СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 2](#_Toc516505548)

[1. АНАЛИЗ СРЕДСТВ обработки и хранения МЕТЕОРОЛОГИЧЕСКИХ ДАННЫХ 4](#_Toc516505549)

[1.1. Полевые исследования свойств почв 4](#_Toc516505550)

[1.2. Лизиметрическая станция 5](#_Toc516505551)

[1.3. Краткий обзор существующих приложений по данной тематике 7](#_Toc516505552)

[2. РАЗРАБОТКА БАЗЫ ДАННЫХ И WEB-ПРИЛОЖЕНИЯ 10](#_Toc516505553)

[2.1 Модель базы данных 10](#_Toc516505554)

[2.2. Разработка web-приложения 18](#_Toc516505555)

[3. ИНТЕРФЕЙС И ФУНКЦИОНАЛ 24](#_Toc516505556)

[3.1 Стартовая страница 24](#_Toc516505557)

[3.2 Страница импорта данных 28](#_Toc516505558)

[3.3 Страница ручного ввода данных 29](#_Toc516505559)

[3.4 Страница табличного вывода данных 30](#_Toc516505560)

[3.5 Страница графического вывода данных 31](#_Toc516505561)

[ЗАКЛЮЧЕНИЕ 32](#_Toc516505562)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 33](#_Toc516505563)

[Приложение 1. SQL-код базы данных 34](#_Toc516505564)

[Приложение 2. Скрипт module.php 38](#_Toc516505565)

[Приложение 3. Скрипт conf.php 46](#_Toc516505566)

[Приложение 4. Скрипт script.js 47](#_Toc516505567)

# ВВЕДЕНИЕ

Сегодня информационные технологии развиваются очень стремительно и проникают во все сферы человеческой деятельности. Для принятия обоснованных и эффективных решений в производственной и научной деятельности современный специалист должен уметь с помощью компьютеров и средств связи получать, накапливать, хранить и обрабатывать данные, представляя результат в виде наглядных отчетов.

При получении данных с метеорологических станций, поступают большие массивы данных, для упрощения исследования которых, необходимо специализированное программное обеспечение. Однако, на современном рынке представлено не так много программных средств, удовлетворяющих нуждам пользователя, поэтому необходимо разработать актуальное приложение, которое позволит облегчить работу научному сотруднику.

Целью данной дипломной работы является разработка удаленной базы данных и Web-приложения для сбора, предварительной обработки и хранения данных, получаемых с почвенной лизиметрической станции, гидрометеорологической станции и стационарных станций измерения влажности почв в неупорядоченном виде.

Для достижения поставленной цели необходимо решить следующие задачи:

* проведения анализа существующих приложений;
* проектирование и реализация архитектуры базы данных и приложения;
* проектирование и реализация пользовательского интерфейса;
* тестирование разработанного приложения.

Для управления базой данных была выбрана СУБД с открытым исходным кодом phpMyAdmin разработанная phpMyAdmin Developer Team. Для разработки клиентской части были использованы:

* стандартизированный язык разметки документов в сети Internet – HTML;
* формальный язык описания внешнего вида документа, написанного с использованием HTML – CSS;
* скриптовый язык общего назначения, применяемый для разработки Web-приложений – PHP;
* мультипарадигменный язык программирования – JavaScript.

Разработанные в ходе дипломной работы база данных и приложение позволяют увеличить скорость поиска и доступа пользователя к необходимым метеорологическим данным, упорядочить и систематизировать их.

# АНАЛИЗ СРЕДСТВ обработки и хранения МЕТЕОРОЛОГИЧЕСКИХ ДАННЫХ

## **1.1. Полевые исследования свойств почв**

Полевое описание почвенного профиля, исследование структуры, гранулометрического состава, цвета и других морфологических свойств почв в настоящее время сопровождаются прямыми количественными исследованиями. В полевых условиях изучаются, кроме морфологических, разнообразные физические свойства, которые необходимы для проведения расчетов.

Физические свойства и процессы, протекающие в почве, во многом определяют направленность почвообразовательного процесса, условия для роста и развития растений. Наиболее тесный контакт физика почв имеет с земледелием и мелиорацией, задачей которых является временное или коренное улучшение, главным образом, физических свойств почвы для практических целей. Физические свойства учитываются при разработке агротехнических приемов по зонам, а также должны быть положены в основу мелиоративных мероприятий. Так, для зон недостаточного увлажнения разрабатываются приемы улучшения физических свойств почвы, способствующие накоплению и сохранению воды. Наоборот, в зоне избыточного увлажнения агротехнические и мелиоративные мероприятия должны быть направлены в сторону уменьшения содержания воды в почве и увеличения аэрации ее, а для северных районов нужны также приемы тепловых мелиораций.

Оптимальными физическими свойствами и режимами (водным, воздушным, тепловым) будут такие, которые обеспечивают максимальный урожай растений при полной обеспеченности почвы элементами питания.

Знание физических свойств почв и грунтов важно при оценке их как строительного фундамента, так и санитарного состояния.

В настоящее время изучению физических свойств почвы уделяется большое внимание как в стационарных условиях, так и в экспедиционных. Полевые исследования дополняются лабораторными [1].

## **1.2. Лизиметрическая станция**

Недостаток воды является ограничивающим фактором для роста растительности и для развития устойчивой стратегии предотвращения пыльных бурь.

Поэтому точная информация о водном балансе почв необходима для оценки системы управления степи. Только лизиметры позволяют точно измерить баланс воды в почве. Лизиметр ‐ это устройство для сбора дренажных вод массового и растворенного веществ в почве, материнской породе, растительности по отношению к местному климату и другим условиям местности.

Лизиметр состоит из заполненных почвой металлических цилиндров, для собирания и учёта просочившейся через почву воды. Только лизиметры позволяют определение количества воды, просачивающуюся через почву, и типа и количества растворенных в ней питательных веществ напрямую. Таким образом, они более надежны при расчете нагрузки растворенных веществ, транспортируемых в грунтовую воду, чем какие‐либо другие методы.

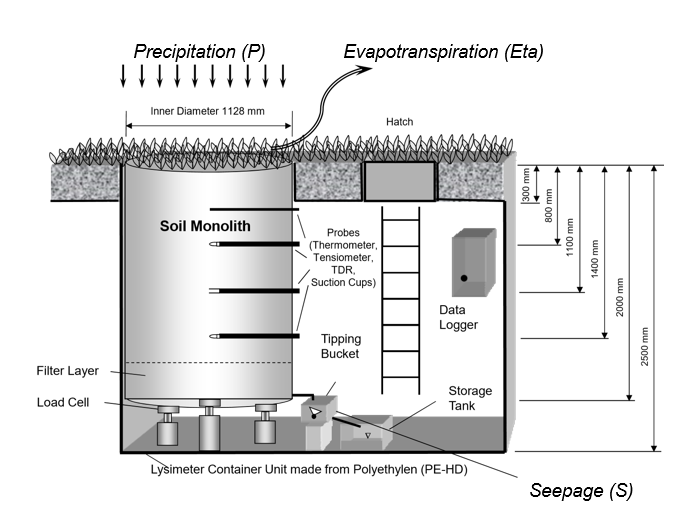


Рисунок 1.1 – Схема контейнерной лизиметрической станции

Если устройство позволяет взвешивание лизиметра, то возможно произвести расчет фактического суммарного испарения с помощью изменения веса. За счет этих свойств лизиметр является отличным инструментом для построения или калибровки моделей транспорта растворенных веществ. Технические свойства лизиметра (определенная площадь и длина) в основном зависит от научного вопроса, характера наполнения (нарушенное или не нарушенное), нижней границы и расположения установки. Небольшие недостатки местности компенсируется большой площадью основания лизиметра. Кроме того, лизиметр с растительным покровом представляет собой естественный инвентарь сельскохозяйственных культур и может принимать во внимание максимальную глубину проникновения корня. Чтобы обеспечить в цилиндрах лизиметра условия сходные с естественными условиями в почве, рекомендуется их заполнять монолитно. Большой взвешиваемый лизиметр является лучшим устройством для получения достоверных данных о фильтрации, количестве и качестве воды. Однако это подразумевает значительное инвестирование и дополнительные расходы на содержание [2].

