**Содержание**

[Введение 4](#_Toc37170794)

[1. Предварительный анализ 5](#_Toc37170795)

[2. Основные проектные решения 9](#_Toc37170796)

[2.1 Разработка интерфейса пользователя 10](#_Toc37170797)

[2.2 Разработка модуля программы 23](#_Toc37170798)

[2.3 Разработка печатной формы (отчета) 28](#_Toc37170799)

[2.4 Разработка ролей и пользователей 30](#_Toc37170800)

[3. Тестирование 34](#_Toc37170801)

[3.1 Описание контрольного примера 34](#_Toc37170802)

[3.2 Тестирование системы 35](#_Toc37170803)

[Заключение 38](#_Toc37170804)

[Литература 39](#_Toc37170805)

# 

# Введение

Данный курсовой проект выполнен в соответствии с заданием на курсовое проектирование. Темой курсового проекта является «Эксплуатация и модификация информационной системы по учету статистики данных о загазованности водопроводов города».

Главной целью курсового проекта является закрепление полученных знаний во время обучения дисциплины ПМ. 01. Эксплуатация и модификация информационных систем.

Задачами курсового проектирования являются:

* изучение предметной области;
* выбор языков программирования и технических средств разработки ИС;
* разработка структуры базы данных;
* разработка системы показа данных в читаемом виде;
* разработка интерфейса для пользователя;
* тестирование системы;

В ходе курсового проектирования была изучена предметная область по теме, изучены основные понятия нефтяной отрасли, разработаны алгоритмы решения задачи, которые затем были реализованы в программном коде, выполнено тестирование программного продукта. В итоге разработан качественный продукт, выполненный в соответствии со стандартами разработки приложений.

Преимуществом данного программного продукта заключается в том, что будущие пользователи смогут получать информацию с трубопроводов или иных объектах, с минимальными затратами времени. С помощью легкого интерфейса и простой навигации, что не вызывает осложнений даже у начинающего пользователя.

# 1. Предварительный анализ

Определение загазованности в водопроводе

В соответствии с требованиями Межотраслевых правил по охране труда при эксплуатации водопроводно-канализационного хозяйства контроль загазованности в водопроводных и канализационных колодцах следует осуществлять с помощью газоанализаторов.

При этом используются два основных дистанционных способа контроля:

* с помощью шланга и насоса;
* с помощью измерительного модуля на кабеле;

Независимо от способа контроля загазованности установлен порядок работы с газоанализатором: перед спуском человека в колодец проверяется отсутствие загазованности. В одном случае в колодец опускается шланг, а контролируемая воздушная смесь со дна колодца насосом прокачивается на поверхность к газоанализатору. В другом случае в колодец опускается измерительный модуль, и с него по кабелю на поверхность передается информация о контролируемой среде. Если загазованность отсутствует, человек спускается в колодец и приступает к работе.

Уровень обеспечения безопасности человека, работающего в колодце, существенно зависит от способа контроля загазованности. Эта зависимость становится все более очевидной при глубине колодцев более 5 м. Действительно, от глубины колодца зависит время получения сигнала о загазованности, т. е. время, в течение которого спасатель сможет оказать экстренную помощь работающему в колодце человеку. Например, потери времени при прокачке контролируемой воздушной среды через шланг длиной 30 м с помощью современных газоанализаторов составляют при идеальных условиях более 2–3 минут!

Применение газоанализаторов, использующих такой способ контроля загазованности в глубоких колодцах, ставит безопасность людей под угрозу.

Способ контроля загазованности с помощью измерительного модуля на кабеле, помимо неоспоримого преимущества в скорости передачи информации, позволяет работающему в колодце полностью контролировать уровень собственной безопасности, чего он лишен при использовании газоанализаторов со шланговой системой прокачки. Так, рабочий, находясь на большой глубине в колодце, постоянно видит, что измерительный модуль включен и исправен, воздушная среда в норме. Информацию о возникшей опасности рабочий увидит одновременно со спасателем и предпримет усилия для своего спасения, не дожидаясь помощи. Более того, между рабочим и спасателем возможен обмен стандартными фразами или передача светозвуковых сигналов из колодца на поверхность земли.

При глубине колодца до 5 м с точки зрения безопасности не имеет значения, какой способ контроля загазованности используется в газоанализаторе. Выбор модели в этом случае основывается на удобстве, надежности и прочих эксплуатационных характеристиках.

При глубине колодца более 5 м способ контроля загазованности с помощью шланга и насоса не должен быть использован, поскольку существует риск несчастного случая. И чем глубже колодец, тем выше риск. При этом потери времени при использовании шланга и насоса – не единственный фактор увеличения риска. В глубоком колодце даже при отсутствии загазованности воздушная среда более «тяжелая» для пребывания человека, чем в колодце глубиной до 5 м.

Поскольку загазованность может появиться уже в процессе работы человека в колодце (например, когда взмучивается ил на дне, удаляется «пробка», осыпается грунт со стен), Правила требуют не только определить отсутствие загазованности перед началом работ, но и непрерывно отслеживать ее в течение всего времени их выполнения.

При выполнении работ на сетях водоснабжения и канализации должны учитываться возможные специфические опасные и вредные производственные факторы, а именно:

* загазованность колодцев, камер, коллекторов ядовитыми и взрывоопасными газами, что может привести к взрыву, отравлению или ожогам работников;
* возможность падения в колодцы, камеры, емкостные сооружения при спуске в них, а также получение ушибов при открывании и закрывании крышек люков;
* падение различных предметов в открытые люки на работников, работающих в колодцах, камерах;
* опасность воздействия потоков воды на работников, работающих в колодцах, камерах и коллекторах;
* опасность обрушения грунта при выполнении земляных работ;
* опасность наезда транспортных средств при работе на проезжей части улиц;
* повышенная влажность воздушной среды при работе в колодцах, камерах и коллекторах;
* биологическая опасность при соприкосновении со сточными водами.

Работы, связанные со спуском работников в колодцы, камеры, резервуары, аварийно-регулирующие резервуары, насосные станции без принудительной вентиляции, опорожненные напорные водоводы и канализационные коллектора относятся к разряду опасных, к которым предъявляются дополнительные (повышенные) требования безопасности труда и должны проводиться по наряду-допуску на выполнение работ повышенной опасности.

Бригады, выполняющие вышеуказанные работы должны быть обеспечены защитными средствами, необходимым инструментом, инвентарем, приспособлениями, приборами и аптечкой первой доврачебной помощи.

Датчик измеряет текущую загазованность и передает текущие данные в базу данных.

Ведение контроля над датчиком считывающий текущую загазованность необходима для быстрого выявления аварий и направлении работников ремонтной службы на место аварии.

Также датчики позволяют проводить круглосуточный мониторинг и на основе полученных данных вести статистику.

# 2. Основные проектные решения

Данное приложение было разработано с помощью языка разметки HTML, языка описания стилей CSS3, языков программирования PHP, JS и базой данных phpMyAdmin.

HTML — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

CSS — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL.

Диаграмма прецедентов описывает, как администратор, пользователь и гость могут использовать разработанный информационный ресурс. (Рис 2.1)

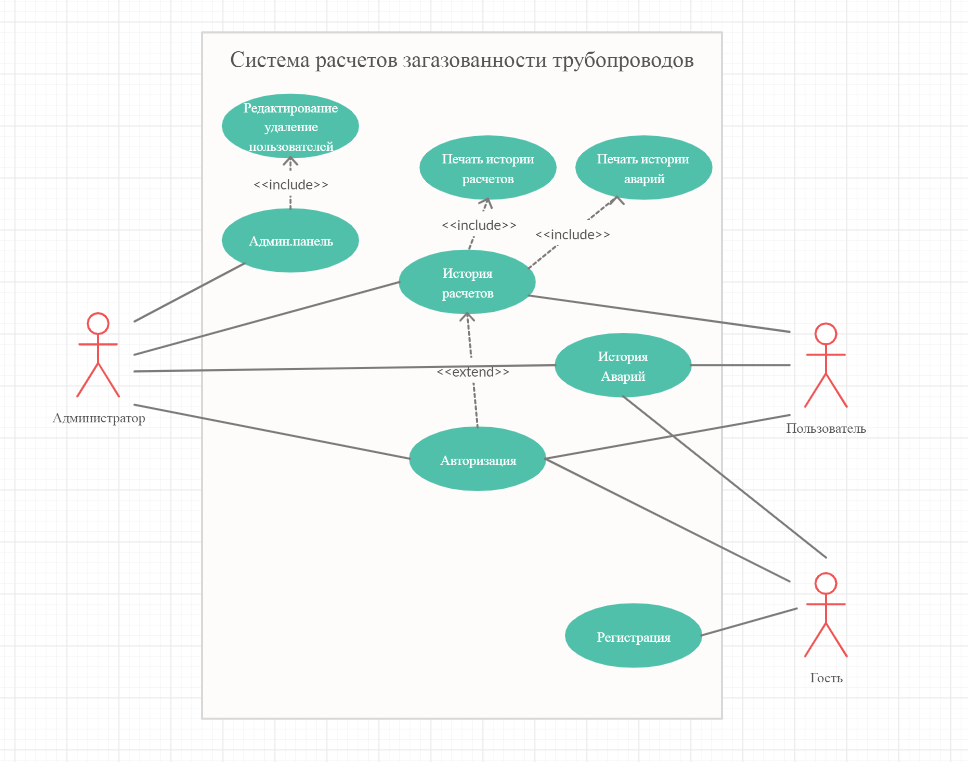


Рис 2.1 Диаграмма прецедентов

# 2.1 Разработка интерфейса пользователя

Интерфейс пользователя, он же пользовательский интерфейс - [интерфейс](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), обеспечивающий передачу информации между [пользователем](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C) и [аппаратными](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) компонентами [компьютерной](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) системы ([ISO](https://ru.wikipedia.org/wiki/ISO)/IEC/IEEE 247652010).

