные копии, сжимать и шифровать данные перед отправкой, а также планировать задания для резервного копирования [18];

- Handy Backup позволяет создавать инкрементальные резервные копии, имеет простой и удобный интерфейс, который позволяет легко

настроить резервное копирование и восстановление. Функционал сильно зависит от версии продукта [9];

- EaseUS Todo Backup Business также является простым и мощным инструментом для резервного копирования и восстановления данных. Программа предлагает дополнительные функции: автоматическое обновление резервных копий, шифрование и др.;

- LiteSpeed для SQL Server обещает максимально быстрое сжатие и восстановление данных;

- SQL Backup – программное обеспечение для резервного копирования баз данных MS SQL, которое дает возможность создавать полные и инкрементальные копии, автоматически их восстанавливать. Также поддерживает функции шифрования и сжатия;

- IDERA SQL Safe Backup также является утилитой для работы с базой MS SQL. Позволяет создавать копии баз данных, журналов транзакций и системных объектов. Можно настроить автоматическое копирование, сжатие и шифрование данных;

- Vembu BRD отличается от других решений своей модульной архитектурой, благодаря чему данная программа подходит как для крупных компаний, так и для небольших организаций. Основные недостатки: отсутствие ресейлеров в РФ, поддержка и документация на английском языке, отсутствие возможности автоматического тестирования копий [11].

Все описанные программные решения имеют плюсы и минусы. Каждое из них имеет дополнительный функционал, который может упростить и улучшить процесс резервного копирования базы MS SQL.

Далее проведем сравнительный анализ представленных программных продуктов.

1.2 Анализ существующих разработок в сфере резервного копирования СУБД MS SQL

1.2.1 Определение критериев анализа существующих разработок

Для проведения полноценного анализа существующих разработок в сфере резервного копирования СУБД MS SQL необходимо определить его критерии. Существующие системы резервного копирования и восстановления данных различны по функциям. Каждая из них имеет свои плюсы и минусы. Рассмотрим основные критерии для оценки эффективности существующих разработок:

- Функциональность (наличие основных функций для копирования и восстановления данных);
- Удобство (понятность интерфейса);
- Автоматизация (наличие функций для автоматизации процесса резервного копирования);
- Шифрование (наличие функции шифрования копии);
- Сжатие (наличие функции сжатия копии);
- «Облачное» хранилище (наличие интеграции с «облачным» хранилищем);
- Стоимость (наличие пробной версии, приемлемая стоимость полной версии).

Также существуют дополнительные функции: возможность клонирования жесткого диска, поддержка VMware и Hyper-V, наличие инструмента аутентификации и т. д.

1.2.2 Сравнительная характеристика разработок

Проведем сравнение существующих на данный момент программных продуктов для осуществления функций резервного копирования. Для этого необходимо составить сравнительную таблицу с характеристикой каждого из продуктов. 1.

Таблица 1 – Сравнительная характеристика программных продуктов

Название продукта	Функциональность	Удобство	Автоматизация	Шифрование	Сжатие	«Облачное» хранилище	Стоимость	Дополнительный функционал
SQL Backup And FTP	полное, дифференциальное резервное копирование, копирование журналов транзакций	+	+	+	+	+	- наличие бесплатной версии; -стоимость зависит от версии (от 39 до 500 долларов США за версию).	- возможность резервного копирования файлов; - отправка отчета по электронной почте.
Handy Backup	полное, дифференциальное, инкрементное или смешанное резервное копирование	+	+	+	+	+	- 30 дней пробного периода; - стоимость зависит от версии (от 39 до 299 долларов США за версию); - MS SQL поддерживается только в двух версиях (249 и 299 долларов США).	- подключение к базе напрямую; - уведомление по электронной почте; -выборочное восстановление.

Продолжение таблицы 1

EaseUS Todo Backup Business	дифференциальное, инкрементальное резервное копирование.	+	+	+	-	+	- бесплатная версия с урезанным функционалом; - стоимость зависит от функционала (от 49 до 299 долларов США в год)	- возможность копирования диска;
LiteSpeed для SQL Server	полное резервное копирование	+	+	+	+	+	- бесплатная пробная версия; - стоимость не указана на официальном сайте.	- возможность восстановления отдельных объектов; - проверка резервных копий на повреждения; - стратегии резервного копирования.
SQL Backup	полное, дифференциальное резервное копирование и копирование журналов транзакций.	+	+	шифрование не доступно в базовой версии	+	+	- базовая версия бесплатна; - стоимость от 49 до 399 долларов за стандартные версии; - существует версия Pro, стоимость не указана.	- справки и подсказки, встроенные в интерфейс; - уведомления на электронную почту.

Продолжение таблицы 1

IDERA SQL Safe Backup	полное резервное копирование	+	+	+	+	+	- 1036 долларов США за экземпляр; - 14-дневная пробная версия; - только платное использование; - необходима лицензия для каждого сервера.	- возможность работы с виртуальной БД;
Vembu BRD	полное резервное копирование	+	+	+	-	+	-1,8 доллара США в месяц; - 30 минут демо-версии.	- работа в виртуальных средах (VMware или Hyper-V); - возможность резервного копирования данных рабочих мест и серверов.

Вывод: помимо полного резервного копирования некоторые программы дают возможность выполнить инкрементальное или дифференциальное. Также несколько продуктов предоставляют выбор стратегии резервного копирования. Все программы из нашего списка позволяют создавать копии по расписанию и шифровать их. Некоторые имеют функцию сжатия копий. Функционал многих программ зависит от версии, есть бесплатные варианты с небольшим набором функций. От функционала зависит стоимость подписки или экземпляра. Выбор определенного продукта зависит от задач, выполняемых на предприятии и его финансовых возможностей.

1.3 Постановка задачи на разработку интерфейса

Цель выпускной квалификационной работы: разработка интуитивно-понятного интерфейса для взаимодействия с базой данных MS SQL с целью резервного копирования данных. Разрабатываемый интерфейс должен содержать необходимые кнопки и гиперссылки для перехода между объектами конфигурации, с отображением актуальной информации.

Данный интерфейс должен упростить и ускорить работу администратора базы данных, взаимодействующего с базой данных.

Функции, которые необходимо реализовать:

- формы;
- авторизация и аутентификация администратора базы данных;
- функция резервного копирования;
- функция создания расписания для резервного копирования.

Выпускная квалификационная работа должна содержать:

- описание используемых технологий;
- описание процесса проектирования;
- описание процесса реализации.

Для результативного выполнения данной выпускной квалификационной работы необходимо:

- провести логическое и физическое проектирование информационной системы;
- выбрать и описать основные технологии, которые будут применены при разработке веб-приложения;
- разработать функциональное веб-приложение, необходимое для резервного копирования базы данных MS SQL;
- описать основные построенные модули системы;
- описать основные функциональные возможности разработанного приложения;
- произвести тестирование реализованного веб-приложения.