## **1.3. Краткий обзор существующих приложений по данной тематике**

На данный момент на рынке представлено несколько информационных систем со схожими функциями и параметрами:

* «АСК»;
* GeoMixer.

Программа «ACK» предназначена для организации сбора метеоданных от метеокомплексов МК-26. Программа выполняет следующие функции:

* приём данных по каналу связи по запросу с заданным интервалом времени;
* запуск программы визуализации INFO.EXE для каждого метеокомплекса;
* сохранение принятых данных в формате CSV (текстовый файл с разделителем ';') в месячном архиве на жестком диске;
* ведение протокола работы программы и запись его на жёсткий диск;
* формирование таблиц измерений, печать таблиц, экспорт данных на диск в формате MS Word, HTML, XML;
* формирование графиков измерений, масштабирование графиков, печать графиков, экспорт графиков на диск в формате JPEG, SVG, EMF;
* формирование выборок данных из месячных архивов для дальнейшей обработки в программе Microsoft Excel или просмотре в таблице или на графике;
* передача данных в формате CSV (текстовый файл с разделителем ';') на WEB-сервер по FTP-протоколу для отображения на WEB-странице. Параметры настройки для доступа к FTP-серверу описываются в файле FTP.INI;
* передача данных в формате BUFR в ЦКС по FTP-протоколу для передачи по АСПД и по локальной сети для обработки и архивирования [3].

Основными минусами системы «АСК» являются узкая направленность, т.е. работа только с метеокомплексами МК-26, а также отсутствие данного программного обеспечения в свободном доступе.

Scanex Web-GIS GeoMixer — это веб-картографическая интеграционная платформа, которая помогает создавать и внедрять ГИС во внутренние системы и базы данных предприятий.

Возможности платформы GeoMixer:

* + поддержка основных ГИС-форматов;
  + геокодинг из текстовых файлов по координатам или адресам;
  + возможность интеграции дополнительных данных (фото, видео, гипертекст);
  + внедрение подключаемых сервисов (карты, снимки, адресный поиск, публичная кадастровая карта (ПКК), метаданные космосъемки);
  + визуализация пространственных и атрибутивных данных;
  + поиск объектов, пространственные и атрибутивные запросы;
  + редактирование векторных объектов;
  + работа с набором разновременных данных (мультивременные слои);
  + экспорт данных;
  + наличие системы прав доступа к проектам, многопользовательский доступ с разграничением прав пользователей;
  + наличие клиентского и серверного API, инструментария для разработчиков приложений и сервисов;
  + возможность интеграции с информационными системами на базе СУБД: MS SQL Server или Postgre SQL;
  + поддержка мобильных устройств [4].

Главным минусом данной платформы, также, является отсутствие данного программного обеспечения в свободном доступе.

Исходя из узкой направленности тематики, и отсутствия нужных программных средств в свободном доступе возникает необходимость разработки собственного программного обеспечения, для осуществления структурирования и первичной обработки специфических метеорологических данных.

# 2. РАЗРАБОТКА БАЗЫ ДАННЫХ И WEB-ПРИЛОЖЕНИЯ

## **2.1. Модель базы данных**

Для хранения и работы с данными была выбрана СУБД с открытым исходным кодом phpMyAdmin. phpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

На сегодняшний день phpMyAdmin широко применяется на практике. Последнее связано с тем, что разработчики интенсивно развивают свой продукт, учитывая все нововведения СУБД MySQL [5].

База данных состоит из 3 сущностей: Hidromet, Stational, Lisemetric, представляющих основные единицы информации в web-приложении (Рисунок 2.1).



Рисунок 2.1 - архитектура базы данных

Выделены следующие сущности:

1. «Hidromet» («No», «Time», «mm», «deg», «msdo», «Cdo», «Prdo», «hPado», «mmcharp», «WmI», «HKBat», «HKTemp», «HLrH») – эта сущность отводится для хранения показателей, регистрируемых гидрометеостанцией.

Значения атрибутов:

1. «No» - номер точки;
2. «Time» - время измерения;
3. «mm» - количество атмосферных осадков в капельном виде;
4. «deg» - направление ветра;
5. «msdo» - скорость ветра;
6. «Cdo» - температура воздуха;
7. «Prdo» - относительная влажность воздуха;
8. «hPado» - атмосферное давление;
9. «mmcharp» - количество атмосферных осадков в капельном виде;
10. «WmI» - интенсивность солнечной радиации;
11. «HKBat» - напряжение батареи питания;
12. «HKTemp» - температура батареи питания;
13. «HLrH» - доля свободной памяти логгера.
14. «Stational» («Time», «CThirdty», «procdoThirdty», «SmdoThirdty», «CThirdtyF», «pFdoThirdty», «CSixty», «procdoSixty», «SmdoSixty», «CSixtyF», «pFdoSixty», «CHThirdty», «doHThirdty», «SmdoHThirdty», «CHThirdtyF», «pFdoHThirdty», «HKBatV», «HKTempC», «HKrH») – эта сущность отводится для хранения показателей, регистрируемых стационарными станциями измерения влажности почв.

Значения атрибутов:

1. «Time» - время измерения;
2. «CThirdty» - температура на глубине 30 см;
3. «procdoThirdty» - объемная влажность почвы на глубине 30 см;
4. «SmdoThirdty» - проводимость на глубине 30 см;
5. «CThirdtyF» - температура на глубине 30 см;
6. «pFdoThirdty» - потенциал почвенной влаги на глубине 30 см;
7. «CSixty» - температура на глубине 60 см;
8. «procdoSixty» - объемная влажность почвы на глубине 60 см;
9. «SmdoSixty» - проводимость на глубине 60 см;
10. «CSixtyF» - температура на глубине 60 см;
11. «pFdoSixty» - потенциал почвенной влаги на глубине 60 см;
12. «CHThirdty» - температура на глубине 130 см;
13. «doHThirdty» - объемная влажность почвы на глубине 130 см;
14. «SmdoHThirdty» - проводимость на глубине 130 см;
15. «CHThirdtyF» - температура на глубине 130 см;
16. «pFdoHThirdty» - потенциал почвенной влаги на глубине 130 см;
17. «HKBatV» - напряжение батареи питания;
18. «HKTempC» - температура батареи питания;
19. «HKrH» - доля свободной памяти логгера.
20. «Lisemetric» («Date», «Time», «Tens30.1», «Tens30.1max», «Tens30.1min», «Tens30.2», «Tens30.2max», «Tens30.2min», «Vacuum30», «Vacuum30max», «Vacuum30min», «Tens50.1», «Tens50.1max», «Tens50.1min», «Tens50.2», «Tens50.2max», «Tens50.2min»,»Vacuum50», «Vacuum50max», «Vacuum50min», «Tens120.1», «Tens120.1max», «Tens120.1min», «Tens120.2», «Tens120.2max», «Tens120.2min», «Vacuum120», «Vacuum120max», «Vacuum120min», «Accu», «Accumax», «Accumin», «UMP30.1», «UMP30.2», «UMP50.1», «UMP50.2», «UMP120.1», «UMP120.2», «EC30.1», «EC30.2», «EC50.1», «EC50.2», «EC120.1», «EC120.2», «Temp30.1», «Temp30.2», «Temp50.1», «Temp50.2», «Temp120.1», «Temp120.2», «Weight1», «Weight2»,»Drain1», «Drain1max», «Drain1min», «Drain2», «Drain2max», «Drain2min») – эта сущность отводится для хранения показателей, регистрируемых почвенной лизиметрической станцией.