Под совокупностью средств и методов интерфейса пользователя подразумеваются:

Средства:

* вывода информации из устройства к пользователю — весь доступный диапазон [воздействий на организм человека](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D1%8B_%D1%87%D1%83%D0%B2%D1%81%D1%82%D0%B2) (зрительных, слуховых, тактильных, обонятельных и т. д.) — [экраны](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%80%D0%B0%D0%BD) (дисплеи, проекторы) и лампочки, динамики, зуммеры и сирены, вибромоторы и т. д. и т. п.
* ввода информации/команд пользователем в устройство — множество всевозможных устройств для контроля состояния человека — кнопки, переключатели, [потенциометры](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%BE%D0%BC%D0%B5%D1%82%D1%80), датчики положения и движения, сервоприводы, [жесты лицом](https://ru.wikipedia.org/w/index.php?title=%D0%96%D0%B5%D1%81%D1%82%D1%8B_%D0%BB%D0%B8%D1%86%D0%BE%D0%BC&action=edit&redlink=1) и руками, даже [съём мозговой активности](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) пользоват-- еля.

По наличию тех или иных средств ввода, интерфейсы разделяются на типы — жестовый, голосовой, [брэйн](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), и т. д., возможны смешанные варианты. Средства эти должны быть необходимыми и достаточными, быть удобными и практичными, расположенными и скомпонованными разумно и понятно, соответствовать физиологии человека, не должны приводить к негативным последствиям для организма пользователя (всё это входит в понятие [эргономики](https://ru.wikipedia.org/wiki/%D0%AD%D1%80%D0%B3%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B0)).

Методы:

* набор правил, заложенных разработчиком устройства, согласно которым совокупность действий пользователя должна привести к необходимой реакции устройства и выполнения требуемой задачи — так называемый [логический интерфейс](https://ru.wikipedia.org/w/index.php?title=%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81&action=edit&redlink=1).

Интерфейс пользователя [компьютерного приложения](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) включает:

* средства отображения информации, отображаемую информацию, [форматы](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82) и [коды](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4);
* командные режимы, язык «пользователь — интерфейс»;
* устройства и технологии ввода данных;
* диалоги, взаимодействие и транзакции между пользователем и компьютером, обратную связь с пользователем;
* поддержку принятия решений в конкретной [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C);
* порядок использования программы и документацию на неё.

Пользовательский интерфейс часто понимают только как внешний вид программы. Однако, на деле пользователь воспринимает через него всю программу в целом, а значит, такое понимание является слишком узким.  
В действительности ПИ объединяет в себе все элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением (ПО), это не только экран, который видит пользователь.

К этим элементам относятся:

* набор задач пользователя, которые он решает при помощи системы;
* используемая системой метафора (например, рабочий стол в MS Windows®);
* элементы управления системой;
* навигация между блоками системы;
* визуальный (и не только) дизайн экранов программы;
* средства отображения информации, отображаемая информация и форматы;
* устройства и технологии ввода данных;
* диалоги, взаимодействие и транзакции между пользователем и компьютером;
* обратная связь с пользователем;
* поддержка принятия решений в конкретной предметной области;
* порядок использования программы и документация на неё.

Для упрощения восприятия функции программы пользователем при разработке пользовательского интерфейса желательно использовать [метафоры](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D1%84%D0%BE%D1%80%D0%B0).

При первоначальном запуске пользователя встречает главная страница сайта (Рис. 2.2)

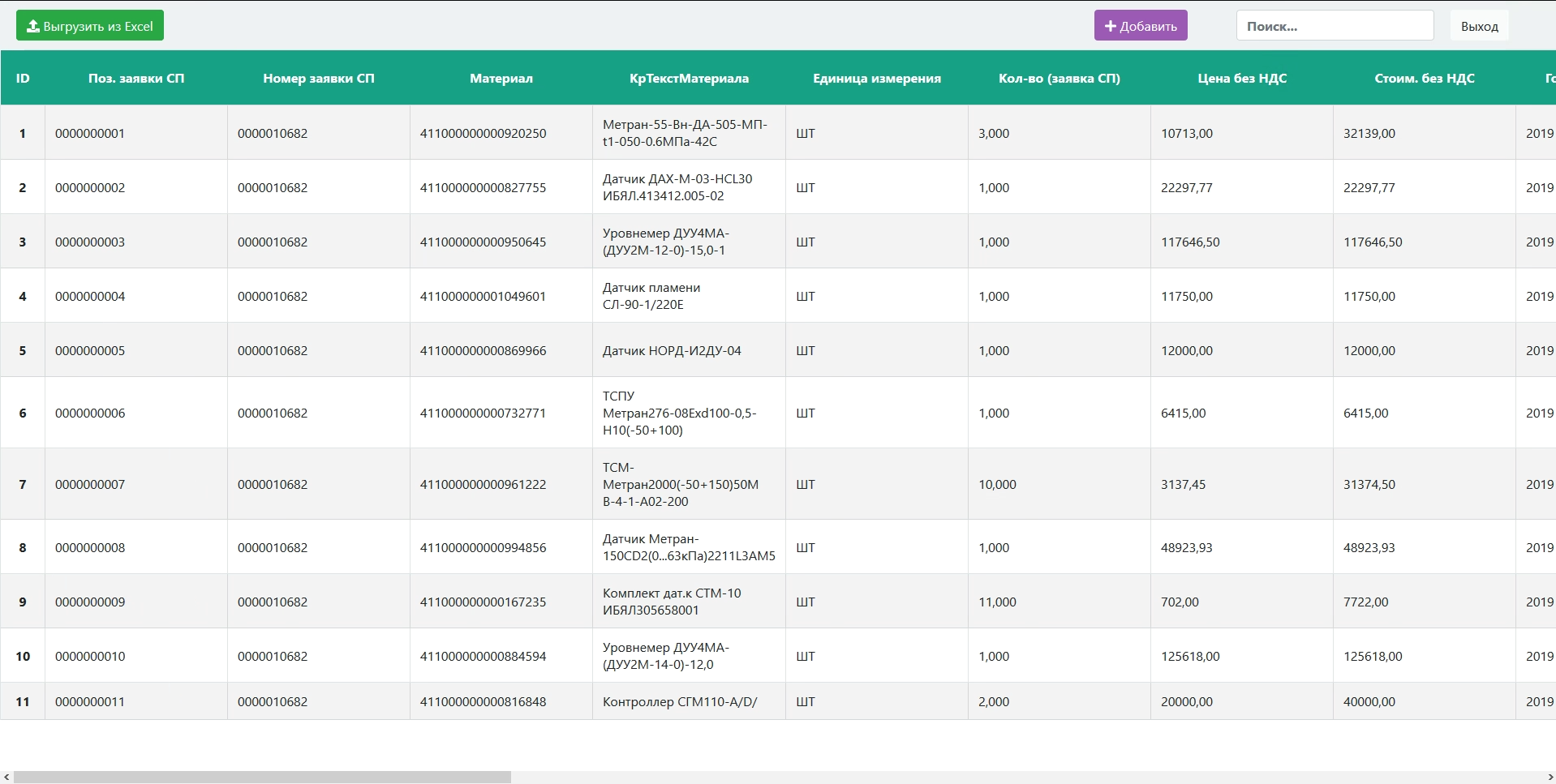


Рис. 2.2 - Главная страница

Далее нажимаем на кнопку «Регистрация» и выходит всплывающее окно с формой (Рис. 2.3)

