

1 Общее описание

1.1 Задача дипломной работы

Задача оперативного управления проектными заданиями встает в организациях самых различных типов: коммерческих и государственных, производственных и научных, в самых разнообразных сферах деятельности, поскольку в условиях жесткой конкуренции все предприятия стремятся повысить производительность своей работы путем эффективного распределения обязанностей среди сотрудников. При этом руководитель должен учитывать общую загруженность работников при выполнении ими должностных функций и обеспечить контроль выполнения заданий.

Повышение качества и эффективности управления - одна из главных задач большинства компаний. Качественное управление открывает путь к реальному повышению конкурентоспособности и прибыльности, к привлечению инвестиций, а значит, к максимальным темпам роста стоимости бизнеса. Для повышения качества управления менеджменту компании необходимы современные и многофункциональные инструменты, которые позволяют анализировать большое количество экономических параметров и принимать грамотные управленческие решения, адекватные сложившейся ситуации на рынке.

Автоматизация проектных задач является одной из главной целью в работе как коммерческих так и государственных организаций, в условиях жесткой конкуренции все предприятия стараются повысить производительность работы путем распределения задач между сотрудниками. Всю работу должен контролировать директор, который может управлять загруженностью, а так же контролировать выполнение заданий.

Система управления проектами - компьютерное приложение для организации распределения заданий между сотрудниками и их группами. Системы управления проектами позволяют создавать и изменять задачи, контролировать ход их выполнения, поддерживают различные варианты отчетности.

Основные задачи проекта которые мы должны выполнить:

- Проанализировать аналогичные систему управления;
- Настроить распределение задач между сотрудниками;
- Разделить права пользователей;
- Создать групповые работы;
- Создать поисковик;
- Создать хранилище выполненных работ;
- Управление своей страницы каждым пользователем;
- Создание единого интерфейса;

1.2 Цель

Разработка автоматизированной системы должно соответствовать следующим требованиям:

- Использование системы в локальной среде;
- Возможность работы на любой из распространенных систем;
- Хорошая производительность на старых компьютерах;
- Простота в использовании системы;
- Возможности анализа эффективности работы.

1.3 Термины и сокращения

Термины и сокращения, которые используются в пояснительной записке к дипломному проекту приведены ниже в таблице 1.

Таблица 1 – Термины и сокращения

БД	База данных
СУБД	Система управления базами данных
ERP	Enterprise resource planning
SQL	Structured query language
PHP	Personal HomePage
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
UML	Unified Modeling Language
ER	Entity Relationship
JavaScript	Объектно-ориентированный язык программирования
URL	Уникальный адрес электронного ресурса
WEB приложение	Приложение с которым клиент взаимодействует через браузер

Продолжение Таблицы 1

Кэш	Буфер с информацией которая вероятнее всего будет использоваться
PhpStorm	Среда разработки ПО
ПО	Программное обеспечение
Фреймворк	Платформа программы с библиотеками

2 Специфика автоматизированных систем и их требования

2.1 Потребности малых организаций в автоматизации бизнес-процессов

Большую часть представителей класса ERP-решений относят к так называемым «тяжелым» системам по причине широты класса решаемых задач, богатой функциональности, высокой стоимости решения и владения, долгих сроков внедрения. Для крупных предприятий, промышленных и сельскохозяйственных агломераций, дистрибьюторских сетей выбор «тяжелой» системы зачастую оправдан: временные, финансовые и трудовые вложения окупаются за счет упорядочения бизнес-процессов компании, повышения эффективности управления ресурсами, упорядочения бухгалтерского, управленческого, финансового и других видов учета.

Однако среди небольших предприятий, относящихся к так называемому «малому» бизнесу и активно работающих с информацией, также велика потребность в упорядочении и автоматизации протекающих информационных процессов.

Примерами компаний такого типа, в частности, являются: небольшие организации, поставляющие товары или оказывающие профессиональные услуги, консультационные компании, переводческие и копирайт-агентства. Отличительными чертами таких компаний являются:

- значимая роль информационных процессов в управлении компанией и разнообразие видов данных подлежащих хранению и учету;
- небольшое количество сотрудников (до 100);
- ограниченное число уровней управления процессами (обычно не более 2) и их участников (работа малыми группами);
- невысокая сложность бизнес-процессов;
- простота бухгалтерского учета (зачастую применение упрощенной системы налогообложения);

С точки зрения протекающих процессов и принципов управления можно охарактеризовать рассматриваемый класс задач как проектный. Как правило, есть четко выделенные этапы, сроки, результаты, поэтапная приемка работ, сдельно-премиальная система поощрения, ответственное лицо на каждом этапе.