Значения атрибутов:

1. «Date» - дата измерений;
2. «Time» - время измерений;
3. «Tens30.1» - осмотическое давление почвенной влаги на глубине 30см в первом монолите (среднее значение);
4. «Tens30.1max» - осмотическое давление почвенной влаги на глубине 30см в первом монолите (максимальное значение за интервал измерений);
5. «Tens30.1min» - осмотическое давление почвенной влаги на глубине 30см в первом монолите (минимальное значение за интервал измерений);
6. «Tens30.2» - осмотическое давление почвенной влаги на глубине 30см в втором монолите (среднее значение);
7. «Tens30.2max» - осмотическое давление почвенной влаги на глубине 30см в втором монолите (максимальное значение за интервал измерений);
8. «Tens30.2min» - осмотическое давление почвенной влаги на глубине 30см в втором монолите (минимальное значение за интервал измерений);
9. «Vacuum30» - понижение давления в системе отбора проб воды с глубины 30см (среднее значение);
10. «Vacuum30max» - понижение давления в системе отбора проб воды с глубины 30см (максимальное значение за интервал измерений);
11. «Vacuum30min» - понижение давления в системе отбора проб воды с глубины 30см (минимальное значение за интервал измерений);
12. «Tens50.1» - осмотическое давление почвенной влаги на глубине 50см в первом монолите (среднее значение);
13. «Tens50.1max» - осмотическое давление почвенной влаги на глубине 50см в первом монолите (максимальное значение за интервал измерений);
14. «Tens50.1min» - осмотическое давление почвенной влаги на глубине 50см в первом монолите (минимальное значение за интервал измерений);
15. «Tens50.2» - осмотическое давление почвенной влаги на глубине 50см в втором монолите (среднее значение);
16. «Tens50.2max» - осмотическое давление почвенной влаги на глубине 50см в втором монолите (максимальное значение за интервал измерений);
17. «Tens50.2min» - осмотическое давление почвенной влаги на глубине 50см в втором монолите (минимальное значение за интервал измерений);
18. «Vacuum50» - понижение давления в системе отбора проб воды с глубины 50см (среднее значение);
19. «Vacuum50max» - понижение давления в системе отбора проб воды с глубины 50см (максимальное значение);
20. «Vacuum50min» - понижение давления в системе отбора проб воды с глубины 50см (минимальное значение);
21. «Tens120.1» - осмотическое давление почвенной влаги на глубине 120см в первом монолите (среднее значение);
22. «Tens120.1max» - осмотическое давление почвенной влаги на глубине 120см в первом монолите (максимальное значение за интервал измерений);
23. «Tens120.1min» - осмотическое давление почвенной влаги на глубине 120см в первом монолите (минимальное значение за интервал измерений);
24. «Tens120.2» - осмотическое давление почвенной влаги на глубине 120см во втором монолите (среднее значение);
25. «Tens120.2max» - осмотическое давление почвенной влаги на глубине 120см в втором монолите (максимальное значение за интервал измерений);
26. «Tens120.2min» - осмотическое давление почвенной влаги на глубине 120см в втором монолите (минимальное значение за интервал измерений);
27. «Vacuum120» - понижение давления в системе отбора проб воды с глубины 120см (среднее значение);
28. «Vacuum120max» - понижение давления в системе отбора проб воды с глубины 120см (максимальное значение);
29. «Vacuum120min» - понижение давления в системе отбора проб воды с глубины 120см (минимальное значение);
30. «Accu» - напряжение аккумулятора резервного питания (среднее значение);
31. «Accumax» - напряжение аккумулятора резервного питания (максимальное значение);
32. «Accumin» - напряжение аккумулятора резервного питания (минимальное значение);
33. «UMP30.1» - влажность почвы на глубине 30см в первом монолите;
34. «UMP30.2» - влажность почвы на глубине 30см во втором монолите;
35. «UMP50.1» - влажность почвы на глубине 50см в первом монолите;
36. «UMP50.2» - влажность почвы на глубине 50см во втором монолите%
37. «UMP120.1» - влажность почвы на глубине 120см в первом монолите;
38. «UMP120.2» - влажность почвы на глубине 120см во втором монолите;
39. «EC30.1» - электропроводность почвы на глубине 30см в первом монолите;
40. «EC30.2» - электропроводность почвы на глубине 30см во втором монолите;
41. «EC50.1» - электропроводность почвы на глубине 50см в первом монолите;
42. «EC50.2» - электропроводность почвы на глубине 50см во втором монолите;
43. «EC120.1» - электропроводность почвы на глубине 120см в первом монолите;
44. «EC120.2» - электропроводность почвы на глубине 120см во втором монолите;
45. «Temp30.1» - температура почвы на глубине 30см в первом монолите;
46. «Temp30.2» - температура почвы на глубине 30см во втором монолите;
47. «Temp50.1» - температура почвы на глубине 50см в первом монолите;
48. «Temp50.2» - температура почвы на глубине 50см во втором монолите;
49. «Temp120.1» - температура почвы на глубине 120см в первом монолите;
50. «Temp120.2» - температура почвы на глубине 120см во втором монолите;
51. «Weight1» - масса первого монолита;
52. «Weight2» - масса второго монолита;
53. «Drain1» - объём стока воды через дренаж первого монолита (среднее значение);
54. «Drain1max» - объём стока воды через дренаж первого монолита (максимальное значение);
55. «Drain1min» - объём стока воды через дренаж первого монолита (минимальное значение);
56. «Drain2» - объём стока воды через дренаж второго монолита (среднее значение);
57. «Drain2max» - объём стока воды через дренаж второго монолита (максимальное значение);
58. «Drain2min» - объём стока воды через дренаж второго монолита (минимальное значение).

В результате, основываясь на смоделированной архитектуре, была разработана база данных при помощи СУБД phpMyAdmin (см. Приложение 1).

## **2.2. Разработка web-приложения**

Для разработки web-приложения и его взаимодействия с базой данных, был выбран скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений php.

Для разработки пользовательского интерфейса был использован стандартизированный язык гипертекстовой разметки документов в сети интернет HTMl.

Для реализации функции визуализации данных в виде графиков и гистограмм была использована свободно распространяемая php-библиотека pChart.

Приложение должно обеспечивать выполнение перечисленных ниже функций:

1. администрирование доступа к данным, т.е. подразделение пользователей на группы: администратор (обладает полным перечнем функций, может применять все возможные функции при работе с данными, включая добавление и удаление данных, метеостанций, почвенных станций), читатель (может только просматривать данные и графики, формировать запросы и просматривать результат);
2. возможность поиска запроса по базе данных, с учетом выбранных пользователем параметров;
3. функция визуализации данных в виде графиков и гистограмм, т.е. результат выполнения запроса из пункта 2, по желанию можно представить в виде диаграммы, тип которой выбирается из списка;
4. возможность импорта данных в базу данных из первичных файлов, т.е. запись в базу должна осуществляться посредством импорта из файла формата \*.csv, импорт данных будет осуществляться администратором с той периодичностью, в которой есть необходимость;
5. возможность экспорта данных из базы данных в файл формата \*.xls или \*.csv, т.е. для проведения более углубленной аналитики и моделирования необходимо предусмотреть экспорт данных, полученных в результате запроса в таблицы MS Excel;
6. возможность записи диаграмм в файл изображения.