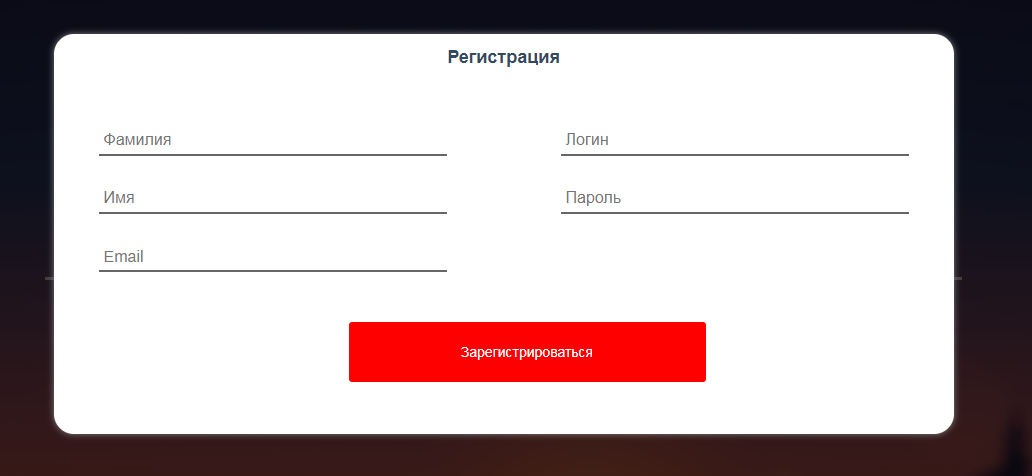


Рис. 2.3 - Страница главного сайта

Нажав на кнопку «Авторизация» выходит всплывающее окно с входными данными (Рис. 2.4)

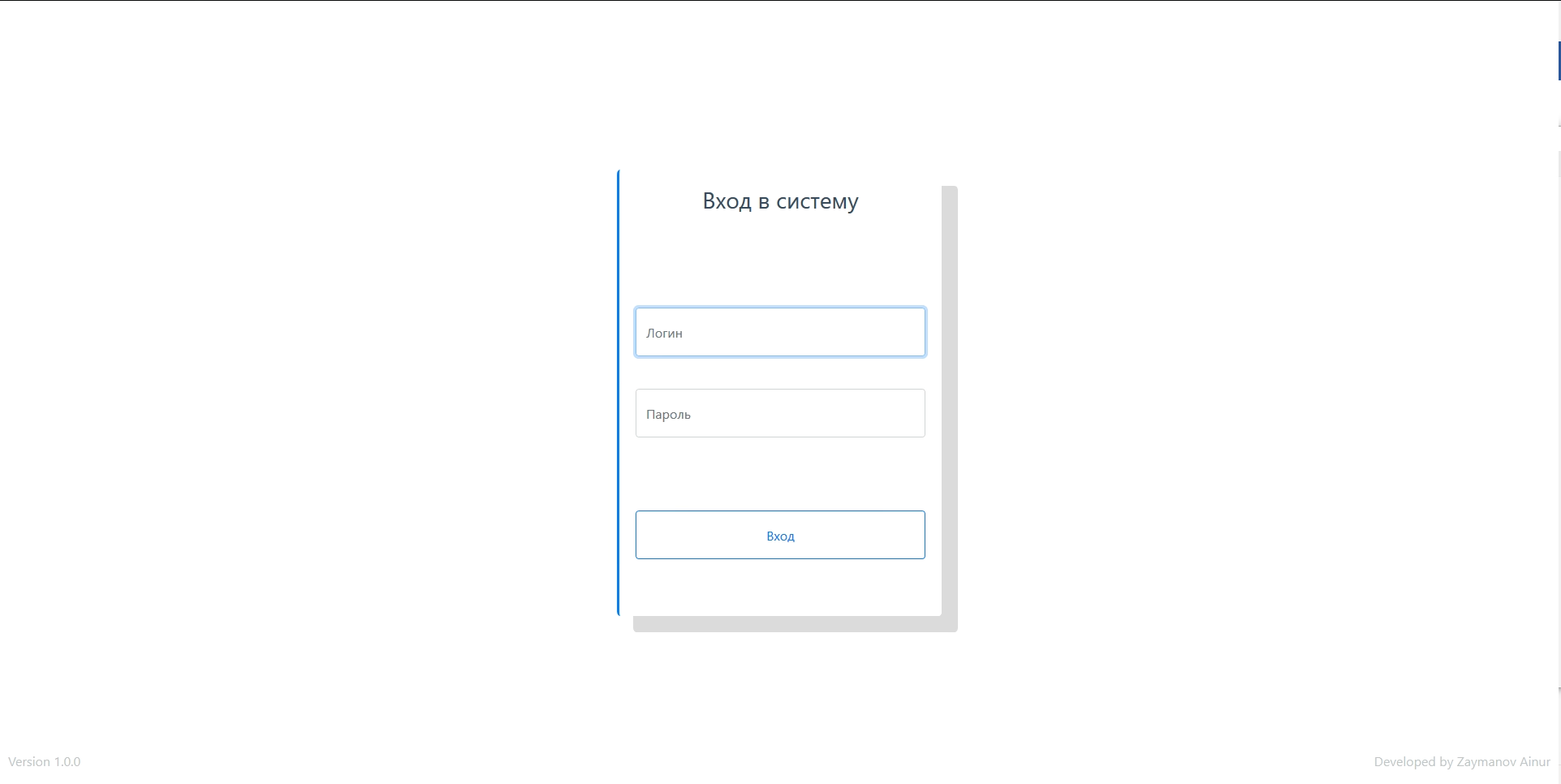


Рис. 2.4 - Статистика загазованности

Переходим во вкладку «Расчет» и выходит таблица с формой расчета и входными данными. (Рис. 2.5)

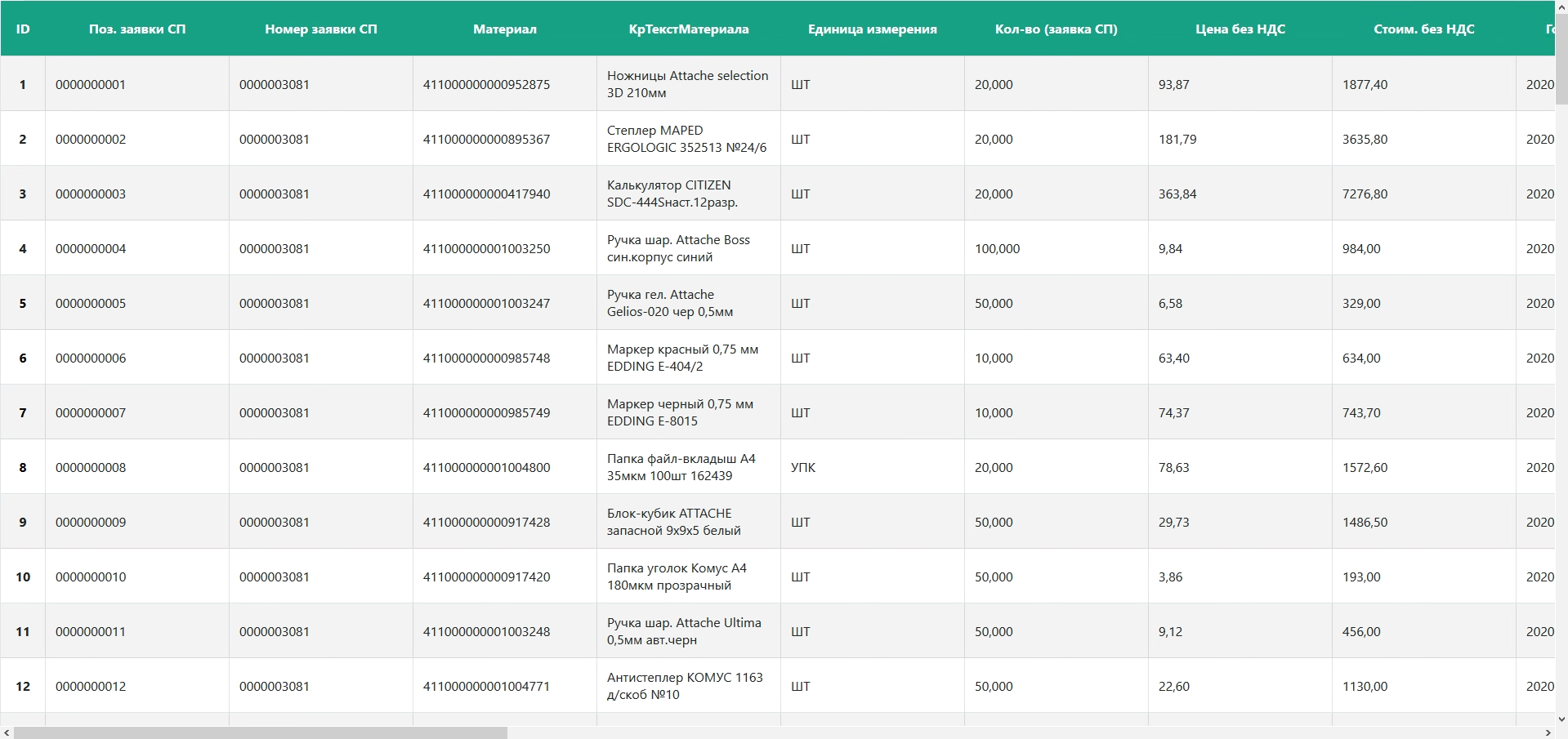


Рис. 2.5 - Страница с расчетами

Перейдя на вкладку «Панель администратора» выходит таблица с формой аварии. (Рис. 2.6)