2.2 Системы управления проектами и задачами

Система должна включать в себя комплекс программного обеспечения, в котором будет планирование задач, управление групповыми работами,

административные системы и контроль выполнения задач.

Основными требованиями в управлении системы будут:

- Создание задач;
- Создание групповых работ;
- Сортировка по уровню сложности;
- Установка дедлайнов;
- Управление несколькими задачами одновременно;
- Программное обеспечение может предоставлять различную информацию;
- Информация о сотрудниках;
- Поставленные задачи;
- Сроки выполнения задач;
- Информация о выполненной работе;
- Информации о группах и их нагрузке.

2.3 Требования к системе

Сам факт того, что системы управления задачами являются «легкими» по отношению к полноценным ERP-системам, обладают урезанным функционалом и простотой освоения не избавляет их от недостатков данного семейства ИС.

В малых организациях особенно остро стоит вопрос с ценой и временем внедрения, т.к. они не могут себе позволить крупные долгосрочные инвестиции. Кроме того, системы для малых организаций должны быть ориентированы на уже имеющуюся сетевую архитектуру, аппаратное обеспечение организации, IT-инфраструктуру. Это означает поддержку наиболее распространенных серверных платформ, которыми на данный момент являются Windows и Unix-подобные системы, возможность использования не на профессиональном серверном оборудовании, интеграцию с распространенными системами и поддержку популярных форматов документов.

Снижение затрат на внедрение возможно при использовании системой популярных открытых технологий за счет последующей доработки сторонними специалистами. Уменьшение стоимости переобучения сотрудников может быть достигнуто при использовании графического интерфейса, основанного на широко распространенных технологиях и принципах, например веб-интерфейса.

Исходя из вышеизложенного, к системе можно предъявить следующие требования:

Функциональные

- Список задач – возможность создания задач с описанием, набором характеристик и сроками выполнения, а также отображение списка задач для данного сотрудника;

- Информирование о новых событиях – своевременное отражение в системе информации об изменениях, связанных с задачами сотрудника;
- Иерархия задач – возможность хранения и представления задач в виде иерархического дерева. Возможность декомпозиции задач на подзадачи с их делегированием;
- Комментирование – обсуждение задач в системе, с сохранением истории обсуждения;
- Теги – назначение задачам неограниченного числа меток - ключевых слов с возможностью поиска по тегу и набору тегов;
- Поиск – реализация поиска по параметрам задачи, ответственным лицам, описанию и т.д.;
- Хранение внешних контактов – возможность хранения в системе любых важных рабочих контактов и прикрепления их к конкретной задаче для ускорения взаимодействия;
- Разделение доступа – назначение ролей по задаче и ограничения доступа на их основе;
- Хранение файлов – прикрепление к задачам связанных с ними файлов;
- Планирование – возможность менеджеру вести в системе учет и прогнозирование рабочего времени сотрудников (в т.ч. отпуска и прочее), и других ресурсов;
- Аналитика – сбор статистики и анализ эффективности сотрудников, их загруженности в данный момент;
- Требования к развертыванию, настройке, интеграции и эксплуатации
- Модульность, расширяемость – установка только модулей с необходимыми функциями, добавление функций при необходимости;
- Открытые технологии – для обеспечения возможности доработки системы
- Интеграция с имеющейся IT-инфраструктурой – например, авторизация в системе через домен организации, переносимость документов между системами и т.д.;
- Настраиваемость под структуру компании – возможность внедрения системы без значительной переработки под конкретную компанию;
- Логирование изменений и резервное копирование – устойчивость системы к потере и ошибочному изменению информации, возможность отката к неповрежденному состоянию;
- Мультиплатформенность – возможность развертывания на большинстве распространенных операционных систем;
- Работа удаленных пользователей – возможность работы в системе вне офиса;
- Интерфейс пользователя – понятный и удобный графический интерфейс, основанный на распространенных технологиях.