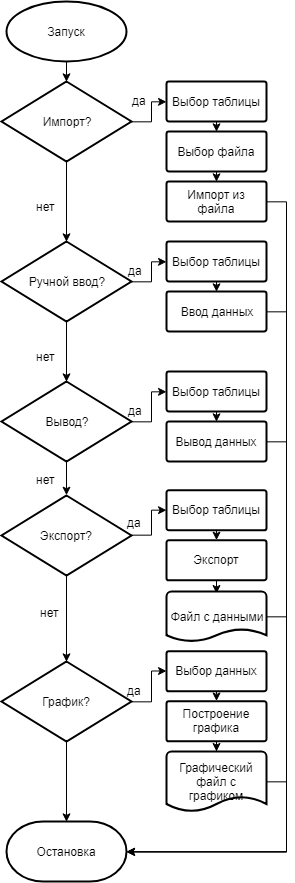


Рисунок 2.2 – Блок схема приложения

Основная структура приложения реализована с использованием следующих скриптов.

Скрипт **module.php**

Этот скрипт содержит основные классы, которые были использованы при разработке web-приложения (см. Приложение 2).

Класс mysql отвечает за основные действия, совершаемые при работе с базой данных.

Данный класс содержит методы:

1. connect() – данный метод отвечает за подключение к базе данных, расположенной на удаленном сервере, а также проверку на наличие ошибок подключения;
2. query() – данный метод отвечает за осуществление SQL-запроса к базе данных;
3. screening() – данный метод отвечает за экранирование данных.

Класс auth отвечает за аутентификацию и регистрацию пользователей на сайте.

Данный класс содержит методы:

1. reg() – данный метод отвечает за регистрацию нового пользователя в системе;
2. check\_new\_user() – данный метод отвечает за проверку данных, введенных при регистрации нового пользователя, а также за проверку заполнения обязательных полей;
3. generateCode() – данный метод отвечает за генерацию случайной строки, используемой для шифрования пароля пользователя при регистрации в системе;
4. authorization() – данный метод отвечает за авторизацию пользователей в системе путем проверки наличия пользователя в системе и задания новой сессии или же cookie;
5. check() – данный метод отвечает за проверку авторизации пользователя в системе путем проверки существования сессии или cookie;
6. isAdmin() – данный метод используется для проверки группы пользователя системы (администратор, читатель);
7. exit\_user() – данный метод используется для выхода пользователя из системы путем удаления сессии и cookie;
8. error\_print() – данный метод используется для формирования списка ошибок.

Скрипт **conf.php** отвечает за параметры подключения к базе данных (адрес сервера, имя пользователя, пароль, название базы данных), а также за подключения к базе данных, используя данные параметры (см. Приложение 3).

Скрипт **script.js** отвечает за исключение ошибок при графическом выводе данных, при помощи блокировки использования конфликтующих данных (см. Приложение 4).

# 3. ИНТЕРФЕЙС И ФУНКЦИОНАЛ

## **Стартовая страница**

При открытии web-приложения, пользователь видит стартовую страницу, представляющую из себя несколько форм, созданных при помощи языка гипертекстовой разметки HTML (Рисунок 3.1).

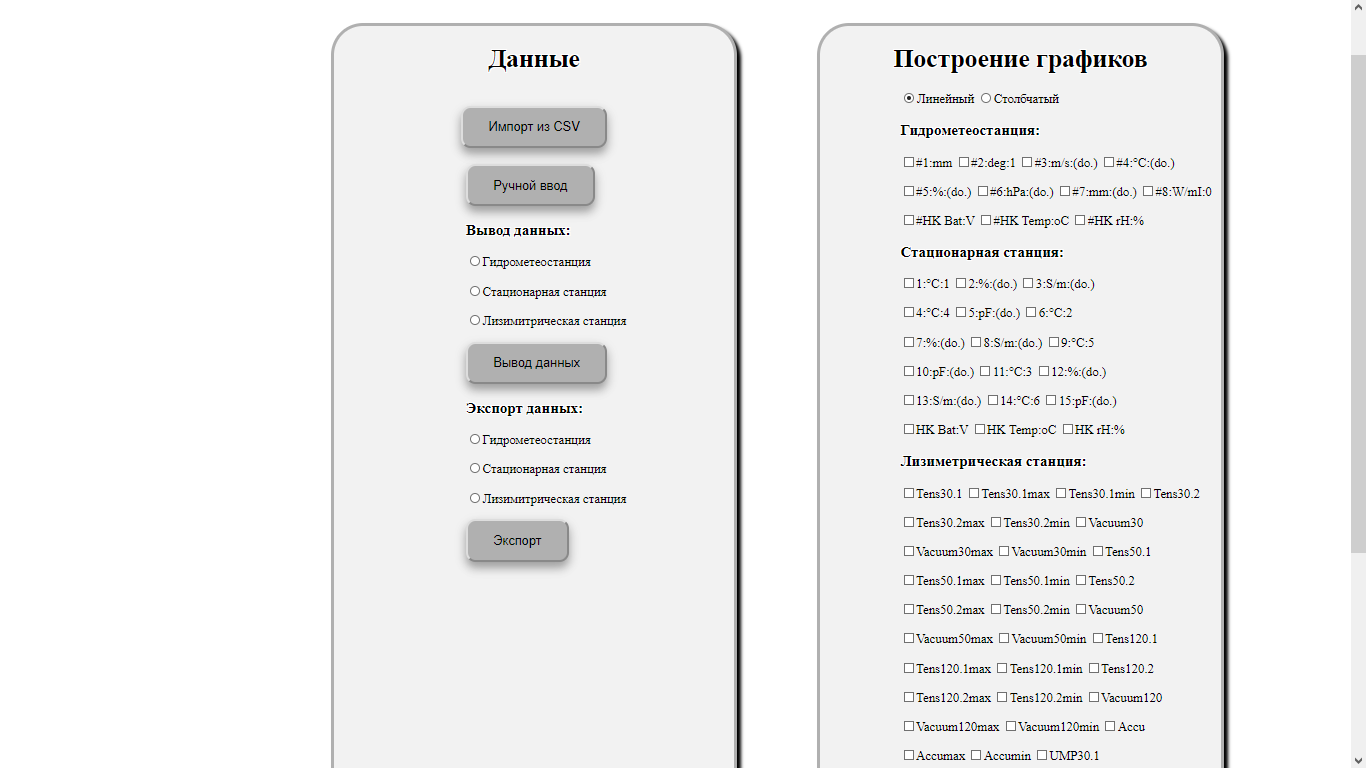


Рисунок 3.1 - стартовая страница

Блок ввода данных в базу данных состоит из двух кнопок: «Импорт из CSV» и «Ручной ввод» (Рисунок 3.2). При нажатии на кнопку «Импорт из CSV» производится переход на страницу импорта данных из файла. При нажатии на кнопку «Ручной ввод производится переход на страницу ручного ввода данных в базу данных.

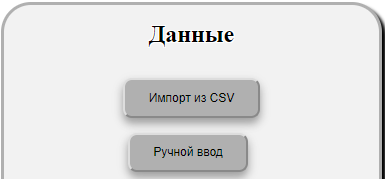


Рисунок 3.2 - блок ввода данных

Блок тестового вывода данных состоит из двух подблоков: «Вывод данных» и «Экспорт данных».

Подблок «Вывод данных» (Рисунок 3.3) состоит из выбора данных для вывода (гидрометеостанция, стационарная станция, лизиметрическая станция) и кнопки «Вывод данных». После выбора данных для вывода и нажатия кнопки «Вывод данных», производится переход на страницу табличного вывода данных.

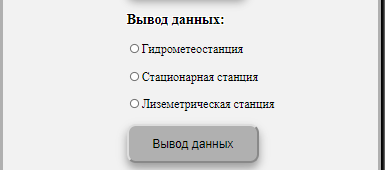


Рисунок 3.3 - подблок «Вывод данных»

Подблок «Экспорт данных» (Рисунок 3.4) состоит из выбора данных для вывода (гидрометеостанция, стационарная станция, лизиметрическая станция) и кнопки «Экспорт». После выбора данных и нажатия кнопки «Экспорт», при помощи встроенных средств языка php, формируется SQL-запрос, содержащий выбранные данные, и происходит запись данных в CSV файл (Рисунок 3.5).