1.4 Разработка модели бизнес-процесса «как должно быть»

Чтобы создать информативную модель необходимого бизнес-процесса, применим метод IDEF0. На рисунке 6 представлена диаграмма данного стандарта «как должно быть»

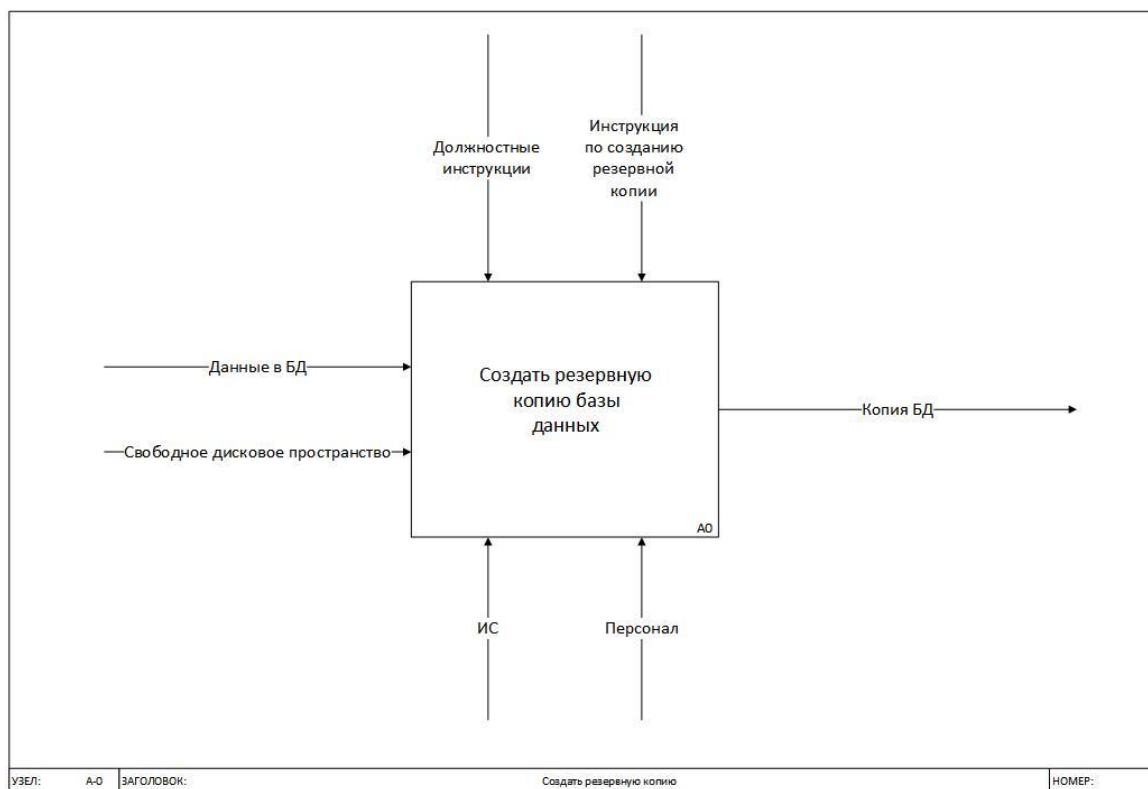


Рисунок 6 – Диаграмма «как должно быть»

На представленной диаграмме определены информационные потоки. Поток управления – Должностные инструкции и инструкции по созданию резервной копии. Входящие потоки – данные в базе и свободное дисковое пространство. Механизмы – информационная система и персонал. Выходящий поток – копия базы данных. Далее совершим декомпозицию 1 уровня (рисунок 7).

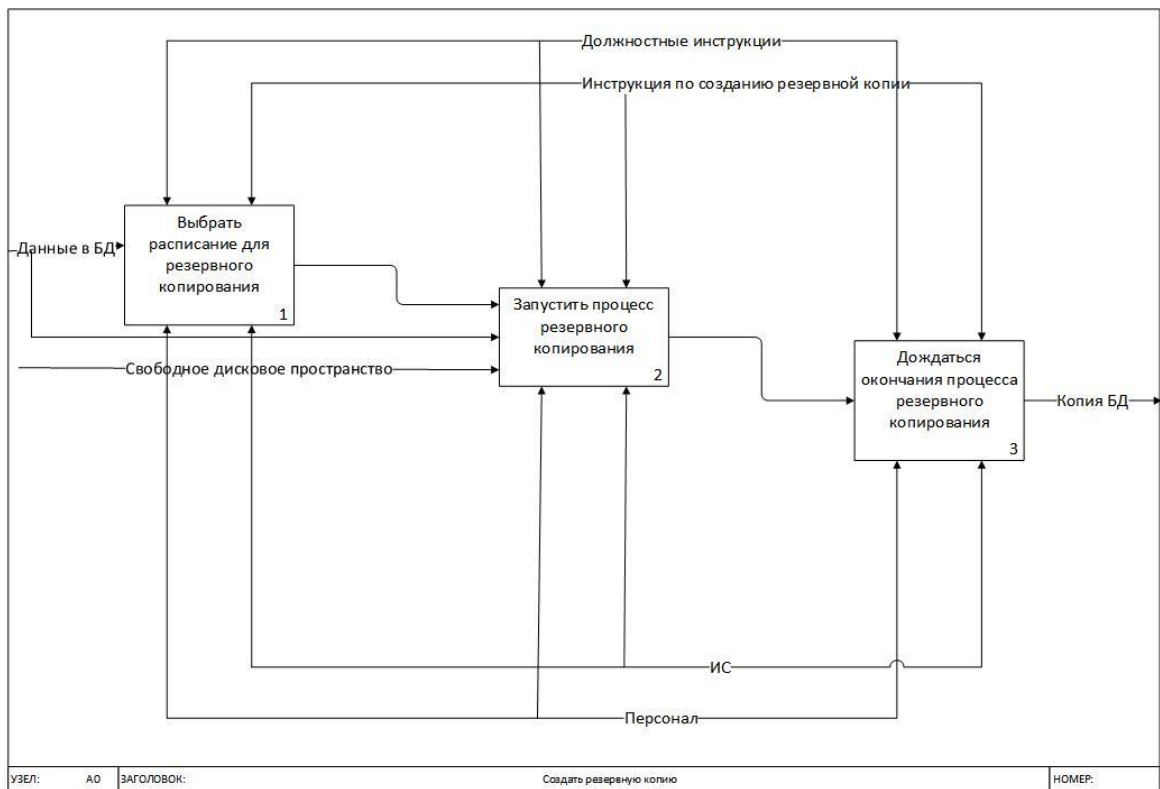


Рисунок 7 – Декомпозиция 1 уровня диаграммы «как должно быть»

На данном рисунке мы видим, что процесс создания резервной копии должен состоять из таких подпроцессов:

- Выбрать расписание для создания резервной копии;
- Запустить процесс резервного копирования;
- Дождаться окончания процесса резервного копирования.

Вывод по главе 1. В данной главе было рассмотрено понятие резервного копирования, описаны основные его методы. Также был составлен список существующих программных продуктов, предназначенных для резервного копирования баз MS SQL. Далее проведен сравнительный анализ данных продуктов. После была поставлена задача на разработку интерфейса для выполнения резервного копирования.

2 Логическое моделирование и проектирование веб-приложения

2.1 Выбор технологии для выполнения логического моделирования приложения

Логическое моделирование является важнейшим этапом разработки приложения. Данный этап позволяет определить, как будет функционировать система. Он позволяет понять цели и структуру программы, внести ясность в требования к системе. Необходимо построить логическую модель приложения и модели его информационной базы.

Моделирование процессов выполняется с помощью специальных программ и CASE-средств. К ним относятся BPwin, StarUML, Rational Rose, Microsoft Visio.

«В концептуальном и логическом проектировании базы данных широко применяется модель высокого уровня, основанная на понятиях модели "сущность-связь"» [6]. В данной дипломной работе в качестве инструмента визуального моделирования будет использована программа StarUML. Он является наиболее универсальным, так как позволяет создавать любые необходимые диаграммы. Также StarUML обладает удобным и интуитивно понятным интерфейсом, что весьма облегчает процесс моделирования. С помощью унифицированного языка моделирования UML сложный процесс проектирования информационных систем становится проще и интереснее [23].

Для осуществления логического моделирования необходимо построить:

- диаграмму прецедентов (use case diagram), которая покажет отношения между пользователем и приложением;
- диаграммы последовательности (sequence diagram), которая описывает все способы взаимодействия пользователя с приложением и приложения с базой данных.

2.2 Логическая модель веб-приложения и ее описание

Для полного понимания логического взаимодействия администраторов с системой создадим диаграмму прецедентов. Диаграмма отображена на рисунке 8.