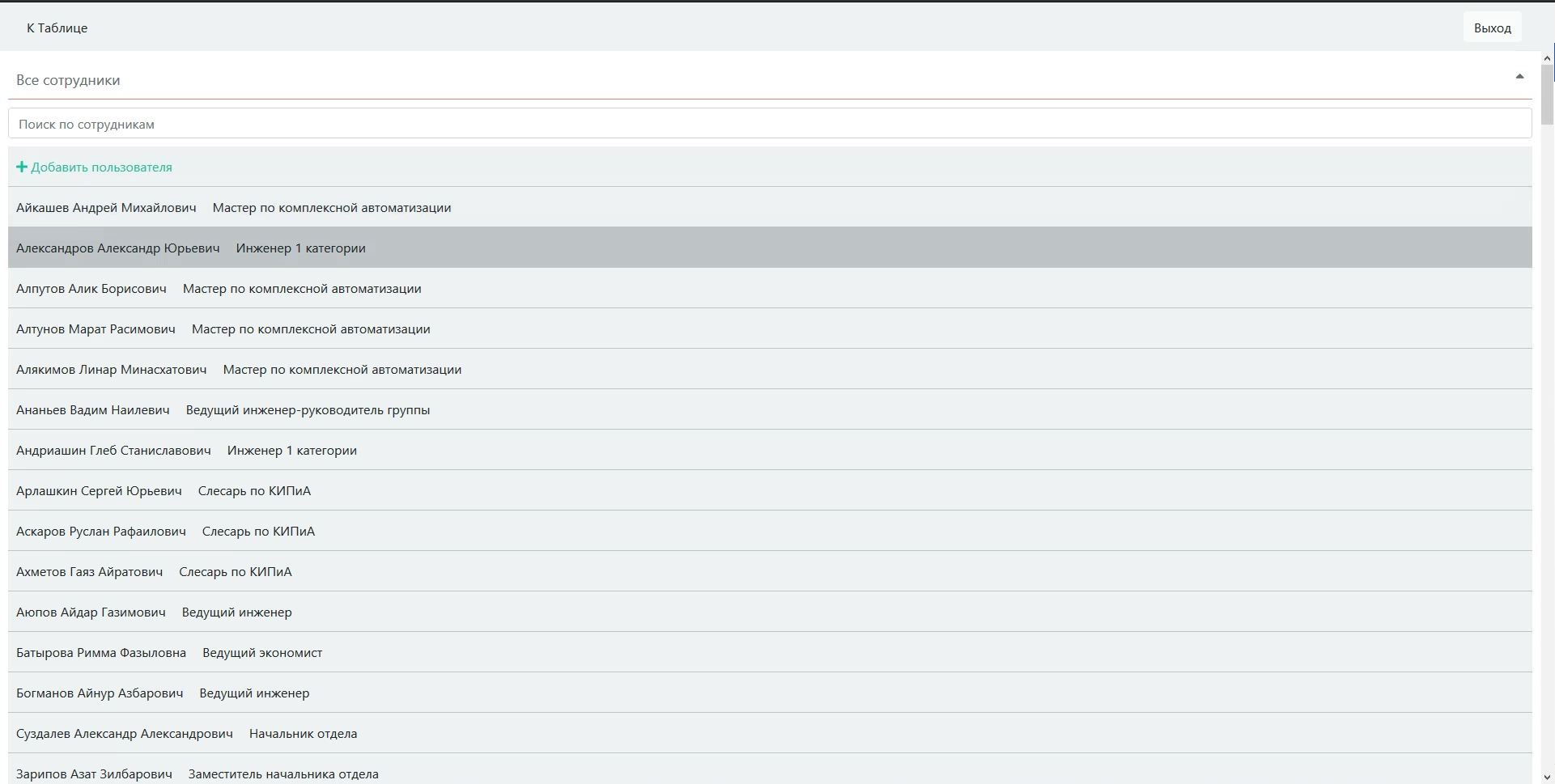


Рис. 2.6 - Форма аварий

Перейдя на вкладку «О программе», можно узнать краткое описание программы и для чего она нужна. (Рис. 2.7)

Дизайн разработан с помощью таблицы стилей CSS и языком разметки HTML:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Главная</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/auth.css">

<link rel="stylesheet" href="css/reg.css">

</head>

<body>

<div class="pagewarp">

<div class="topnav">

<a class="active" href="index.php">Главная</a>

<a id="info\_reg" href="stat.php">Расчет</a>

<a id="info\_reg" href="avar.php">Аварии</a>

<?php

if (isset($\_SESSION['id'])) {

echo "<a id='info\_reg' href='hist\_user.php'>История</a>";

}

?>

<a id="info\_reg" href="about.php">О программе</a>

<?php

if ($\_SESSION['role'] == 1) {

echo "<a href='admin/admin.php'>Админ.Панель</a>";

}

?>

<div class="search-container">

<div class = "content">

<div class="auth">

<form action="logout.php">

<?php

if (isset($\_SESSION['user'])){

echo "<button class='btn btn-dang'>Выход</button>";

}

?>

</form>

</div>

<?php

if (!isset($\_SESSION['user'])){

echo "<button class='show\_popup btn\_1 btn-dang' rel='popup1'>Авторизация</button>";

echo "<button class='show\_popup btn\_1 btn-dang' rel='popup2'>Регистрация</button>";

}

?>

</div>

</div>

</div>

<div class="overlay\_popup"></div>

<div class="popup" id="popup1">

<div class="login-block">

<div class="headline">

<p>Войдите в аккаунт</p>

</div>

<div class="inputs-block">

<input type="text" autofocus id="login" placeholder="Логин" required>

<input type="password" id="pass" placeholder="Пароль" required>

<div class="sign-buttons">

<button id="authi" type="submit" class="signIn wow slideInUp" data-wow-delay="0.4s" onclick="authorize();">Авторизация</button>

</div>

</div>

</div>

</div>

<div class="overlay\_popup"></div>

<div class="popup" id="popup2">

<div class="reg-block wow slideInDown">

<p class="headline">Регистрация</p>

<div class="inputs-block">

<div class="inputs">

<div class="leftside-inputs">

<input class="wow slideInUp" data-wow-delay="0.3s" type="text" title="Введите только латинские или русские символы, не менее трех символов" id="lastname" placeholder="Фамилия" pattern="[a-zA-Z\u0400-\u04ff]{3,30}" required/>

<input class="wow slideInUp" data-wow-delay="0.4s" type="text" title="Введите только латинские или русские символы, не менее трех символов" id="firstname" name="firstname" placeholder="Имя" pattern="[a-zA-Z\u0400-\u04ff]{3,30}" required/>

<input class="wow slideInUp" data-wow-delay="0.5s" type="text" id="email" placeholder="Email" pattern="[a-zA-Z0-9]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$" title="test@domain.com" required/>

</div>

<div class="rightside-inputs">

<input class="wow slideInUp" data-wow-delay="0.7s" title = "Введите только латинские символы" type="text" id="username" placeholder="Логин" pattern="[a-zA-Z]{3,30}" required/>

<input class="wow slideInUp" data-wow-delay="0.8s" type="password" id="password\_1" placeholder="Пароль" required/>

</div>

<div class="sign-buttons">

<button id="regist" type="submit"class="wow signUp" data-wow-delay="1s" onclick="regist();">Зарегистрироваться</button>

</div>

</div>

</div>

</div>

</div>

<!-- Main block with information about theme -->

<div class="main\_text">

<h1>Загазованность</h1>

<p>В соответствии с требованиями Межотраслевых правил по охране труда при эксплуатации водопроводно-канализационного хозяйства контроль загазованности в водопроводных и канализационных колодцах следует осуществлять с помощью газоанализаторов.</p>

</div>

<!-- Script for check on authorize user -->

<script>

var user = "<?php echo $\_SESSION['user'];?>";

if (user != '') {

check\_authorize(200);

}

else {

check\_authorize(404);

}

</script>

<!-- JQuery -->

<script src="js/jquery.js"></script>

<!-- Default Scripts -->

<script src="js/main.js"></script>

<script src='js/scripts.js'></script>

</div>

</body>

</html>

CSS:

@import url('https://fonts.googleapis.com/css?family=Roboto&display=swap');

.pagewarp {

overflow-y: hidden;

background-image: url('../images/bg.png');

/\* Full height \*/

height: 100%;

/\* Center and scale the image nicely \*/

background-position: center;

background-repeat: no-repeat;

background-size: cover;

}

html,

body {

height: 100%;

margin: 0;

padding: 5px;

}

span {

color: white;

}

td:hover {

background-color: rgb(253, 253, 253);

}

.tabledit-span {

color: black;

}

.table\_footer {

background-color: #dcdde1;

color: #2f3640;

display: none;

}

.nav-item {

margin-right: 10px

}

.nav-item:last-child {

margin-right: none

}

nav label {

color: white

}

/\* Buttons to copy \*/

button.dt-button {

display: inline-block;

font-weight: 400;

text-align: center;

white-space: nowrap;

vertical-align: middle;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

border: 2px solid transparent;

padding: .300rem .75rem;

font-size: 1rem;

line-height: 1.5;

border-radius: 10px;

transition: color .15s ease-in-out,background-color .15s ease-in-out,border-color .15s ease-in-out,box-shadow .15s ease-in-out;

}

@media screen and (max-width: 750px) {

.topnav .search-container {

float: none;

}

.topnav a, .topnav .search-container button {

float: none;

display: block;

text-align: left;

width: 100%;

margin: 0;

padding: 14px;

}

body {

font-size: 2em;

}

}

@media screen and (max-width: 991px){

.form-group ul li{

margin-left: auto;

margin-right: auto;

}

.form-group ul li input {

height: 40px;

}

.form-group {

margin-left: auto;

margin-right: auto;

margin-top: 10px;

align-items: center;

}

}

# 2.2 Разработка модуля программы

Разработка программного решения было сделано с помощью языков программирования PHP и JavaScript.