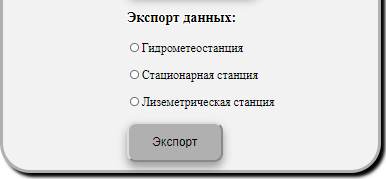


Рисунок 3.4 - подблок «Экспорт данных»

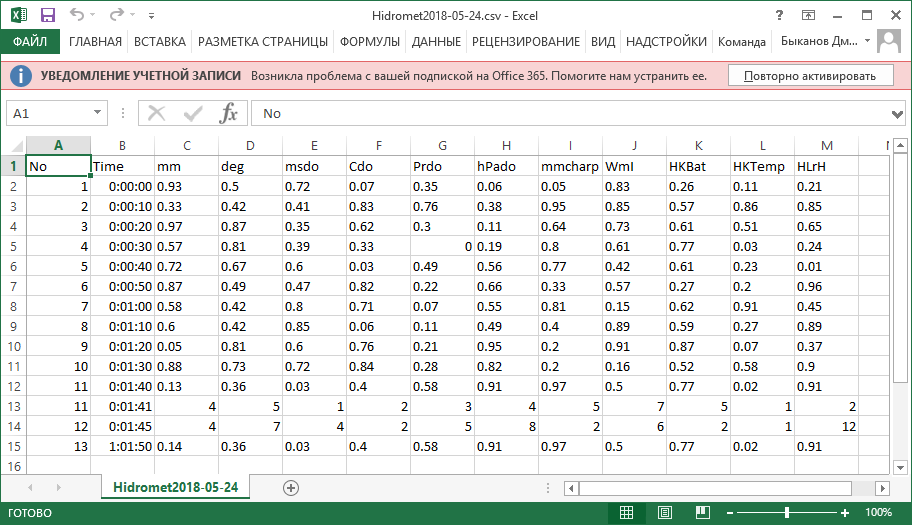


Рисунок 3.5 - пример экспортированного CSV файла

Блок графического вывода данных (Рисунок 3.6) состоит из выбора данных для вывода и кнопки «Ok». После осуществления выбора данных и нажатия кнопки «Ok», производится переход на страницу графического вывода данных.

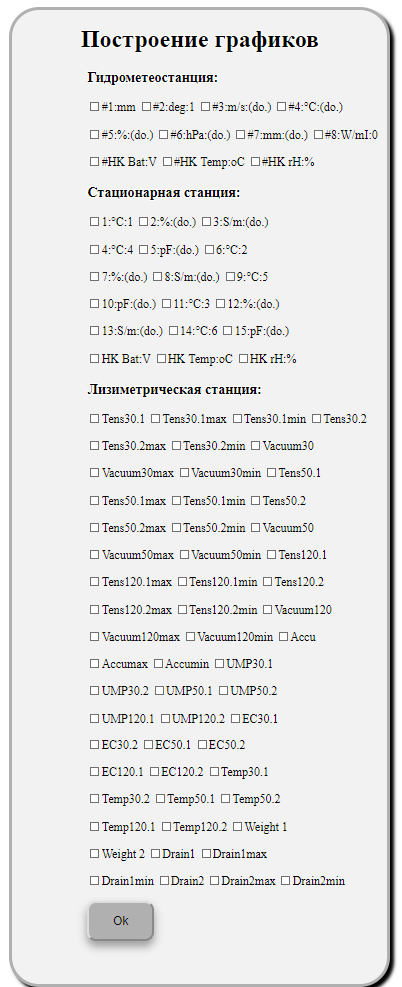


Рисунок 3.6 - блок графического вывода

## **Страница импорта данных**

Страница импорта (Рисунок 3.7) состоит из выбора данных для импорта (гидрометеостанция, стационарная станция, лизиметрическая станция), поля выбора файла для импорта и кнопки «Загрузить».

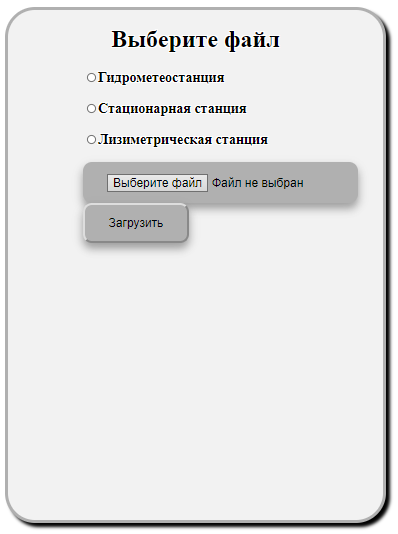


Рисунок 3.7 - страница импорта данных

При нажатии на кнопку «Выберите файл», открывается диалоговое окно, предлагающее пользователю выбрать CSV файл (Рисунок 3.8) на диске для импорта данных.

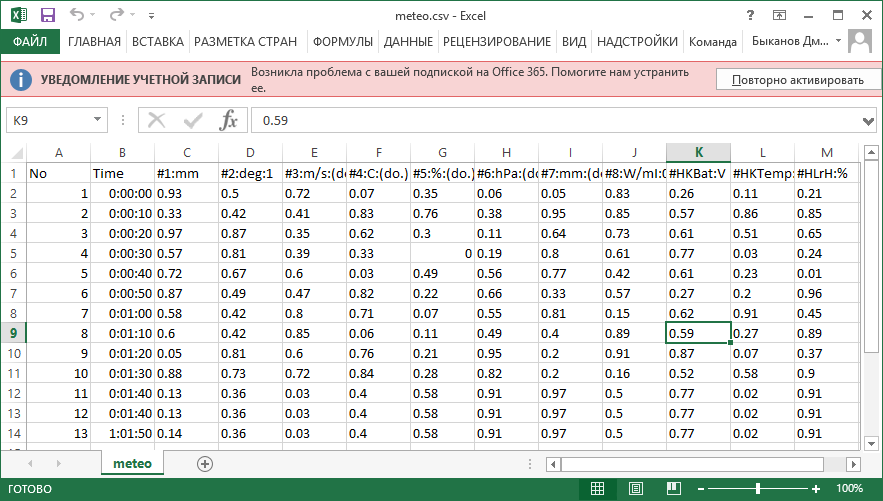


Рисунок 3.8 - пример CSV файла для импорта

После выбора данных для импорта, выбора файла для импорта и нажатия кнопки «Загрузить», при помощи встроенных средств языка php производится загрузка, выбранного пользователем файла на сервер, а также считывание данных из файла и формирования SQL-запроса для осуществления записи данных в базу данных.

## **Страница ручного ввода данных**

Страница ручного ввода данных в базу данных (Рисунок 3.9) состоит из трех форм: «Гидрометеостанция», «Стационарная», «Лизиметрическая».