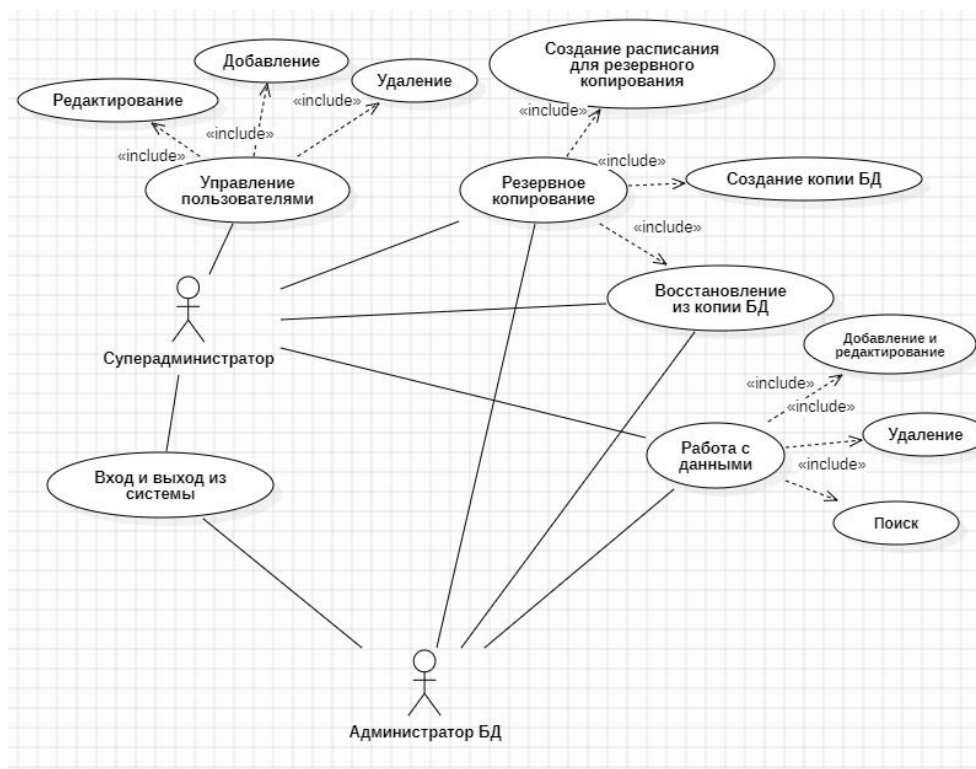


Рисунок 8 – Диаграмма прецедентов

На данной диаграмме мы видим все события, осуществляемые системой, при взаимодействии с участниками.

Диаграммы последовательности, представленные на рисунках ниже, более детально демонстрируют, как пользователь взаимодействует с приложением в рамках прецедентов. Рассмотрим основные операции в приложении.

На рисунке 9 мы видим, как пользователь взаимодействует с приложением на примере использования функции аутентификации. Данная

функция дает возможность администратору зайти в приложение для выполнения основных функций.

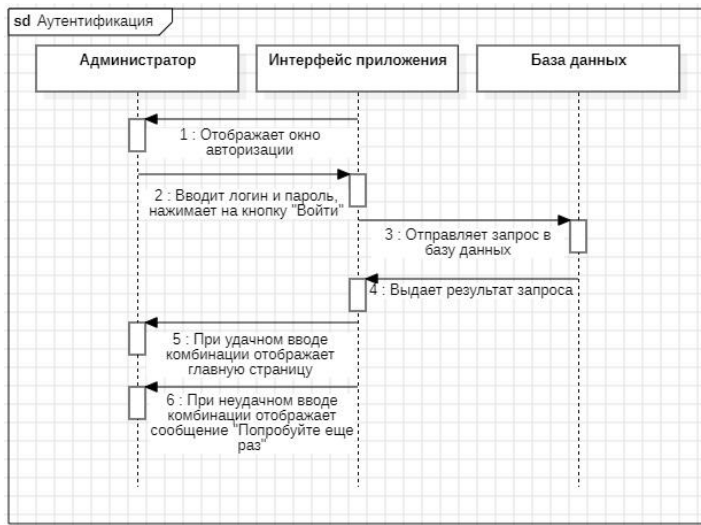


Рисунок 9 – Процесс аутентификации в приложении

Ниже (рисунок 10) представлены процессы создания резервной копии из существующей базы данных. В результате выполнения процесса резервного копирования программа создает файл json, который содержит все данные из базы.

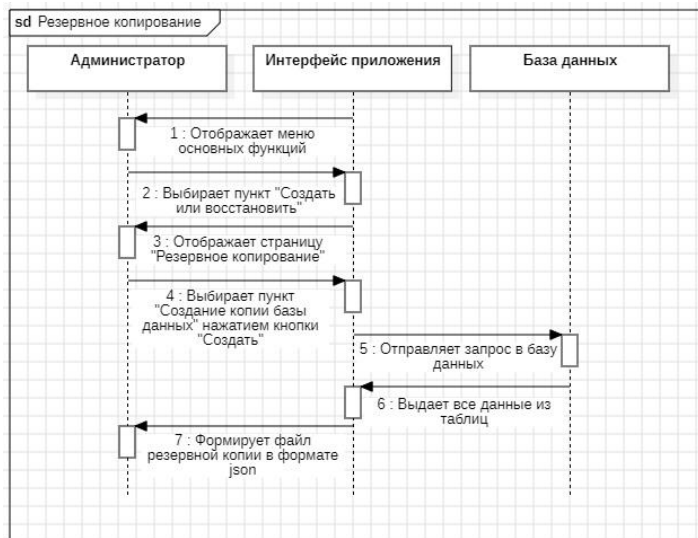


Рисунок 10 – Процесс создания резервной копии

На рисунке 11 представлен процесс восстановления данных в базе из резервной копии.

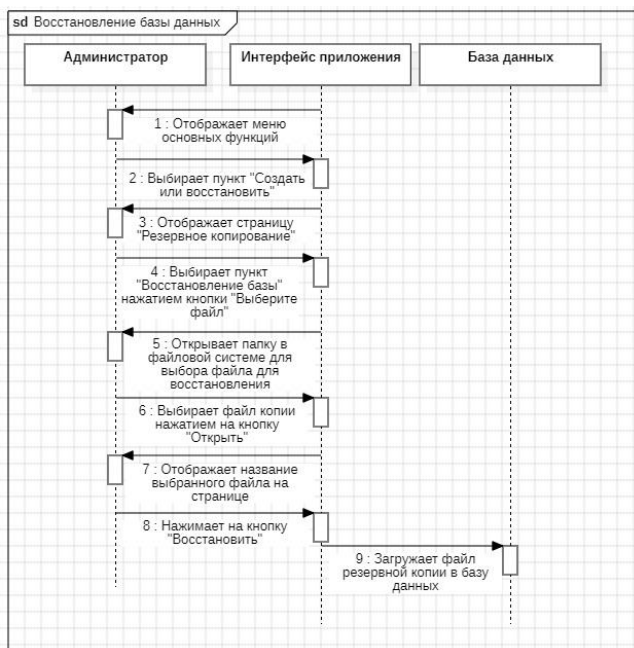


Рисунок 11 – Процесс восстановления базы из резервной копии

На рисунке 12 изображен процесс создания расписания для резервного копирования.

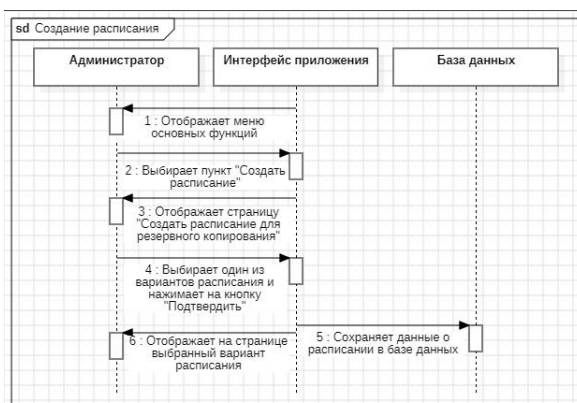


Рисунок 12 - Процесс создания расписания для резервного копирования

В результате выполнения данной функции администратор базы данных имеет возможность автоматизировать процесс резервного копирования.

2.3 Проектирование базы данных веб-приложения

2.3.1 Разработка концептуальной модели данных веб-приложения

«Модель данных – это совокупность допустимых структур данных и операций над ними, поддерживаемая компьютерной средой (в т. ч. СУБД) для определения логической структуры базы данных и динамического моделирования состояний предметной области» [4].