Гость имеет ограниченные права, поэтому зайдем под гостя. На сайте появляются новые вкладки (Рис. 2.7)

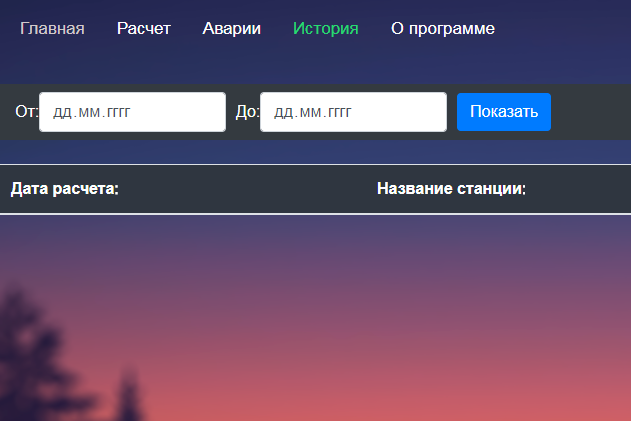


Рис. 2.8 Вкладка «Истории»

Код регистрации:

На вкладке история можно узнать результаты предыдущих расчетов.

Код на JavaScript:

function table\_hist() {

$('.openData').prop('disabled', true);

var d\_user = document.getElementById("date\_user").value;

var d\_user\_2 = document.getElementById("date\_user\_2").value;

if (d\_user == '') {

$('.openData').prop('disabled', false);

return alert('Заполните первое поле с датой');

}

var config = {

destroy: true,

bPaging: true,

ordering: true,

info: false,

dom: 'Bfrtip',

buttons: [

'copy', 'csv', 'excel', 'pdf', 'print'

],

ajax: {

url: "table\_user\_hist.php",

type: "POST",

data: {

date\_user: d\_user,

date\_user\_2: d\_user\_2,

},

complete: function() {

$('.openData').prop('disabled', false);

}

},

aoColumns: [

{mData: "date\_result"},

{mData: "name\_station"},

{mData: "analyzer"},

{mData: "al\_tag\_value"}

],

order: [

[1, 'asc']

]

}

$('.openData').prop('disabled', false);

$('.dt-button button').addClass('btn-1');

$('#example').DataTable(config);

$('#example\_filter input').addClass('form-control');

$('#example\_filter input').attr('placeholder', 'Поиск');

$('#example\_length select').addClass('form-control');

$('.table\_footer').show();

}

Он обращается к файлу php, который забирает данные из базы данных:

<?php

session\_name('logses');

session\_start();

require\_once("functions.php");

require\_once("db.php");

$conn = mysqli\_connect($server, $user, $pass, $db\_name);

$date\_user = $\_POST['date\_user'];

$date\_user\_2 = $\_POST['date\_user\_2'];

if (!$date\_user\_2){

$date\_start = $date\_user." 00:00:00";

$date\_end = $date\_user." 23:59:59";

$qur = "SELECT name\_station, gaz\_name, value, date\_result FROM raschet WHERE date\_result BETWEEN '$date\_start' AND '$date\_end' ORDER BY date\_result ASC";

}

else{

$date\_start = $date\_user." 00:00:00";

$date\_end = $date\_user\_2." 23:59:59";

$qur = "SELECT name\_station, gaz\_name, value, date\_result FROM raschet WHERE date\_result BETWEEN '$date\_start' AND '$date\_end' ORDER BY date\_result ASC";

}

$query = mysqli\_query($conn, $qur);

$data = array();

while( $row = mysqli\_fetch\_array($query) ) {

$name\_station = $row['name\_station'];

$analyzer = $row['gaz\_name'];

$value = $row['value'];

$date\_result = $row['date\_result'];

$data[] = array('date\_result'=>$date\_result, 'name\_station'=>$name\_station, 'analyzer'=>$analyzer, 'al\_tag\_value'=>$value);

}

$results = array(

"draw" => 1,

"iTotalRecords" => count($data),

"iTotalDisplayRecords" => count($data),

"aaData"=>$data);

$date\_user = '';

$date\_user\_2 = '';

echo json\_encode($results, JSON\_UNESCAPED\_UNICODE);

?>

Авторизованный пользователь имеет возможность просмотреть только свои результаты

Теперь зайдем под аккаунт администратора. Теперь у нас появилась админ.панель в которой можно редактировать, либо удалять пользователей (Рис. 2.9)

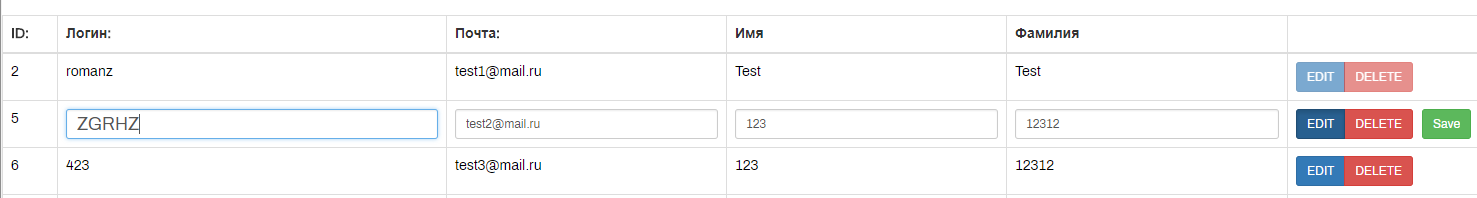


Рис. 2.9 – Редактирование и удаление в Админ.Панели

Код редактирования и удаления:

<?php

header('Content-Type: application/json');

$input = filter\_input\_array(INPUT\_POST);

$mysqli = new mysqli('localhost', 'root', '', 'gaz\_town');

if (mysqli\_connect\_errno()) {

echo json\_encode(array('mysqli' => 'Failed to connect to MySQL: ' . mysqli\_connect\_error()));

exit;

}

$id = $input['id'];

$login = $input['login'];

$email = $input['email'];

$firstname = $input['firstname'];

$lastname = $input['lastname'];

if ($input['action'] === 'edit') {

$ssql = "UPDATE `users` SET `user\_login`='$login',`email`='$email', `firstname`='$firstname', `lastname`='$lastname' WHERE `id\_user`= $id";

$mysqli->query($ssql);

} else if ($input['action'] === 'delete') {

$mysqli->query("DELETE FROM `users` WHERE `id\_user` = $id");

}

mysqli\_close($mysqli);

echo json\_encode($input);

# 2.3 Разработка печатной формы (отчета)

Отчет - документ, содержащий информацию о выполненных действиях, результатах проведённой работы. В системе все отчет проводятся при помощи MS Excel, PDF или провести на печать но есть ограничения для неавторизованного пользователя. Создавать отчеты могут только администратор и авторизованный пользователь. Для создания отчета необходимо зайти на любую вкладку с таблицами, как представлено на (Рис. 2.10):