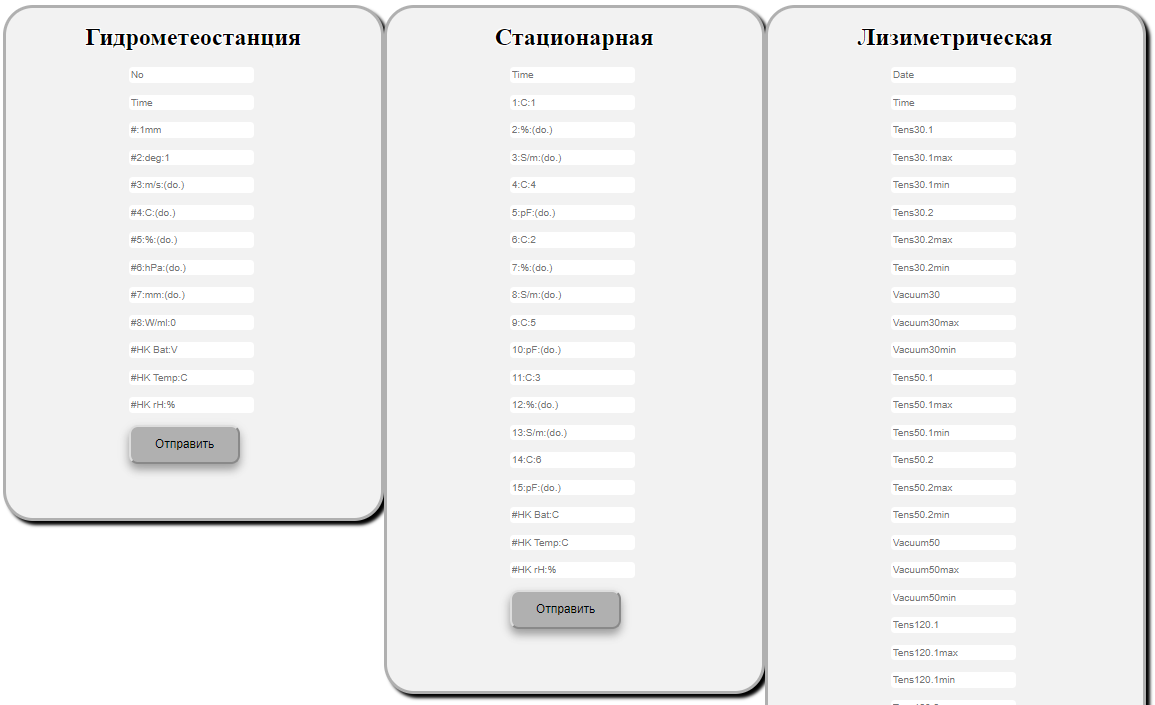


Рисунок 3.9 - страница ручного ввода данных

Каждая форма состоит из полей с обозначением данных соответствующим атрибутам в базе данных и кнопки «Отправить».

После ввода соответствующих данных и нажатия кнопки «Отправить», при помощи встроенных средств языка php формируется SQL-запрос, производящий запись этих данных в соответствующую таблицу в базе данных.

## **Страница табличного вывода данных**

Страница табличного вывода данных (Рисунок 3.10) состоит из таблицы, сформированной при помощи языка гипертекстовой разметки HTML и содержащей данные, полученные путем осуществления, с помощью встроенных средств языка php, соответствующего SQL-запроса в базу данных.

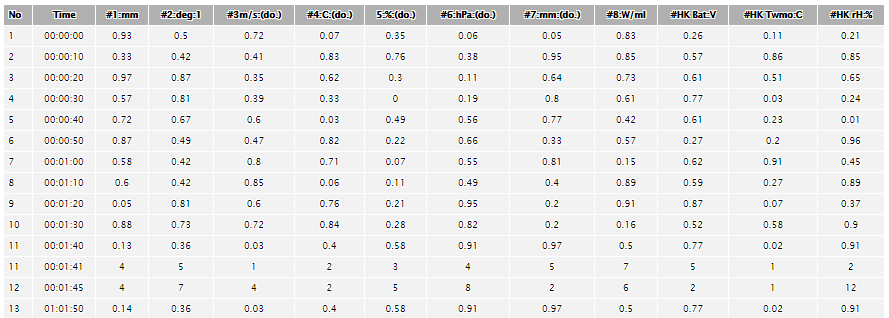


Рисунок 3.10 - страница табличного вывода данных

## **Страница графического вывода данных**

Страница графического вывода (Рисунок 3.11) состоит из изображения, сформированного при помощи встроенных средств языка php и свободно распространяемой библиотеки «pChart», которая, при помощи данных, полученных из базы данных путем формирования SQL-запроса, строит диаграммы с определенными параметрами (сетка, цвет диаграммы и т.д.).

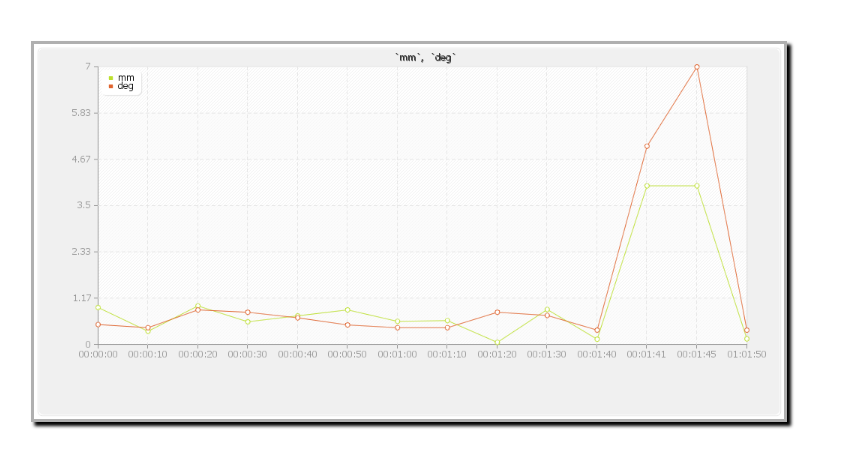


Рисунок 3.11 - страница графического вывода данных

# ЗАКЛЮЧЕНИЕ

Основная задача работы заключалась в создании базы данных и web-приложения для структурирования и первичной обработки метеорологических данных.

В результате проделанной работы были рассмотрены приложения со схожими функциями. С учетом их недостатков, были разработаны собственные база данных и web-приложение, осуществляющие структурирование метеорологических данных и помощь пользователю при работе с ними.

Работа web-приложения была протестирована на различных браузерах.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мазиров М. А., Полевые исследования свойств почв. - Владимир: Изд-во ВлГУ, 2012. — 72 с.
2. Кулунда [Электронный ресурс]. – Режим доступа: http://www.kulunda.eu/sites/default/files/КУЛУНДА%20lysimeter.pdf, свободный.
3. Центр сбора метеоданных «АСК» руководство пользователя [Электронный ресурс]. – Режим доступа: http://www.hydro-meteo.ru/doc/Ack.pdf, свободный.
4. GeoMixer [Электронный ресурс]. – Режим доступа: http://www.scanex.ru/software/web-gis/geomixer/, свободный.
5. phpMyAdmin [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/PhpMyAdmin, свободный.

# **Приложение 1. SQL-код базы данных**

-- База данных: `meteo`

-- Структура таблицы `Hidromet`

CREATE TABLE `Hidromet` (

`No` int(10) NOT NULL,

`Time` time NOT NULL,

`mm` float DEFAULT NULL,

`deg` float NOT NULL,

`msdo` float NOT NULL,

`Cdo` float NOT NULL,

`Prdo` float NOT NULL,

`hPado` float NOT NULL,

`mmcharp` float NOT NULL,

`WmI` float NOT NULL,

`HKBat` float NOT NULL,

`HKTemp` float NOT NULL,

`HLrH` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Структура таблицы `Lisemetric`