Концептуальная модель, несомненно, необходима для выполнения исчерпывающего описания всех сущностей системы и связей между ними. Этап концептуального проектирования данных считается наиболее важным для создания функционального приложения. На рисунке 13 представлена данная модель в виде диаграммы «сущность-связь» (Entity-Relationship Diagram).

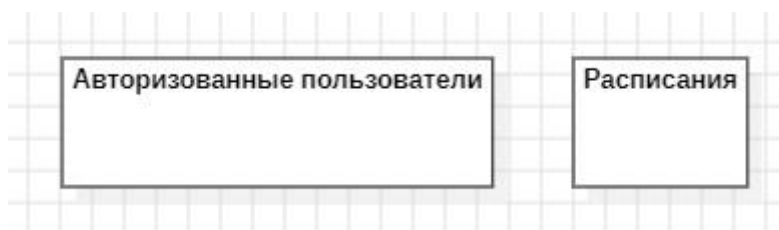


Рисунок 13 – Концептуальная модель данных

С помощью данной диаграммы определены следующие сущности:

- Авторизованные пользователи (хранит данные обо всех авторизованных администраторах базы данных);
- Расписания (хранит данные о существующем расписании).

2.3.2 Обоснование вида логической модели веб-приложения

Существует несколько видов логических моделей данных. К ним относятся:

- иерархическая (древовидная);
- реляционная (сущности и связи);

- сетевая (сетевая структура).

Наиболее простой и в то же время информативной считается реляционная модель данных. Главным достоинством данного метода является наглядность представления структуры всех данных приложения.

2.3.3 Разработка логической модели данных веб-приложения

Логическая модель данных – это основа для создания будущей базы данных. Она определяет структуру и логические связи между таблицами. На рисунке 14 представлена логическая модель данных.

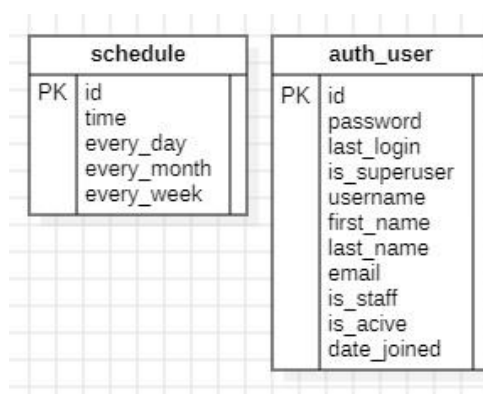


Рисунок 14 – Логическая модель данных

На данной диаграмме все сущностям присваивается уникальный идентификатор (id). Такие идентификаторы являются первичными ключами для сущностей. Также определены необходимые атрибуты для каждой сущности.

Сущность «schedule» создается для хранения информации о существующем расписании. Она необходима для вывода информации на экранную форму.

Сущность «auth_user» необходима для хранения информации об авторизованных администраторах БД.

2.4 Требования к аппаратно-программному обеспечению веб-приложения

«Резервное копирование является совокупностью технологий хранения и соединения систем хранения с вычислительными системами» [12]. В представленной ниже таблице 2 описаны компоненты оборудования и требования к ним для сервера управления приложением.

Таблица 2 – Требования к оборудованию

Компонент	Требование
Жесткий диск	Не менее 6 ГБ свободного пространства
Монитор	Минимальное разрешение 1024 на 768 px
Оперативная память	Не менее 4 ГБ
Процессор	x64 Минимальная частота 1,6 ГГц
Клавиатура и мышь	Необходимо наличие манипулятора «мышь» и клавиатуры.
Интернет	Для работы необходимо подключение к сети Интернет

Требования к программному обеспечению сервера управления приложением:

- операционная система Windows 10 TH1 1507, Windows Server 2016;
- минимальная платформа .NET Framework;
- Python 3;
- Pycharm.

Требования к серверу СУБД представлены в таблице 3.

Таблица 3 – Требования к серверу СУБД

Компонент	Требование
Жесткий диск	Не менее 6 ГБ свободного пространства
Монитор	Минимальное разрешение 800 на 600 px
Оперативная память	Не менее 4 ГБ

Продолжение таблицы 3

Процессор	x64 Минимальная частота 1,6 ГГц
Клавиатура и мышь	Необходимо наличие манипулятора «мышь» и клавиатуры.
Интернет	Для работы необходимо подключение к сети Интернет

Требования к программному обеспечению сервера СУБД:

– операционная система Windows 10 TH1 1507 (или позднее), Windows Server 2016 (или позднее);

- минимальная платформа .NET Framework;

Требования к оборудованию и программному обеспечению клиентского персонального компьютера:

- наличие любого современного браузера;

- наличие стабильного соединения с сетью Интернет;

- наличие «мыши» и клавиатуры;

- объем оперативной память – не менее 512 Мб;

- объем свободного пространства на жестком диске – не менее 350 Мб;

- разрешение экрана – не менее 800 на 600 пикселей.

Доменное имя должно соответствовать требованиям к доменам в зоне .ru.

Требования к безопасности:

- разграничение прав доступа для администратора базы данных и супер администратора;

- разграничение прав доступа к данным на уровне СУБД;

Выводы по главе 2. В процессе написания второй главы произведено логическое моделирование приложения и моделирование структуры данных. Были построены: диаграмма прецедентов, диаграммы последовательности, диаграммы «сущность-связь». В конце представлены технические требования к оборудованию сервера для управления приложением, СУБД и к персональному компьютеру для работы клиента в веб-браузере.

3 Физическое проектирование веб-приложения

3.1 Выбор архитектуры веб-приложения

Любая информационная система включает в себя такие компоненты:

- Данные (база данных приложения);
- Бизнес-логика (правила обработки данных);
- Представление (взаимодействие пользователя с данными).

Архитектура информационной системы – это структура информационной системы, которая определяет логику взаимодействия её компонентов.

Виды архитектуры ИС:

- файл-сервер (слои представления и бизнес-логики на ПК-клиенте, БД на сервере данных);
- клиент-сервер двухзвенная (слой представления и часть бизнес-логики на ПК-клиенте, часть бизнес-логики и БД на сервере данных);
- клиент-сервер трехзвенная (слой представления на ПК-клиенте, слой бизнес-логики на сервере приложения, БД на сервере данных) [2].

Ниже представлена сравнительная таблица (таблица 4) видов архитектуры информационных систем.

Таблица 4 – Сравнение видов архитектуры ИС

	Файл-сервер	Клиент-сервер двухзвенная	Клиент-сервер трехзвенная
Слой представления	Клиент	Клиент	Клиент
Бизнес-логика	Клиент	Часть на клиенте + Часть на сервере	Сервер приложения
Слой данных	Часть на клиенте + Часть на сервере	Сервер	Сервер БД

Продолжение таблицы 4

Недостатки	-низкая надежность; -низкая производительность; -низкая масштабируемость.	-сложность организации бизнес-логики; -сложность разработки; -сложность администрирования; -высокая стоимость. -высокие технические требования;	-сложность разработки; -сложность администрирования; -высокая стоимость.
Преимущества	-низкая стоимость; -высокая скорость разработки.	-масштабируемость выше, чем у файл-серверной архитектуры;	-минимум одновременно передаваемых данных; -высокая масштабируемость; -нагрузка на серверы ниже; -высокая масштабируемость; -выше надежность.

Также существует микросервисная архитектура, которая подразумевает разделение на части – микросервисы. Такая система удобна для создания отдельных модулей крупного приложения.

Вывод: если выбирать наиболее надежную и производительную систему, то следует остановиться на трехзвенной клиент-серверной архитектуре.