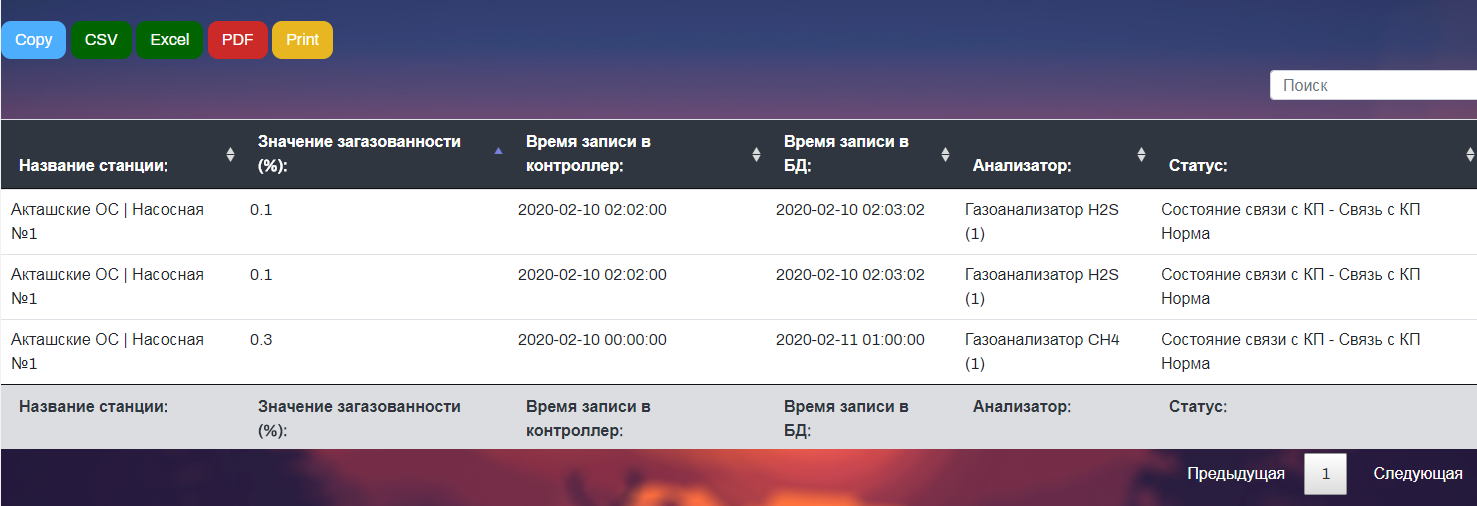


Рисунок 2.10 – Возможность создания отчета

JavaScript код для отображения таблицы и ее элементов:

function table\_result() {

$('.openData').prop('disabled', true);

var d\_user = document.getElementById("date\_user").value;

var d\_user\_2 = document.getElementById("date\_user\_2").value;

var station = document.getElementById("station").value;

var gaz = document.getElementById("gaz").value;

if (d\_user == '') {

$('.openData').prop('disabled', false);

return alert('Заполните первое поле с датой');

}

var config = {

destroy: true,

bPaging: true,

ordering: true,

info: false,

dom: 'Bfrtip',

buttons: [

'copy', 'csv', 'excel', 'pdf', 'print'

],

ajax: {

url: "table\_stat.php",

type: "POST",

data: {

date\_user: d\_user,

date\_user\_2: d\_user\_2,

station\_name: station,

gaz\_name: gaz

},

complete: function() {

$('.openData').prop('disabled', false);

}

},

aoColumns: [

{mData: "name\_station"},

{mData: "analyzer"},

{mData: "al\_tag\_value"}

],

order: [

[1, 'asc']

]

}

$('.openData').prop('disabled', false);

$('.dt-button button').addClass('btn-1');

$('#example').DataTable(config);

$('#example\_filter input').addClass('form-control');

$('#example\_filter input').attr('placeholder', 'Поиск');

$('#example\_length select').addClass('form-control');

$('.table\_footer').show();

}

# 2.4 Разработка ролей и пользователей

Система разделяет роли на администратора, авторизованного и неавторизованного пользователя. Функционал у ролей значительно различается. Для администратора имеется страница «Админ.Панель» - где имеется возможность просмотра таблицы с пользователем, редактирования и их удаления. Результат представлен на рисунке (Рис. 2.11):

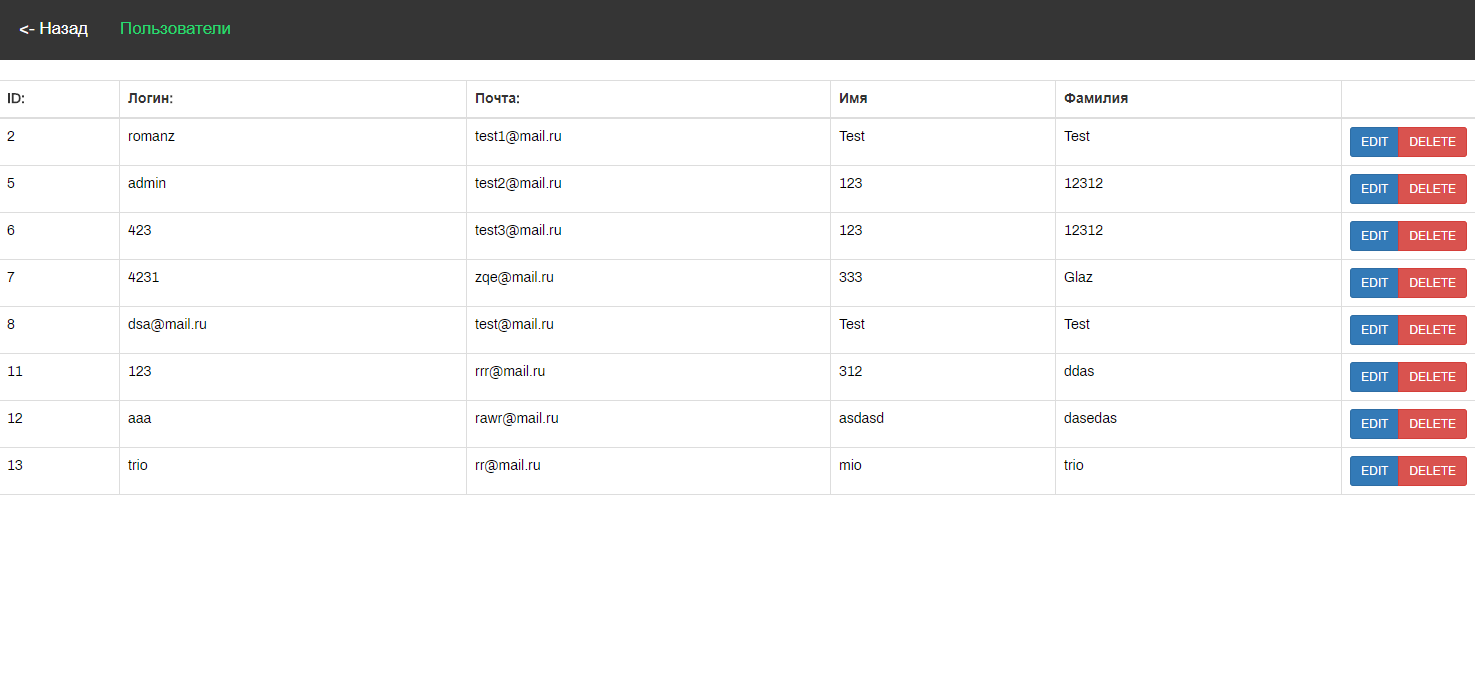


Рисунок 2.11 – Страница для администратора «Админ.Панель»

Код вызывающий таблицу:

<?php

require\_once('../db.php');

$conn = mysqli\_connect($server, $user, $pass, $db\_name);

$sql = "SELECT \* FROM users ORDER BY id\_user ASC";

$qur = mysqli\_query($conn, $sql);

if (!$qur) {echo mysqli\_error($conn);}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Админ.Панель</title>

<link rel="stylesheet" href="../css/style.css">

<link rel="stylesheet" href="../css/admin.css">

<link rel="stylesheet" href="../css/auth.css">

<link rel="stylesheet" href="../css/reg.css">

<link rel="stylesheet" href="../css/bootstrap.css">

</head>

<body>

<div class="topnav">

<a href="../index.php"><- Назад</a>

<a class="active" id="info\_reg" href="users.php">Пользователи</a>

</div>

<div class="containerr">

<br>

<div class="table-responsive">

<table id="example1" class="table table-bordered">

<thead>

<tr>

<th>ID:</th>

<th>Логин:</th>

<th>Почта:</th>

<th>Имя</th>

<th>Фамилия</th>

</tr>

</thead>

<tbody>

<?php

while ($row = mysqli\_fetch\_array($qur)) {

echo '

<tr>

<td>'.$row['id\_user'].'</td>

<td>'.$row['user\_login'].'</td>

<td>'.$row['email'].'</td>

<td>'.$row['firstname'].'</td>

<td>'.$row['lastname'].'</td>

</tr>

';

}

?>

</tbody>

</table>

</div>

</div>

<script src="../js/jquery.js"></script>

<script src='../js/jquery.tabledit.min.js'></script>

<script>

$('#example1').Tabledit({

url: 'get\_users\_admin.php',

restoreButton: false,

buttons: {

edit: {

class: 'btn btn-sm btn-primary',

html: 'EDIT',

action: 'edit'

},

delete: {

class: 'btn btn-sm btn-danger',

html: 'DELETE',

action: 'delete'

},

confirm: {

class: 'btn btn-sm btn-default',

html: 'Вы уверены?'