CREATE TABLE `Lisemetric` (

`Date` date NOT NULL,

`Time` time NOT NULL,

`Tens30.1` float NOT NULL,

`Tens30.1max` float NOT NULL,

`Tens30.1min` float NOT NULL,

`Tens30.2` float NOT NULL,

`Tens30.2max` float NOT NULL,

`Tens30.2min` float NOT NULL,

`Vacuum30` float NOT NULL,

`Vacuum30max` float NOT NULL,

`Vacuum30min` float NOT NULL,

`Tens50.1` float NOT NULL,

`Tens50.1max` float NOT NULL,

`Tens50.1min` float NOT NULL,

`Tens50.2` float NOT NULL,

`Tens50.2max` float NOT NULL,

`Tens50.2min` float NOT NULL,

`Vacuum50` float NOT NULL,

`Vacuum50max` float NOT NULL,

`Vacuum50min` float NOT NULL,

`Tens120.1` float NOT NULL,

`Tens120.1max` float NOT NULL,

`Tens120.1min` float NOT NULL,

`Tens120.2` float NOT NULL,

`Tens120.2max` float NOT NULL,

`Tens120.2min` float NOT NULL,

`Vacuum120` float NOT NULL,

`Vacuum120max` float NOT NULL,

`Vacuum120min` float NOT NULL,

`Accu` float NOT NULL,

`Accumax` float NOT NULL,

`Accumin` float NOT NULL,

`UMP30.1` float NOT NULL,

`UMP30.2` float NOT NULL,

`UMP50.1` float NOT NULL,

`UMP50.2` float NOT NULL,

`UMP120.1` float NOT NULL,

`UMP120.2` float NOT NULL,

`EC30.1` float NOT NULL,

`EC30.2` float NOT NULL,

`EC50.1` float NOT NULL,

`EC50.2` float NOT NULL,

`EC120.1` float NOT NULL,

`EC120.2` float NOT NULL,

`Temp30.1` float NOT NULL,

`Temp30.2` float NOT NULL,

`Temp50.1` float NOT NULL,

`Temp50.2` float NOT NULL,

`Temp120.1` float NOT NULL,

`Temp120.2` float NOT NULL,

`Weight1` float NOT NULL,

`Weight2` float NOT NULL,

`Drain1` float NOT NULL,

`Drain1max` float NOT NULL,

`Drain1min` float NOT NULL,

`Drain2` float NOT NULL,

`Drain2max` float NOT NULL,

`Drain2min` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Структура таблицы `Stational`

CREATE TABLE `Stational` (

`Time` time NOT NULL,

`CThirdty` float NOT NULL,

`procdoThirdty` float NOT NULL,

`SmdoThirdty` float NOT NULL,

`CThirdtyF` float NOT NULL,

`pFdoThirdty` float NOT NULL,

`CSixty` float NOT NULL,

`procdoSixty` float NOT NULL,

`SmdoSixty` float NOT NULL,

`CSixtyF` float NOT NULL,

`pFdoSixty` float NOT NULL,

`CHThirdty` float NOT NULL,

`doHThirdty` float NOT NULL,

`SmdoHThirdty` float NOT NULL,

`CHThirdtyF` float NOT NULL,

`pFdoHThirdty` float NOT NULL,

`HKBatV` float NOT NULL,

`HKTempC` float NOT NULL,

`HKrH` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Индексы таблицы `Hidromet`

ALTER TABLE `Hidromet`

ADD PRIMARY KEY (`Time`);

-- Индексы таблицы `Lisemetric`

ALTER TABLE `Lisemetric`

ADD PRIMARY KEY (`Date`,`Time`);

-- Индексы таблицы `Stational`

ALTER TABLE `Stational`

ADD PRIMARY KEY (`Time`);

# **Приложение 2. Скрипт module.php**

<?php

class mysql {

###

# Подключение к бд

function connect($db\_host, $db\_login, $db\_passwd, $db\_name) {

mysql\_connect($db\_host, $db\_login, $db\_passwd) or die ("MySQL Error: " . mysql\_error()); //~ устанавливаем подключение с бд

mysql\_query("set names utf8") or die ("<br>Invalid query: " . mysql\_error()); //~ указываем что передаем данные в utf8

mysql\_select\_db($db\_name) or die ("<br>Invalid query: " . mysql\_error()); //~ выбираем базу данных

}

###

# Запрос к базе и его производные

function query($query, $type, $num) {

if ($q=mysql\_query($query)) {

switch ($type) {

case 'num\_row' : return mysql\_num\_rows($q); break;

case 'result' : return mysql\_result($q, $num); break;

case 'accos' : return mysql\_fetch\_assoc($q); break;

case 'none' : return $q;

default: return $q;

}

} else {

print 'Mysql error: '.mysql\_error();

return false;

}

}

###

# экранирование данных

function screening($data) {

$data = trim($data); //~ удаление пробелов из начала и конца строки

return mysql\_real\_escape\_string($data); //~ экранирование символов

}

}

class auth {

###

# Проверка входных данных при регистрации

function check\_new\_user($login, $passwd, $passwd2, $mail) {

//~ Проверка валидности данных

if (empty($login) or empty($passwd) or empty($passwd2)) $error[]='Все поля обязательны для заполнения';

if ($passwd != $passwd2) $error[]='Введенные пароли не совпадают';

if (strlen($login)<3 or strlen($login)>30) $error[]='Длинна логина должна быть от 3 до 30 символов';

if (strlen($passwd)<3 or strlen($passwd)>30) $error[]='Длинна пароля должна быть от 3 до 30 символов';

//~ Валидация почты не используя регулярки http://www.php.net/manual/en/filter.examples.validation.php

if (!filter\_var($mail, FILTER\_VALIDATE\_EMAIL)) $error[]='Не корректный email';

//~ Проверяем наличее пользователя с таким именем в бд

$db = new mysql(); //~ Создаем новый объект класса

$login = $db->screening($login);

if ($db->query("SELECT \* FROM users WHERE login\_user='".$login."';", 'num\_row', '')!=0) $error[]='Пользователь с таким именем уже существует';

if ($db->query("SELECT \* FROM users WHERE mail\_user='".$mail."';", 'num\_row', '')!=0) $error[]='Пользователь с таким email уже существует';

//~ Возвращаем массив ошибок или положительный ответ

if (isset($error)) return $error;

else return 'good';

}

###

# Регистрация

function reg($login, $passwd, $passwd2, $mail) {

if (($this->check\_new\_user($login, $passwd, $passwd2, $mail))=='good') {

$db = new mysql(); //~ Создаем новый объект класса

$passwd = md5($db->screening($passwd).'lol'); //~ хеш пароля с солью

$login = $db->screening($login);

if ($db->query("INSERT INTO `users` (`id\_user`, `login\_user`, `passwd\_user`, `mail\_user`) VALUES (NULL, '".$login."', '".$passwd."', '".$mail."');", '', '')) return true;

else {

print 'Возникла ошибка при регистрации нового пользователя. Свяжитесь с администрацией';

return false;

}

} else {

print $this->error\_print($this->check\_new\_user($login, $passwd, $passwd2, $mail));

return false;

}

}

###

# Проверка авторизации

function check() {

if (isset($\_SESSION['id\_user']) and isset($\_SESSION['login\_user'])) {return true; print 'Сессия';}

else {

//~ проверяем наличие кук

if (isset($\_COOKIE['id\_user']) and isset($\_COOKIE['code\_user'])) {

//~ куки есть - сверяем с таблицей сессий

// print 'Кука';

$db = new mysql(); //~ создаем новый объект класса

$id\_user=$db->screening($\_COOKIE['id\_user']);

$code\_user=$db->screening($\_COOKIE['code\_user']);

if ($db->query("SELECT \* FROM `session` WHERE `id\_user`=".$id\_user.";", 'num\_row', '')==1) {

//~ Есть запись в таблице сессий, сверяем данные

$data = $db->query("SELECT \* FROM `session` WHERE `id\_user`=".$id\_user.";", 'accos', '');

if ($data['code\_sess']==$code\_user and $data['user\_agent\_sess']==$\_SERVER['HTTP\_USER\_AGENT']) {

//~ Данные верны, стартуем сессию

$\_SESSION['id\_user']=$id\_user;