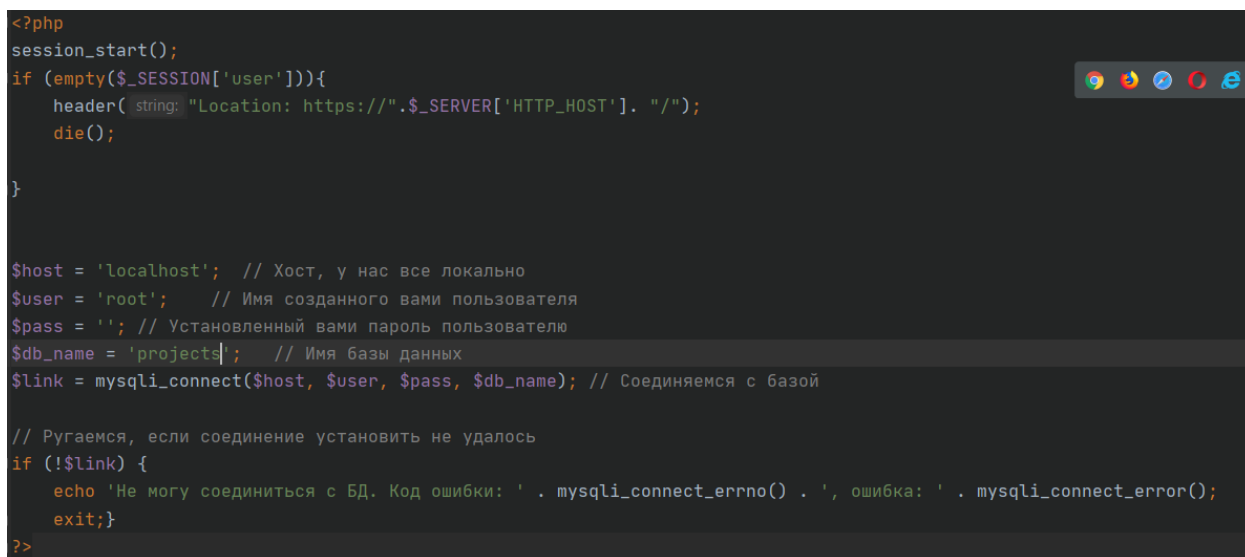
3 Проектирование и разработка автоматизированной системы управления проектами

3.1 Выбор инструментальных средств разработки

3.1.1 Php

Php – это среда разработки, которая служит для написания кода. В данной дипломной работе мы будем использовать такие языки как PHP, CSS, JavaScript, HTML. Во всем этом нам поможет PhpStorm, это компилятор который может использовать в себе все вышеперечисленные языки программирования. Для запуска кода мы будем использовать OpenServer, это программа которая создает локальный сервер с помощью которого мы будем открывать нашу работу.

Php направлено строго на написание скриптов, которые будут запущены на сервере. Php широко используется в написании веб-серверов и браузеров. Пример написания кода представлен на 3.1



```
<?php
session_start();
if (empty($_SESSION['user'])) {
    header( string: "Location: https://" . $_SERVER['HTTP_HOST'] . "/" );
    die();
}

$host = 'localhost'; // Хост, у нас все локально
$user = 'root';      // Имя созданного вами пользователя
$pass = '';          // Установленный вами пароль пользователю
$db_name = 'projects'; // Имя базы данных
$link = mysqli_connect($host, $user, $pass, $db_name); // Соединяемся с базой

// Ругаемся, если соединение установить не удалось
if (!$link) {
    echo 'Не могу соединиться с БД. Код ошибки: ' . mysqli_connect_errno() . ', ошибка: ' . mysqli_connect_error();
    exit;
}
?>
```

Рисунок-3.1 Наглядный пример кода PHP

3.1.2 HTML

HTML – является главным языком в написании сайта, так как это и есть страничка сайта, он используется для того чтобы браузер понимал что нужно ему

показать. В основном код HTML состоит из тегов, с помощью них мы можем понять что нужно на веб-страничке. 3.2

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="ATU UIT">
  <link rel="shortcut icon" href="favicon.ico">
  <title>Главная</title>
```

Рисунок-3.2 - Пример тегов HTML

3.1.3 CSS

CSS является неотъемлемой частью в создании сайтов. CSS служит для определения стилей в оформлении сайта. Оно включает в себя оформление дизайна, создание макетов, управление картинками либо же гифками. CSS имеет много преимуществ, такие как: легкое в написании, простота в использовании, управлять можно всей страничкой, теги не дублируются. 3.3

```
body {
  height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  font-family: Montserrat, sans-serif;
}
```

Рисунок-3.3 – CSS пример интерфейса

3.1.4 JavaScript

JavaScript это скрипты которые оживляют веб-страницу. Скрипты не нуждаются в подготовке для компиляции, они просты в написании. JavaScript

дает страничке такие функции как: реагирование на клик мышки и клавиатуры, получение и создание куки, скачивать и загружать документы, перемещение меж страницами. Принцип написания JavaScript прост 3.4