}

},

columns: {

identifier: [0, 'id'],

editable: [[1, 'login'], [2, 'email'], [3, 'firstname'], [4, 'lastname']]

}

});

</script>

</body>

</html>

# 3. Тестирование

# 3.1 Описание контрольного примера

Таблица 3.1 - Исходные данные к Расчету:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Ед.изм. |
|  | 01.01.2020 | Дата |
|  | 07.04.2020 | Дата |
|  | Акташские ОС | Насосная №1 | Текст |
| Газ | Газоанализатор CH4 (1) | Текст |

Таблица 3.2 - Выходные данные Расчета:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Ед.изм. |
| Название станции | Акташские ОС | Насосная №1 | Текст |
| Название газа | Газоанализатор CH4 (1) | Текст |
| Sr | 0, 3 | % |

Таблица 3.3 – Входные данные Аварии

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Ед.изм. |
|  | 01.01.2020 | Дата |
|  | 07.04.2020 | Дата |

Таблица 3.4 – Выходные данные Аварии:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Ед.изм. |
| Название станции | Акташские ОС | Насосная №1 | Текст |
| Значение загазованности | 0.1 | % |
| Время записи в контроллер | 2020-02-10 02:02:00 | Дата |

Продолжение таблицы 3.4

|  |  |  |
| --- | --- | --- |
| Время записи в БД | 2020-02-10 02:03:02 | Дата |
| Анализатор | Газоанализатор H2S (1) | Текст |
| Статус | Состояние связи с КП - Связь с КП Норма | Текст |

Таблица 3.5 – Входные данные Истории расчета

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Ед.изм. |
|  | 01.01.2020 | Дата |
|  | 07.04.2020 | Дата |

Таблица 3.6 - Выходные данные Истории расчета:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Ед.изм. |
| Дата расчета | 2020-04-07 00:31:04 | Дата |
| Название станции | Акташские ОС | Насосная №1 | Текст |
| Название газа | Газоанализатор CH4 (1) | Текст |
| Sr | 0,1 | % |

# 3.2 Тестирование системы

Широко используемыми методами тестирования являются модульное тестирование, интеграционное тестирование, приемочное тестирование, и тестирование системы. Программное обеспечение подвергается этим испытаниям в определенном порядке. 1) Модульное тестирование 2) Интеграционное тестирование 3) Системное тестирование 4) Приемочные испытания

Модульное тестирование

В первую очередь проводится модульный тест. Как подсказывает название, это метод испытания на объектном уровне. Отдельные программные компоненты тестируются на наличие ошибок. Для этого теста требуется точное знание программы и каждого установленного модуля. Таким образом, эта проверка осуществляется программистами, а не тестерами. Для этого создаются тест-коды, которые проверяют, ведет ли программное обеспечение себя так, как задумывалось.

Интеграционное тестирование

Отдельные модули, которые уже были подвергнуты модульному тестированию, интегрируются друг с другом, и проверяются на наличие неисправностей. Такой тип тестирования в первую очередь выявляет ошибки интерфейса. Интеграционное тестирование можно осуществлять с помощью подхода "сверху вниз", следуя архитектурному сооружению системы.

Приемочные испытания

Это последний тест, который проводится перед передачей программного обеспечения клиенту. Он проводится, чтобы гарантировать, что программное обеспечение, которое было разработано отвечает всем требованиям заказчика.  Есть несколько основных методов тестирования, которые формируют часть режима тестирования программного обеспечения. Эти тесты обычно считаются самодостаточными в поиске ошибок и багов во всей системе.

Тестирование методом черного ящика осуществляется без каких-либо знаний внутренней работы системы. Тестер будет стимулировать программное обеспечение для пользовательской среды, предоставляя различные входы и тестируя сгенерированные выходы. Этот тест также известен как Black-box, closed-box тестирование или функциональное тестирование.

Тестирование методом "Белого ящика", в отличие от "черного ящика", учитывает внутреннее функционирование и логику работы кода. Для выполнения этого теста, тестер должен иметь знания кода, чтобы узнать точную часть кода, имеющую ошибки.

Тестирование методом серого ящика или Graybox тестирование, это что-то среднее между WhiteBox и BlackBox тестированием, где тестер обладает лишь общими знаниями данного продукта, необходимыми для выполнения теста. Эта проверка осуществляется посредством документации и схемы информационных потоков. Тестирование проводится конечным пользователем, или пользователям, которые представляются как конечные.

Тестирование безопасности Безопасность приложения является одной из главных задач разработчика. Тестирование безопасности проверяет программное обеспечение на обеспечение конфиденциальности, целостности, аутентификации, доступности и безотказности. Индивидуальные испытания проводятся в целях предотвращения несанкционированного доступа в программный код.

Программное обеспечение проверяется на совместимость с внешними интерфейсами, такими как операционные системы, аппаратные платформы, веб-браузеры и т.д. Тест на совместимость проверяет, совместим ли продукт с любой программной платформой.

**Заключение**

Хотелось бы отметить, что данный продукт уже используется на предприятии и имеет спрос на рынке и он применяется, как и для обучения специалистов водоснабжительной системы, т.к. в нем учтен самый нужный круг решаемых задач. Для курсового проекта, это было очень даже достаточно, чтобы получить конкретные навыки расчетов в водоснабжительной области и в области программирования веб-приложений. Были получены такие навыки как:

* Программировать в соответствии с ТЗ
* Строить запросы к БД
* Изучение предметной области
* Эффективно использовать аппаратные ресурсы
* Оформлять проектную документацию
* Составлять руководство пользователя
* Использование новых инструментов разработки сайтов

**Литература**

Учебная

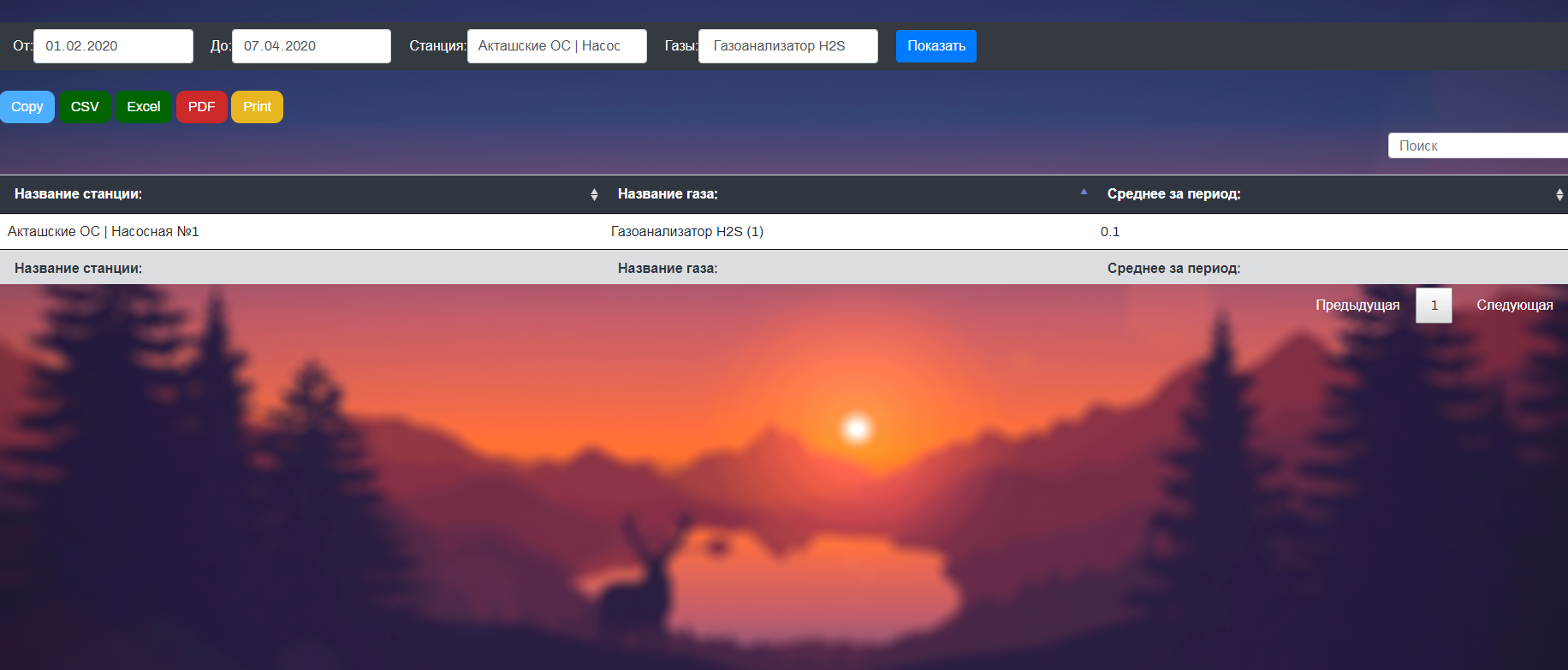
1. Филиппов В.В. – Технологические трубопроводы и трубопроводная арматура, 2015. – 521с.
2. ЧакМаскиано, Бил КеннедиHTML и XHTML. Подробное руководство/ пер.: Сергей Иноземцев. Учебник. – М.:Символ-Плюс, 2014. – 1052с.
3. Кристофер Шмитт CSS. Рецепты программирования/ пер.: Татьяна Коротяева. Руководство. – М.: Русская Редакция, БХВ-Петербург, 2015. – 705с.
4. Эд Титтел, ДжеффНоублHTML, XHTML и CSS для чайников/ пер.: А. Сергеев. Учебник. – М.: Вильямс, 2013. – 400с.
5. [Анар Бабаев](https://www.ozon.ru/person/6293923/), [Николай Евдокимов](https://www.ozon.ru/person/3043329/) и др. Создание сайтов. Учебник. – М.: Питер, 2014. – 304с.