3.2 Выбор технологии для разработки веб-приложения

Для разработки информационной системы были выбраны следующие программные средства:

- Python 3 – высокоуровневый язык программирования [15];
- PyCharm Community Edition 2023.2 – интегрированная среда разработки на Python;
- Django – фреймворк Python для веб-разработки. Данная технология предлагает широкий набор возможностей для быстрого создания работоспособного и эстетичного приложения [14];

- Bootstrap 5.3 – css-фреймворк созданный для разработки сайтов и веб-приложений;
- Microsoft SQL Server 2022 – система управления базами данных SQL;
- SQL Server Management Studio Management Studio 19 – графическая среда для Microsoft SQL Server;
- Popper.js – плагин для создания всплывающих окон;
- Animate.css – библиотека анимаций;
- Jinja – «быстрый, выразительный и расширяемый шаблонизатор» [17].
- Font Awesome – библиотека шрифтов.

Выбор языка программирования Python обусловлен его простотой и лаконичностью. Также наличие большой библиотеки функций значительно упрощает разработку. Фреймворк Django является весьма удобным инструментом для разработки веб-приложений, он позволяет решить множество сложных задач. В данном инструменте реализован паттерн MVC – Model View Controller, который необходим для разделения логических частей приложения друг от друга на независимые блоки (модули). Отдельно стоит отметить такой удобный инструмент в Django как панель администратора, которая автоматически формируется при создании приложения и позволяет управлять учетными записями. Также в данную панель, при необходимости, можно добавлять дополнительный функционал.

3.3 Разработка физической модели данных веб-приложения

Физическая модель должна отображать всю необходимую информацию об объектах базы данных, она создается после логического моделирования и уточняет физические характеристики объектов.

В проекте используется единая база данных, содержащая таблицы. Данные таблицы переданы далее на рисунке 15.

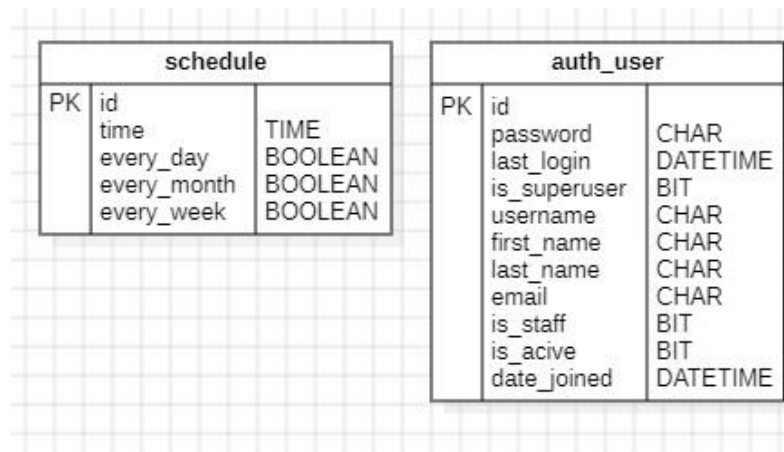


Рисунок 15 – Физическая модель данных приложения «WeBase»

В результате физического моделирования становится доступной реализация самой базы данных в системе.

Для генерации указанных таблиц в базе данных необходимо написать указанный на рисунке 16 код в файле models.py.

```
class schedule (models.Model):
    time_to_copy = models.DateField('Дата для создания')
    every_day = models.BooleanField('Каждый день')
    every_week = models.BooleanField('Каждую неделю')
    every_month = models.BooleanField('Каждый месяц')
```

Рисунок 16 – Код для создания таблицы в базе данных

После описания класса в файле models.py необходимо создать файл миграции и провести миграцию в БД. Нет необходимости самостоятельно писать SQL код, программа делает все за нас. Таблица «auth_user» создается автоматически при создании приложения Django вместе с административной панелью. Данная таблица будет использоваться для аутентификации администратора базы данных при входе в приложение.

3.4 Разработка веб-приложения

3.4.1 Схема взаимосвязи модулей приложения

Разрабатываемая информационная система состоит из модулей. Для более чёткого представления работы программы, необходимо описать, как модули взаимодействуют друг с другом. На рисунке 17 представлена структурная схема, наглядно представляющая алгоритм переходов между формами приложения.



Рисунок 17 – Схема взаимосвязи форм и модулей приложения

Приложение состоит из 4 связанных между собой форм:

- форма аутентификации (модуль аутентификации);
- форма главной страницы (доступ к основным модулям системы);
- форма «Создать или восстановить» (модуль резервного копирования);
- форма «Создать расписание» (модуль создания расписания для автоматизации резервного копирования).

Также существует модуль административной панели для управления учетными записями администраторов БД (стандартная панель Django).

При создании резервных копий формируются файлы формата json.

3.4.2 Описание модулей веб-приложения с примерами кода функций

«Программный модуль – программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения и взаимодействия с другими программными модулями, и загрузки в оперативную память» [3].

Модули разработанного приложения «WeBase», с кодами функций, представленными на рисунках ниже.

Модуль аутентификации администратора БД представлен на рисунке 18.

```
def login_user(request):  
    if request.method == "POST":  
        username = request.POST["username"]  
        password = request.POST["password"]  
        user = authenticate(request, username=username, password=password)  
        if user is not None:  
            login(request, user)  
            return redirect('home')  
        else:  
            messages.success(request, message="Попробуйте еще раз")  
            return redirect('login')  
    else:  
        return render(request, template_name='authenticate/login.html', context={})  
  
1 usage  
def logout_user(request):  
    logout(request)  
    messages.success(request, message="Вы вышли из системы")  
    return redirect('login')
```

Рисунок 18 – Код модуля аутентификации

В данном программном модуле реализованы такие функции как «login_user» и «logout_user», которые необходимы для входа и выхода из системы. Экранная форма данной функции представлена в приложении А.1, HTML-код страницы в приложении Б.3.

Модуль создания расписания изображен на рисунке 19.

```
def schedule1(request):
    if request.method == "POST":
        cho = request.POST["flexRadioDefault"]
        if cho == 'каждый день':
            scheduler.add_job(dump_base, trigger=CronTrigger
                              (hour="11", minute="00"),
                              max_instances=1,replace_existing=True)
            scheduler.start()
        elif cho == 'раз в неделю':
            scheduler.add_job(dump_base, trigger=CronTrigger
                              (day_of_week="mon", hour="11", minute="00"),
                              max_instances=1,replace_existing=True)
            scheduler.start()
        elif cho == 'раз в месяц':
            scheduler.add_job(dump_base, trigger=CronTrigger
                              (day_of_month="1", hour="11", minute="00"),
                              max_instances=1,replace_existing=True)
            scheduler.start()
        else:
            print("job is not defined")
    return render(request,
                  template_name: 'main/Shedule.html',
                  context: {"key":cho})
```

Рисунок 19 – Код модуля создания расписания

В данном программном модуле реализована функция «schedule1». Она позволяет создавать необходимое расписание для выполнения резервного копирования. Администратор БД может выбрать частоту создания резервной

копии (каждый день, раз в неделю, раз в месяц). Экранная форма для представления данного программного модуля находится в приложении А.4.

Модуль резервного копирования представлен на рисунке 20.

```
def dump_base(request):

    name = ('C:\Backup\pythonProject\Base\dump_{}.json'
            .format(datetime.now().strftime("%d%m%Y_%H%M%S")))
    management.call_command( command_name: "dumpdata",
                             *args: "-o", name)
    my_file = Path("C:\Backup\pythonProject\Base\dump_{}.json"
                  .format(datetime.now().strftime("%d%m%Y_%H%M%S")))
    if my_file.is_file():
        messages.add_message(request, messages.INFO,
                             message: 'Копия базы данных успешно создана')
    else:
        messages.add_message(request, messages.INFO,
                             message: ' ')
    return render(request,
                  template_name: 'main/copy.html')

1 usage
def load_base(request):
    if request.method == 'POST':
        print(request.FILES.get('file'))
        file1 = request.FILES.get('file')
    management.call_command( command_name: "loaddata",
                             *args: file1)
    return render(request,
                  template_name: 'main/copy.html')
```

Рисунок 20 – Код модуля резервного копирования

В данном модуле реализованы функции: «dump_base» - для создания резервной копии в файле формата json. Название файла формируется в соответствии с текущей датой и временем. Функция «load_base» - для загрузки из резервной копии в БД. Экранная форма данной функции представлена в приложении А.3, HTML-код страницы в приложении Б.2.