```
function saveAll() {  
    startLoadingAnimation();  
    var classname = arguments[0];  
    let info = {};  
    for (i = 1; i < arguments.length; i++) {  
        info[arguments[i]] = $('#' + arguments[i]).val();  
    }  
    console.log(info);  
    $.ajax({  
        type: "POST",  
        url: "controller.php",  
        data: {  
            func: "saveAll",  
            classname: classname,  
            info: info  
        },  
        cache: false,  
        success: function (response) {  
            stopLoadingAnimation();  
            $("#result").fadeIn(1500).html(response);  
        }  
    });  
}
```

Рисунок-3.4 – JavaScript

3.1.5 MySQL

MySQL – широкоиспользуется для создания и использования базы данных. База данных состоит из таблиц, ячеек и строк. В эти ячейки вводятся данные, которые в последствии будут использованы в системе.

База данных нам нужна для того чтобы записывать туда всю информацию о пользователях, задачах и т.д. Каждую информацию нужно будет вносить в таблицу, а при добавлении чего-то нового, нужно будет создавать новые строки либо же столбцы.

MySQL не единственная среди баз данных, только она одна из простых и легких в изучении а так же в создании и редактировании. Незначительные изменения будут занимать не больше 30 минут, так как в написании кода нет

ничего лишнего. Так же SQL дает гарантию безопасности данных. 3.5

+ Параметры

		id	full_name	login	email	password	role	position
☐		26	admin	admin	admin@gmail.com	admin	1	admin
☐		27	Мун Артур	Test.Test	NULL	aaa	2	Исполняющий
☐		31	test 1	test1	NULL	test1	2	Исполняющий
☐		32	test2	test2	NULL	test2	3	Директор