# ПриложениеA

**Руководство пользователя**

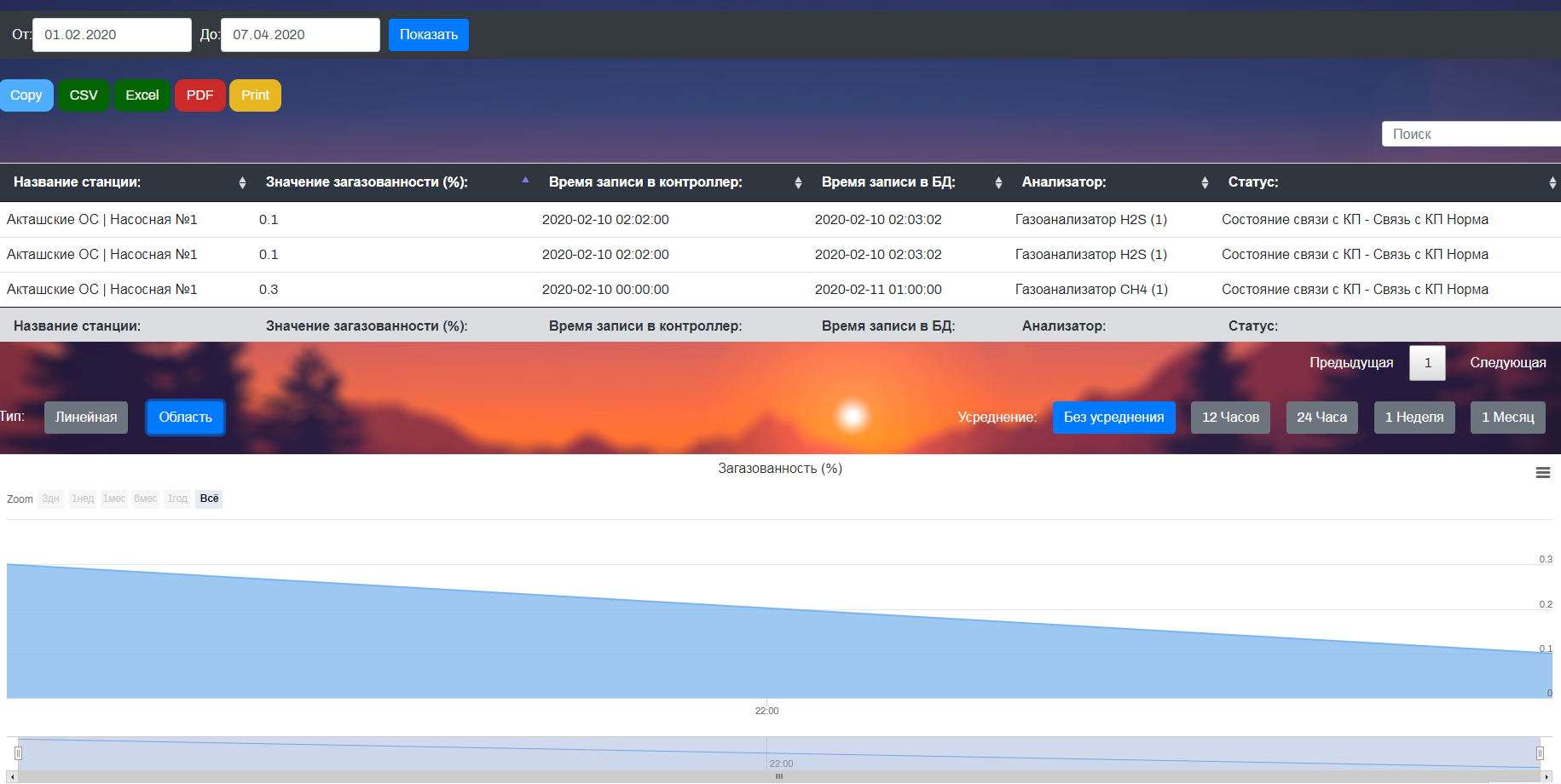
Перед началом работы с приложением, пользователю следует для начала зарегистрироваться или авторизоваться под существующим аккаунтом.

После чего система будет сохранять расчеты этого пользователя, и он сможет экспортировать полученные данные в Excel, PDF или распечатать. Выбираем определенный промежуток времени, выбираем нужную станцию для расчета и ее газоанализатор и нажимаем на кнопку «Показать»



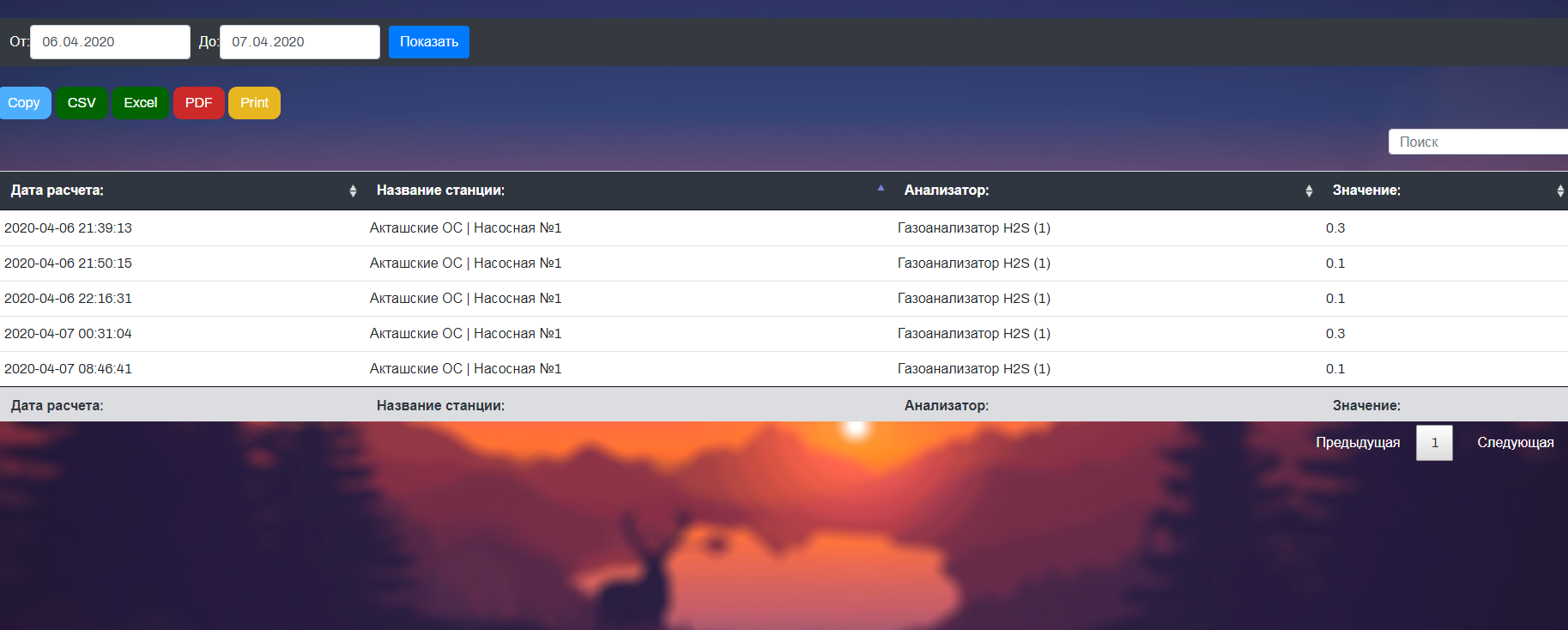
Приложение 1. Страница расчетов

В следующей странице, можно узнать конкретное состояние трубопровода, процент загазованности и состояние газоанализатора и график загазованности. Выбираем определенный промежуток времени и нажимаем на кнопку «Показать»



Приложение 2. Страница аварий

На следующей странице можно узнать результаты расчетов. Выбираем определенный промежуток времени и нажимаем на кнопку «Показать»



Приложение 3. Страница расчетов

# ПРИЛОЖЕНИЕ C

**Электронный носитель с программой и пояснительной запиской**