3.5 Описание функциональности веб-приложения

Работа в приложении начинается с окна аутентификации, в котором администратор БД вводит свои логин и пароль, которые до этого были выданы ему супер администратором. После успешной аутентификации открывается главная страница приложения (экранная форма представлена в приложении А.2, HTML код страницы - Б.1 и Б.4), на которой расположена панель навигации. Данная панель доступна на всех страницах приложения. На ней расположены ссылки на основные страницы приложения:

- «Создать или восстановить». На данной странице пользователь имеет возможность создать резервную копию существующей базы данных и восстановить базу из резервной копии;

- «Создать расписание». Данный модуль дает возможность создавать копии с заданной периодичностью. Возможно выбрать автоматическое создание копии каждый день, один раз в неделю и один раз в месяц;

- административная панель. Стандартный модуль веб-фреймворка Django, необходимый для управления учетными записями. Доступ к данному модулю имеет только супер администратор.

Также под списком страниц находится кнопка «Выход», которая позволяет выйти из системы.

3.6 Тестирование разработанного веб-приложения

3.6.1 Выбор метода тестирования веб-приложения

«На высоком уровне тестирование ПО является способом предоставления оценки качества программного обеспечения заинтересованным лицам» [5].

Объект тестирования – разработанное веб-приложение для взаимодействия с базой данных и резервного копирования.

Цель тестирования — проверка соответствия разработанного продукта предъявляемым требованиям.

Существует несколько классификаций методов тестирования ПО. Рассмотрим некоторые из них.

Классификация по доступу к архитектуре:

- тестирование черного ящика (без доступа к коду программы);
- тестирование белого ящика (основано на знании кода программы);
- тестирование серого ящика (гибрид методов черного и белого ящика).

Классификация по степени автоматизации:

- автоматизированное;
- ручное.

Классификация по уровню детализации:

- модульное (тестирование отдельного модуля);
- системное (тестирование системы и оценка ее качества);
- интеграционное (тестирование взаимодействия модулей).

Так как приложение небольшое, проще всего проверить его работоспособность ручным способом, чем писать отдельные тесты. Поэтому был выбран метод ручного тестирования.

3.6.2 Описание процесса тестирования веб-приложения

Для начала был проведен смоук-тест (smoke test), который показал, что приложение успешно запускается и готово для дальнейшего ручного тестирования функционала [7].

Чтобы успешно протестировать работу веб-приложения, необходимо проверить как оно работает в различных браузерах. Проверка была проведена в таких браузерах: Google Chrome, Yandex, Mozilla Firefox. Во всех данных браузерах приложение работает идентично.

Для проверки работоспособности функционала веб-приложения, был разработан тест-план (таблица 5) для следующих функций приложения [8]:

- процесс аутентификации администратора базы данных;

- процесс резервного копирования;
- процесс восстановления из резервной копии.

Таблица 5 – Тест-план тестирования

№ теста	Операция	Описание операции
1	Аутентификация администратора БД	- Ввод логина; - Ввод пароля; - Нажатие кнопки «Войти»
2	Создание резервной копии	- Переход на страницу «Резервное копирование»; - Нажатие на кнопку «Создать»;
3	Восстановление из резервной копии	- Переход на страницу «Резервное копирование»; - Выбрать файл резервной копии из существующих в файловой системе; - Нажатие на кнопку «Восстановить»; - Проверка восстановления базы в Microsoft SQL Server Management Studio.

Следуя тест-плану, было проведено ручное тестирование в трех описанных выше браузерах. Результаты предъявлены в таблице 6.

Таблица 6 – Итоги тестирования разработанного приложения

№ теста	Описание операции	Результат тестирования
1	Аутентификация администратора БД	Успешно
2	Создание резервной копии	Успешно
3	Восстановление из резервной копии	Успешно

В результате тестирования можно сделать вывод, что выбранные для тестирования модули системы работают правильно, все полученные результаты соответствуют ожиданиям.

Выводы по главе 3. В третьей главе представленной выпускной квалификационной работы произведено физическое проектирование разрабатываемой информационной системы для использования функций резервного копирования базы данных MS SQL.

В процессе оформления 3 главы квалификационной работы были определены: архитектура создаваемой информационной системы, технологии для успешной разработки функционального приложения. Далее была сформулирована физическая модель данных и описана база данных для приложения. Далее была произведена разработка необходимых модулей системы и их описание. Также приведены программные коды созданных программных функций. В заключении произведено успешное тестирование нескольких модулей разработанного веб-приложения.

Заключение

Данная выпускная квалификационная работа посвящена такой весьма актуальной проблеме как резервное копирование базы данных.

В рамках выполнения выпускной квалификационной работы был осуществлен детальный анализ предметной области резервного копирования, проведено сравнение существующих программных продуктов. Далее было выполнено поэтапное проектирование создаваемой информационной системы. Данный трудоёмкий процесс стандартно включает в себя логическое и физическое моделирование данных и процессов системы. После проектирования была осуществлена разработка информационной системы «WeBase» для резервного копирования базы данных MS SQL. По завершении было проведено успешное ручное тестирование нескольких модулей разработанной системы.

Вследствие выполнения выпускной квалификационной работы были решены все поставленные в начале задачи. Цели исследования, заключающиеся в изучении методов резервного копирования и разработке информационной системы, были полностью достигнуты.

В конечном результате разработано функциональное и гармоничное, как в плане эстетики, так и в плане работоспособности, приложение «WeBase». Оно служит для взаимодействия с базой данных MS SQL и её резервного копирования. Созданная система реализует следующие функции:

- аутентификация зарегистрированного администратора базы данных;
- резервное копирование и восстановление базы данных;
- резервное копирование по расписанию.

В дальнейшем, безусловно, необходимо совершенствовать разработанную программу, добавлять новые полезные функции и модули, дорабатывать существующие.

В результате выполнения данной работы получен ценный практический опыт в проектировании и разработке информационных систем.

Список используемой литературы

1. Бережной А. Н. Сохранение данных: теория и практика [Текст] / А. Н. Бережной. — Москва: ДМК Пресс, 2016 — 317 с.
2. Вичугова А. Трехслойная и трехзвенная: введение в архитектуру ИС для аналитика / Вичугова А. [Электронный ресурс]. URL: <https://babok-school.ru/blogs/is-architecture-basics-for-junior-analysts/> (дата обращения: 08.09.2023).
3. ГОСТ Р 53195.3-2009. Безопасность функциональная связанных с безопасностью зданий и сооружений систем. Часть 3. Требования к системам : национальный стандарт Российской Федерации : дата введения 2010-06-01 / Федеральное агентство по техническому регулированию. – Изд. официальное. – Москва : Стандартиформ, 2010. – 60 с.
4. Иванов Н. А. Средства резервного копирования и восстановления данных в операционных системах Windows и Linux : методические указания к проведению практических занятий по направлению подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Системотехника и автоматизация проектирования и управления в строительстве» очной и заочной форм обучения [Текст] / Н. А. Иванов. — Москва: Московский государственный строительный университет, 2015 — 40 с.
5. Кондрашов Ю. Н. Эффективное использование СУБД MS SQL Server: учебное пособие [Текст] / Ю. Н. Кондрашов. — Москва: РУСАЙНС, 2024 — 122 с.
6. Коннолли Т. М., Бегг К. Е. Базы данных: проектирование, реализация и сопровождение. Теория и практика [Текст] / Т. М. Коннолли К. Е. Бегг — 3-е издание. — Москва: Вильямс, 2017 — 1440 с.
7. Лабун Б. Дружеское знакомство с тестированием программ [Текст] / Лабун Б. — Санкт-Петербург: БХВ-Петербург, 2022 — 288 с.: ил.

8. Лекция по разработке ТРПО. Методы тестирования / [Электронный ресурс]. URL: <https://studfile.net/preview/9597391/page:19/> (дата обращения: 17.09.2023).

9. Преимущества Handy Backup / [Электронный ресурс]. URL: <https://www.handybackup.ru/> (дата обращения: 09.09.2023).