☐		33	Robert Einshtein	test3	NULL	test3	3	Директор

↑ ☐ Очистить все ☐ Сортировать по

Рисунок-3.5 – Пример таблицы

3.1.6 OpenServer

OpenServer является профессиональным инструментом для веб-разработок. Он создает локальный сервер с помощью которого можно будет запускать наши разработки. Один из главных плюсов данного приложения является то что на нем можно открывать PhpMyAdmin, а так же и выбирать версию SQL. Само приложение имеет простой интерфейс. 3.6.

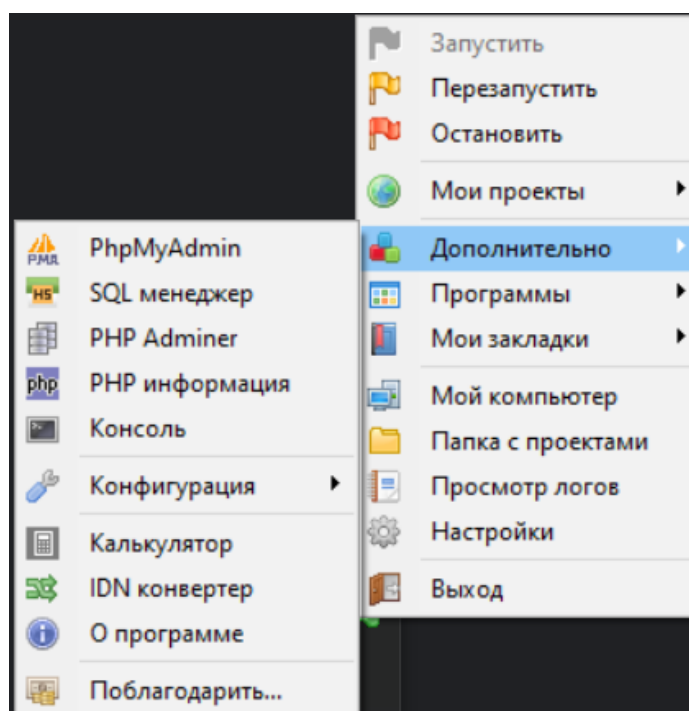


Рисунок-3.6 – Интрефейс OpenServer

3.1.7 PhpMyAdmin

Интерфейсом для создания базы данных SQL служит PhpMyAdmin. Это приложение которое запущено на локальном сервере которое служит для:

- Создание и редактирование базы данных
- Управление и добавление новых строк и столбцов в базах данных
- Можно выполнять запросы SQL
- Возможно загружать диаграммы, графики
- Управлять паролями и ключами

Само веб-приложение PhpMyAdmin мы запускаем с помощью OpenServer. Устанавливаем и подключаем СУБД, а далее при запуске сервера заходим на страничку локального сервера и приступаем к работе. Интерфейс очень прост в использовании. 3.7

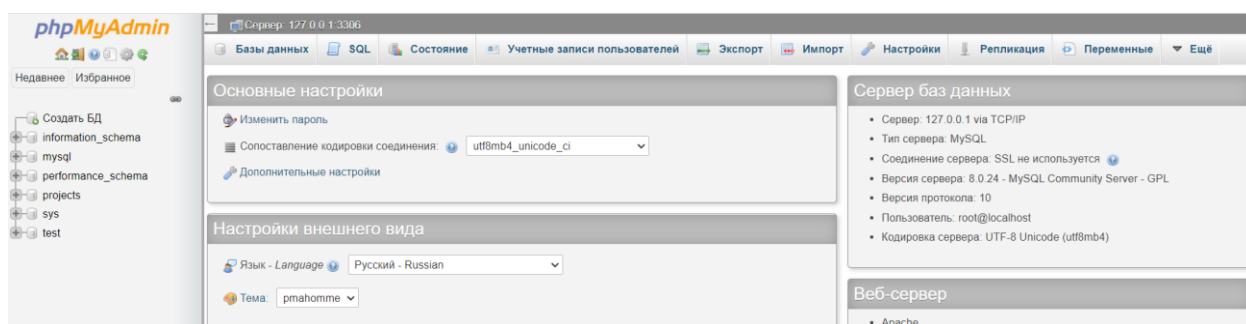


Рисунок-3.7 – Интерфейс PhpMyAdmin

3.1.8 PhpStorm

PhpStorm это среда разработки для PHP, разработанная компанией JetBrains. А так же эта среда разработки подойдет и для нашей системы так как можно использовать так и CSS, HTML, JavaScript. Данный продукт может быстро просматривать результаты и в случае неудачи выделяет неверные строки, а так же указывает на ошибки. Имеет в себе полный набор инструментов для фронтенд разработки, а так же простой и легко доступный интерфейс с которым справиться каждый. 3.8