$\_SESSION['login\_user']=$db->query("SELECT login\_user FROM `users` WHERE `id\_user` = '".$id\_user."';", 'result', 0);

//~ обновляем куки

setcookie("id\_user", $\_SESSION['id\_user'], time()+3600\*24\*14);

setcookie("code\_user", $code\_user, time()+3600\*24\*14);

return true;

} else return false; //~ данные в таблице сессий не совпадают с куками

} else return false; //~ в таблице сессий не найден такой пользователь

} else return false;

}

}

###

# Авторизация

function authorization() {

$db = new mysql(); //~ создаем новый объект класса

$login = $db->screening($\_POST['login']);

$passwd = md5($db->screening($\_POST['passwd']).'lol'); //~ хеш пароля с солью

if ($db->query("SELECT \* FROM `users` WHERE `login\_user` = '".$login."' AND `passwd\_user` = '".$passwd."';", 'num\_row', '')==1) {

//~ пользователь найден в бд, логин совпадает с паролем

$\_SESSION['id\_user']=$db->query("SELECT \* FROM `users` WHERE `login\_user` = '".$login."' AND `passwd\_user` = '".$passwd."';", 'result', 0);

$\_SESSION['login\_user']=$login;

//~ добавляем/обновляем запись в таблице сессий и ставим куку

$r\_code = $this->generateCode(15);

if ($db->query("SELECT \* FROM `session` WHERE `id\_user`=".$\_SESSION['id\_user'].";", 'num\_row', '')==1) {

//~ запись уже есть - обновляем

$db->query("UPDATE `session` SET `code\_sess` = '".$r\_code."', `user\_agent\_sess` = '".$\_SERVER['HTTP\_USER\_AGENT']."' WHERE `id\_user` = ".$\_SESSION['id\_user'].";", '', '');

} else {

//~ записи нету - добавляем

$db->query("INSERT INTO `session` (`id\_user`, `code\_sess`, `user\_agent\_sess`) VALUES ('".$\_SESSION['id\_user']."', '".$r\_code."', '".$\_SERVER['HTTP\_USER\_AGENT']."');", '', '');

}

//~ ставим куки на 2 недели

setcookie("id\_user", $\_SESSION['id\_user'], time()+3600\*24\*14);

setcookie("code\_user", $r\_code, time()+3600\*24\*14);

return true;

} else {

//~ пользователь не найден в бд, или пароль не соответствует введенному

if ($db->query("SELECT \* FROM `users` WHERE `login\_user` = '".$login."';", 'num\_row', 0)==1) $error[]='Введен не верный пароль';

else $error[]='Такой пользователь не существует';

$\_SESSION['error'] = $this->error\_print($error);

return false;

}

}

###

# Выход

function exit\_user() {

//~ разрушаем сессию, удаляем куки и отправляем на главную

session\_destroy();

setcookie("id\_user", '', time()-3600);

setcookie("code\_user", '', time()-3600);

header("Location: login.php");

}

function isAdmin(){

$db= new mysql();

if (isset($\_SESSION['id\_user']) and isset($\_SESSION['login\_user']))

$id\_user=$db->screening($\_SESSION['id\_user']);

else {

//~ проверяем наличие кук

if (isset($\_COOKIE['id\_user']) and isset($\_COOKIE['code\_user']))

$id\_user=$db->screening($\_COOKIE['id\_user']);

}

$adm=$db -> query("SELECT `admin` FROM `users` WHERE `id\_user` = '".$id\_user."';", 'result', 0);

return $adm;

}

###

# Восстановление пароля

function recovery\_pass($login, $mail) {

$db = new mysql(); //~ создаем новый объект класса

$login = $db->screening($login);

$db\_inf = $db->query("SELECT \* FROM `users` WHERE `login\_user`='".$login."';", 'accos', '');

if ($db->query("SELECT \* FROM `users` WHERE `login\_user`='".$login."';", 'num\_row', '')!=1) {

//~ не найден такой пользователь

$error[]='Пользователь с таким именем не найден';

return $this->error\_print($error);

} else {

//~ проверка email

if (!filter\_var($mail, FILTER\_VALIDATE\_EMAIL)) $error[]='Введен не корректный email';

if ($mail != $db\_inf['mail\_user']) $error[]='Введенный email не соответствует введенному при регистрации ';

if (!isset($error)) {

//~ восстанавливаем пароль

$new\_passwd = $this->generateCode(8);

$new\_passwd\_sql = md5($new\_passwd.'lol');

$message = "Вы запросили восстановление пароля на сайте %sitename% для учетной записи ".$db\_inf['login\_user']." \nВаш новый пароль: ".$new\_passwd."\n\n С уважением администрация сайта %sitename%.";

if (mail($mail, "Восстановление пароля", $message, "From: webmaster@sitename.ru\r\n"."Reply-To: webmaster@sitename.ru\r\n"."X-Mailer: PHP/" . phpversion())) {

//~ почта отправлена, обновляем пароль в базе

$db->query("UPDATE `users` SET `passwd\_user`='".$new\_passwd\_sql."' WHERE `id\_user` = ".$db\_inf['id\_user'].";", '', '');

//~ все успешно - возвращаем положительный ответ

return 'good';

} else {

//~ ошибка при отправке письма

$error[]='В данный момент восстановление пароля не возможно, свяжитесь с администрацией сайта';

return $this->error\_print($error);

}

} else return $this->error\_print($error);

}

}

# Функция генерации случайной строки

function generateCode($length) {

$chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPRQSTUVWXYZ0123456789";

$code = "";

$clen = strlen($chars) - 1;

while (strlen($code) < $length) {

$code .= $chars[mt\_rand(0,$clen)];

}

return $code;

}

# Формирование списка ошибок

function error\_print($error) {

$r='<h2>Произошли следующие ошибки:</h2>'."\n".'<ul>';

foreach($error as $key=>$value) {

$r.='<li>'.$value.'</li>';

}

return $r.'</ul>';

}

}

?>

# **Приложение 3. Скрипт conf.php**

<?php

error\_reporting(0);

//~ Старт сессии, файл должен быть сохранен без DOM информации

session\_start();

include 'module.php';

//~ Параметры потключения к бд

$db\_host = 'localhost';

$db\_login = 'admin';

$db\_passwd = '1234';

$db\_name = 'meteo';

// подключаемся к бд

$db = new mysql(); //~ Создаем новый объект класса

$db -> connect($db\_host, $db\_login, $db\_passwd, $db\_name);

?>

# **Приложение 4. Скрипт script.js**

function checkCB() {

var f1 = 0;

var f2 = 0;

var f3=0;

var inp = [];

for (var i = 3; i <= 87; i++)

document.getElementById(i).disabled = false;

for (var i = 3; i <= 13; i++) {

inp[i] = document.getElementById(i);

if (inp[i].checked == true)

f1++;

}

if (f1 > 0){

for (var i = 14; i <= 31; i++)

document.getElementById(i).disabled = true;

for (var i=32; i<=87; i++)

document.getElementById(i).disabled = true;

}

for (var i = 14; i <= 31; i++) {

inp[i] = document.getElementById(i);

if (inp[i].checked == true)

f2++;

}

if (f2 > 0)

{

for (var i = 3; i <= 13; i++)

document.getElementById(i).disabled = true;

for (var i=32; i<=87; i++)

document.getElementById(i).disabled = true;

}

for (var i = 32; i <= 87; i++) {

inp[i] = document.getElementById(i);

if (inp[i].checked == true)

f3++;

}

if (f3>0)

{

for (var i = 3; i <= 13; i++)

document.getElementById(i).disabled = true;

for (var i=14; i<=31; i++)

document.getElementById(i).disabled = true;

}

}

**ПОСЛЕДНИЙ ЛИСТ ВКР**

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

*(подпись) (Ф.И.О)*