10. Резервное копирование 101: определение, виды и схемы бэкапов / [Электронный ресурс]. URL: <https://www.ispsystem.ru/news/backup-101> (дата обращения: 09.09.2023).

11. Рябов А. Обзор Vembu BDR Suite 3.8 / Рябов А. [Электронный ресурс]. URL: <https://www.anti-malware.ru/reviews/vembu-bdr-suite-38> (дата обращения: 10.09.2023).

12. Савватеев М.Е. Анализ технологий резервного копирования данных / Савватеев М.Е. [Электронный ресурс]. URL: <https://scienceforum.ru/2019/article/2018011550> (дата обращения: 12.09.2023).

13. Differential Backup vs. Incremental Backup / [Электронный ресурс]. URL: <https://www.nakivo.com/blog/differential-backup-vs-incremental-backup/> (дата обращения: 10.09.2023).

14. Django documentation / [Электронный ресурс]. URL: <https://docs.djangoproject.com/en/4.2/> (дата обращения: 20.09.2023).

15. Documentation Python 3.11.5 / [Электронный ресурс]. URL: <https://docs.python.org/3/> (дата обращения: 21.09.2023).

16. Incremental Jobs / [Электронный ресурс]. URL: <https://helpcenter.nakivo.com/User-Guide/Content/Overview/Incremental-Jobs.html> (дата обращения: 10.09.2023).

17. Jinja documentation / [Электронный ресурс]. URL: <https://jinja.palletsprojects.com/en/latest/> (дата обращения: 22.09.2023).

18. Top 9 Best SQL Server Backup Software in 2023 / [Электронный ресурс]. URL: <https://www.easeus.com/todo-backup-guide/sql-server-backup-software.html> (дата обращения: 08.09.2023).

19. What is a Full Backup? / [Электронный ресурс]. URL: <https://www.backup4all.com/full-backup-kb.html> (дата обращения: 10.09.2023).

20. What Is Incremental Backup? / [Электронный ресурс]. URL: <https://www.nakivo.com/blog/what-is-incremental-backup> (дата обращения: 10.09.2023).

21. What Is Synthetic Full Backup: Facts Every Sysadmin Should Know / [Электронный ресурс]. URL: <https://www.nakivo.com/blog/what-is-synthetic-backup/> (дата обращения: 10.09.2023).

22. What is Structured Query Language (SQL)? / [Электронный ресурс]. URL: <https://www.techtarget.com/searchdatamanagement/definition/SQL> (дата обращения: 15.09.2023).

23. What is Unified Modeling Language (UML)? / [Электронный ресурс]. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> (дата обращения: 12.09.2023).

Приложение А

Скриншоты экранных форм разработанного веб-приложения «WeBase»

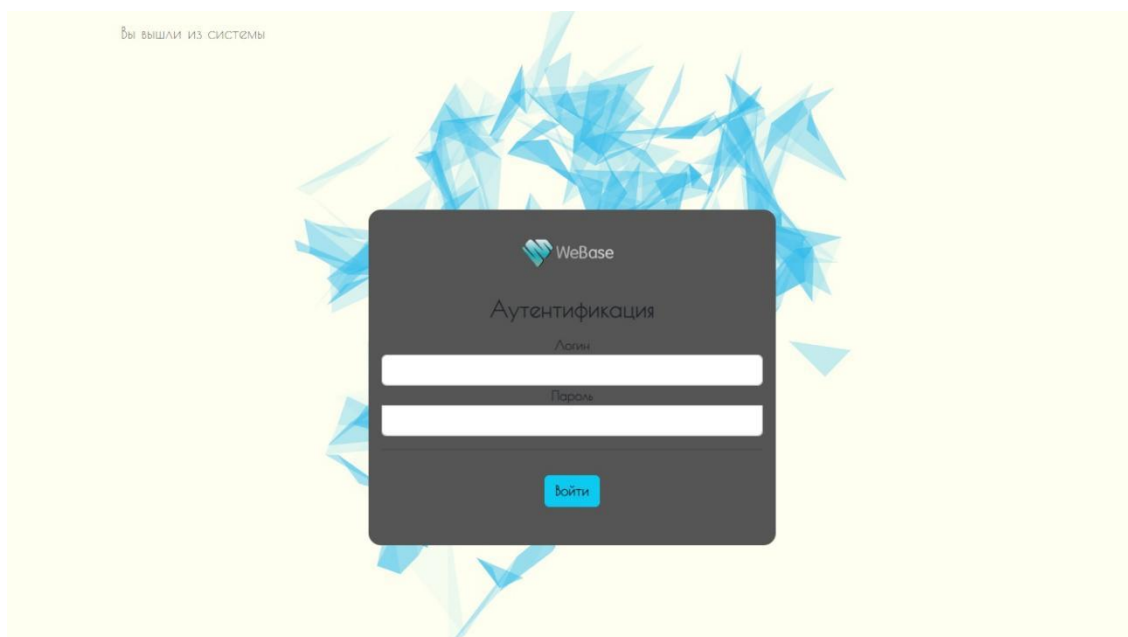


Рисунок А.1 – Скриншот страницы аутентификации

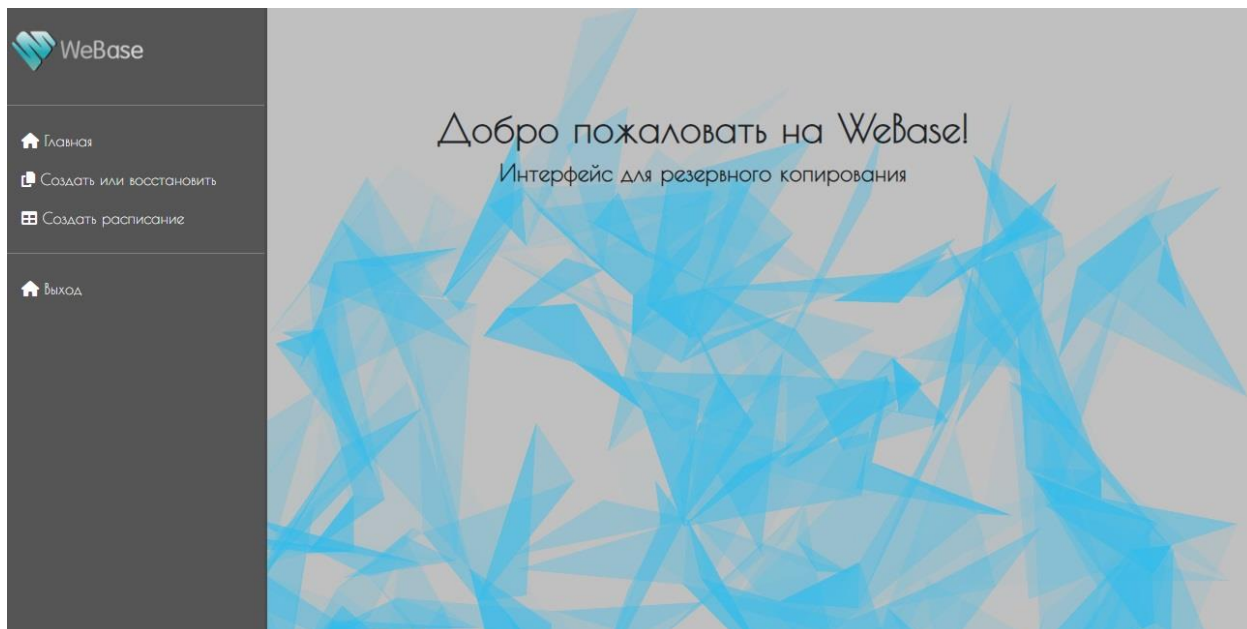


Рисунок А.2 – Скриншот главной страницы с приветствием и боковым навигационным меню