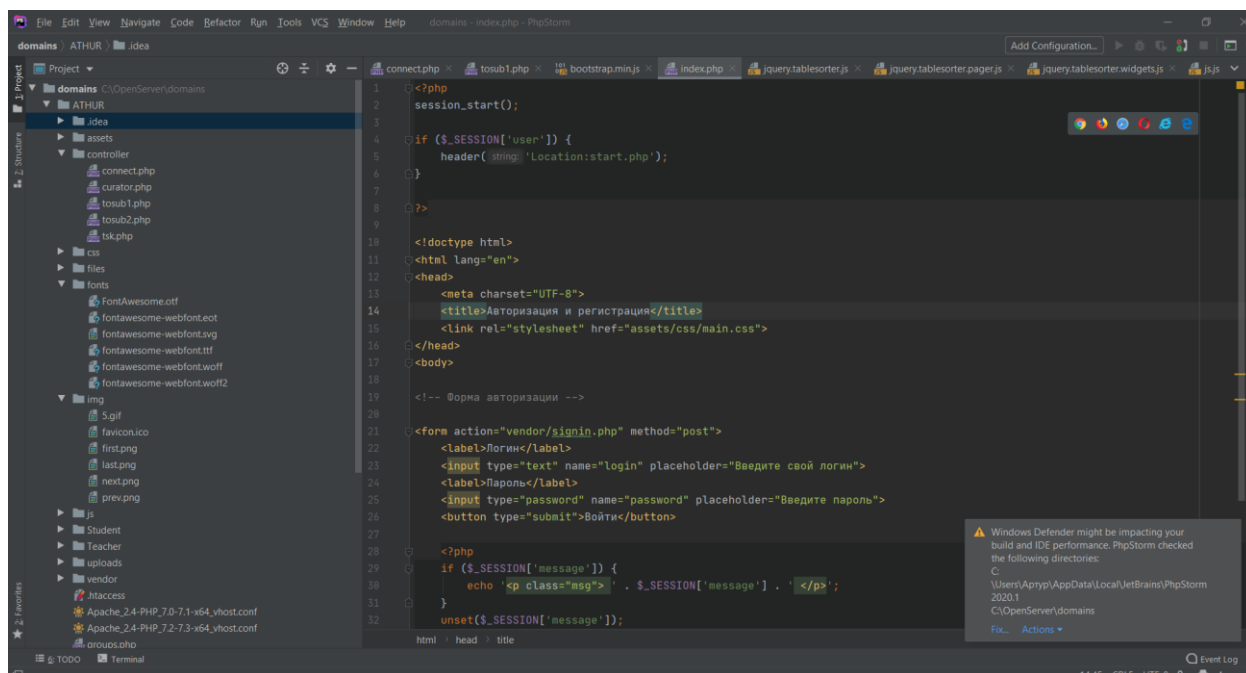


Рисунок-3.8 – Интерфейс PhpStorm

3.2 Разработка приложения

3.2.1 Создание базы данных с помощью PhpMyAdmin

На данном этапе работы мы будем создавать базу данных для в последующем использовании его в нашем проекте. База данных является важнейшей частью в создании проекта.

Начнем с того что запустим OpenServer и откроем PhpMyAdmin. Создаем новый документ под название project, в котором в последствии будем хранить все наши данные и начнем создание таблицы. 3.9

Имя таблицы: Добавить поле(я) Вперед

Имя	Тип	Длина/Значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	А_И	Комментарии
	INT		Нет			☐	---	☐	
	INT		Нет			☐	---	☐	
	INT		Нет			☐	---	☐	
	INT		Нет			☐	---	☐	

Структура

Комментарии к таблице: Сравнение: Тип таблиц:

Рисунок-3.9 – Процесс создания таблицы

Как мы можем видеть мы создали таблицы в которые будем вносить

данные чтобы в дальнейшем их использовать. Начнем с того что создадим пользователей и присвоим каждому свой ID и добавим к ним описание. В последствии мы получим следующие данные. 3.10

SELECT * FROM `users`

Профилировать

Показать все

Количество строк: 50

Фильтровать строки:

Поиск в таблице

Параметры

	id	full_name	login	email	password	role	position
☐	26	admin	admin	admin@gmail.com	admin	1	admin
☐	27	Мун Артур	Test.Test	NULL	aaa	2	Исполняющий
☐	31	test 1	test1	NULL	test1	2	Исполняющий
☐	32	test2	test2	NULL	test2	3	Директор
☐	33	Robert Einshtein	test3	NULL	test3	3	Директор

Рисунок-3.10 – Полученные данные

Далее мы создадим еще несколько таблиц. Одной из таких будет группы. Группы в которые мы будем добавлять пользователей и присваивать им каждому задание. 3.11.

Отображение строк 0 - 2 (3 всего, Запрос занял 0,0145 сек.)

SELECT * FROM `user\_group`

Показать все | Количество строк: 50 | Фильтр

Параметры

	id	user_id	group_id
☐	42	27	9
☐	43	31	8
☐	44	32	13

Рисунок-3.11 – Создание остальных таблиц

Следующим шагом будет создать задания и их дедлайны. Задания мы связываем с группами, а дедлайны с заданиями. В окончании таблица будет выглядеть вот так. 3.12

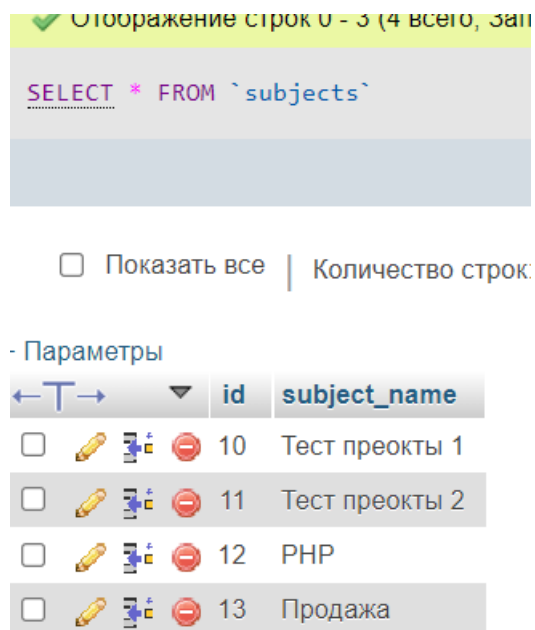


Рисунок-3.12 – Создание таблиц

После я создал пару таблиц связующих, такие как назначение руководителя, дедлайны, процент выполнения задач, новости. Все было связано с каждым пользователем. В окончании мы имеем вот такое количество таблиц представленные на 3.13

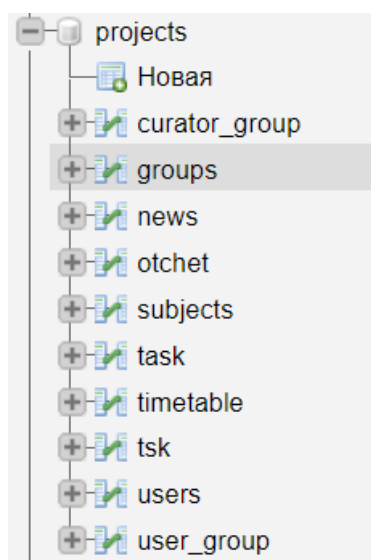


Рисунок-3.13– Окончательная база данных

Полная ER диаграмма данного БД представлена на 3.14

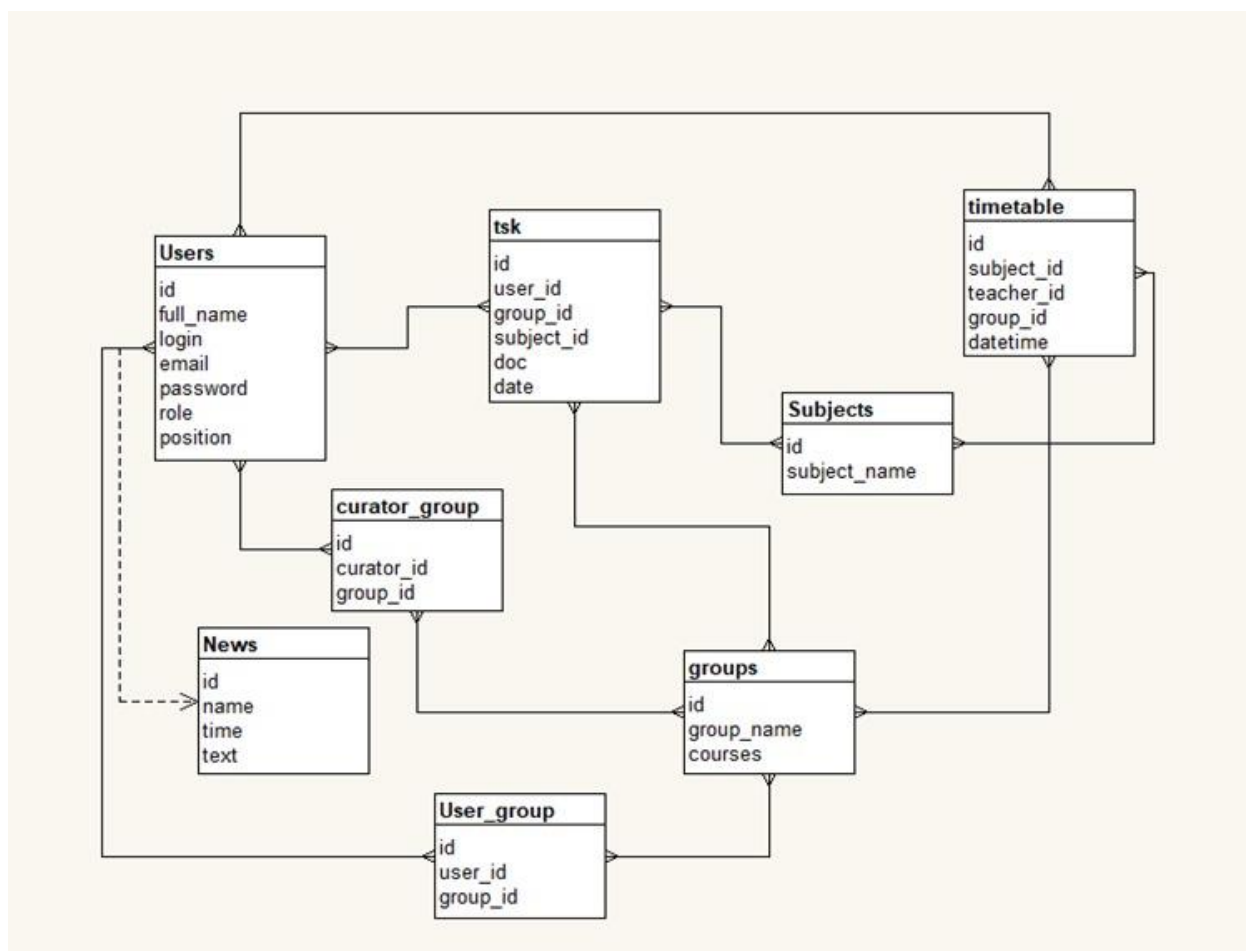


Рисунок-3.14– ER диаграмма нашей базы данных

3.2.2 Разработка графического интерфейса

Одной из задач дипломного проекта является разработка пользовательского графического интерфейса для удаленного доступа к данным системы. Ключевой составляющей удобства пользовательского интерфейса является продуманная система навигации. От того, насколько проста, понятна и удобна система навигации, во многом зависит качество работы сотрудников предприятия. В системе отображается большое количество информации, которое необходимо для работы сотрудникам предприятия, и немаловажным является то, где и в каком виде информация представлена.