Продолжение Приложения А

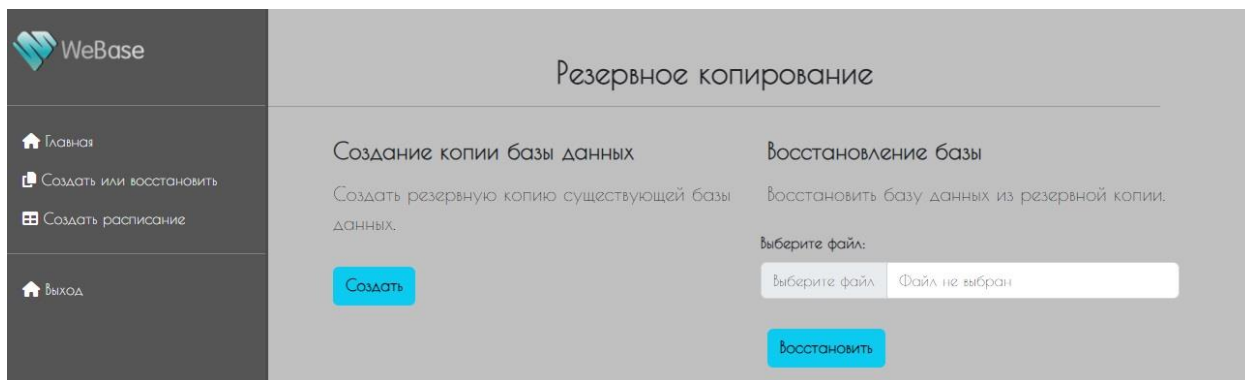


Рисунок А.3 – Скриншот страницы «Резервное копирование»

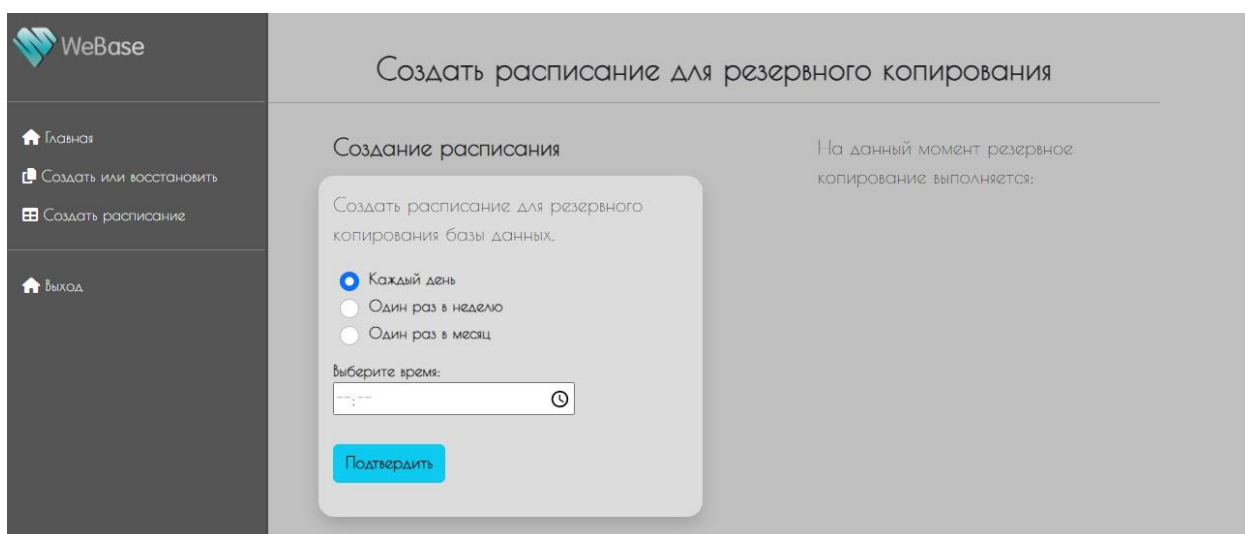


Рисунок А.4 – Скриншот страницы «Создать расписание для резервного копирования»

Приложение Б

Листинги кода страниц HTML

Листинг Б.1 – Страница «Главная»

```
<body>
<div class="wrapper">
<aside>
<div class="d-flex flex-column" >
  
  <hr>
  <ul class="nav nav-pills flex-column">
    <li>
      <a href="{% url 'home' %}" class="nav-link">
        <i class="fa-solid fa-house"></i>
        Главная
      </a>
    </li>
    <li>
      <a href="{% url 'copy' %}" class="nav-link">
        <i class="fa-solid fa-copy"></i>
        Создать или восстановить
      </a>
    </li>
    <li>
      <a href="{% url 'schedule1' %}" class="nav-link">
        <i class="fa-solid fa-table"></i>
        Создать расписание
      </a>
    </li>
  </ul>
  <hr>
  {% if user.is_authenticated %}
  <ul class="nav nav-pills flex-column">
    <li>
      <a href="{% url 'logout' %}" class="nav-link">
        <i class="fa-solid fa-house"></i>
        Выход
      </a>
    </li>
  </ul>
  {% endif %}
  </div>
</aside>
<main>
  {% block content %}
  {% endblock %}
</main>
</div>
</body>
```

Продолжение Приложения Б

Листинг Б.2 – Страница «Создать или восстановить»

```
{% extends 'main/layout.html' %}

{% block title %}Копирование{% endblock %}

{% block content %}
<div class="container px-4 py-5">
<h2 class="animate__animated animate__pulse">Резервное копирование</h2>
<hr>
<div class="row">
  <div class="col d-flex align-items-start">
    <div class="icon-square text-body-emphasis bg-body-secondary d-inline-flex align-items-center justify-content-center fs-4 flex-shrink-0 me-3">
      <svg class="bi" width="1em" height="1em"></svg>
    </div>
    <div class="col-6">
      <h3 class="fs-4 text-body-emphasis">Создание копии базы данных</h3>
      <p>Создать резервную копию существующей базы данных.</p>
      <a href="{% url 'dump' %}" class="btn btn-info">Создать</a>
      <div class="container">
        {% if messages %}
          {% for message in messages %}
            {{ message }}
          {% endfor %}
        {% endif %}
      </div>
    </div>
  </div>
  <div class="col-6">
    <h3 class="fs-4 text-body-emphasis">Восстановление базы</h3>
    <p>Восстановить базу данных из резервной копии.</p>
    <form method="POST" action="{% url 'load' %}" enctype="multipart/form-data">
      {% csrf_token %}
      <div class="mb-2 p-2">
        <label for="formFileMultiple" class="form-label">Выберите файл:</label>
        <input name="file" class="form-control" type="file" id="formFileMultiple">
      </div>
      <button type="submit" class="btn btn-info">Восстановить </button>
    </form>
  </div>
</div>
</div>
</div>
<script>
$(document).ready(function(){
  PopUpHide();
});
function PopUpShow(){
  $("#popup1").show();
}
function PopUpHide(){
  $("#popup1").hide();
}
</script>
</div>
{% endblock %}
```


Продолжение Приложения Б
Листинг Б.3 – Страница «Аутентификация»

```
<body>
<div class="container">
  {% if messages %}
    {% for message in messages %}
      {{ message }}
    {% endfor %}
  {% endif %}
</div>
<main class="form-signin">
<form action="" method="post">
  {% csrf_token %}
  
  <h3>Аутентификация</h3>
  <div class="form-group">
    <label >Логин</label>
    <input type="text" class="form-control" name="username">
  </div>
  <div class="form-group">
    <label >Пароль</label>
    <input type="password" class="form-control" name="password">
  </div>
  <hr>
  <button class="btn btn-info " type="submit">Войти</button>
</form>
</main>
<script src="/docs/5.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-HwwvtgBNo3bZJLYd8oVXjrBZt8cqvSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
  crossorigin="anonymous"></script>
</body>
```

Продолжение Приложения Б

Листинг Б.4 – Страница «Главная-блок контент»

```
{% extends 'main/layout.html' %}
{% load static %}
{% block title %}Главная страница{% endblock %}

{% block content %}

<div class="cont">
<h1>Добро пожаловать на WeBase!</h1>
    <h4>Интерфейс для резервного копирования</h4>

</div>

<style>
    body {
        background-image: url('{% static 'main/img/fon2.png' %}');
        background-repeat: no-repeat;
        background-size: cover;
    }
</style>
{% endblock %}
```