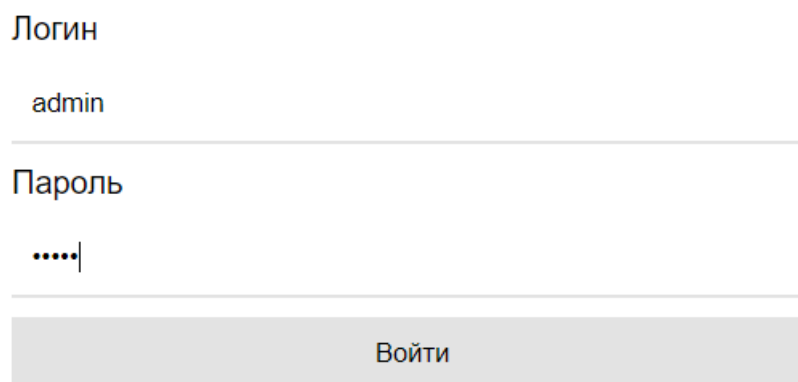
Главными факторами, которыми должен обладать интерфейс это:

- Простота в использовании
- Возможность перемещения между сайтами за один клик

- Визуальное выделение элементов в системе навигации
- Упрощение стандартов

3.2.3 Примеры графического интерфейса

В системе существует три основных типа пользователей, в зависимости от которых, после авторизации, будут предложены страницы различного вида. Администратор системы с предустановленной учетной записью `admin` имеет доступ к редактированию словарей системы, добавлению и редактированию пользователей, ручному запуску служебных скриптов. Интерфейс пользователя-администратора представлен на. 3.15



The image shows a login form with two input fields and a button. The first field is labeled 'Логин' (Login) and contains the text 'admin'. The second field is labeled 'Пароль' (Password) and contains five dots, indicating a masked password. Below the password field is a grey button with the text 'Войти' (Login).

Рисунок-3.15 – Начальная страница

Рядовой сотрудник с правами пользователя имеет доступ с правами просмотра к большинству справочной информации в системе, а так же списку своих задач, с возможностью их фильтрации и сортировки. Интерфейс пользователя с логином `Test1` представлен на 3.16

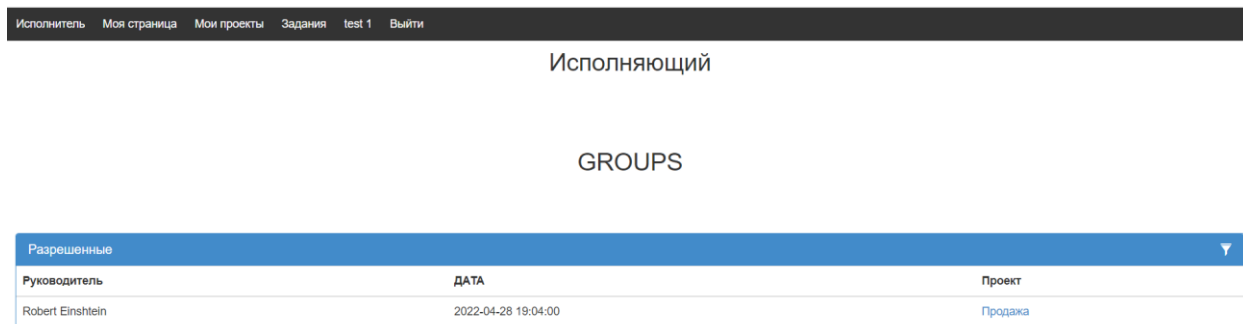


Рисунок-3.16 – Главный интерфейс исполняющего

Пользователь с правами главы подразделения одновременно имеет как доступ к функциям обычного сотрудника, так и к разделу управления проектами. Интерфейс пользователя – главы подразделения с логином test2 представлен на 3.17

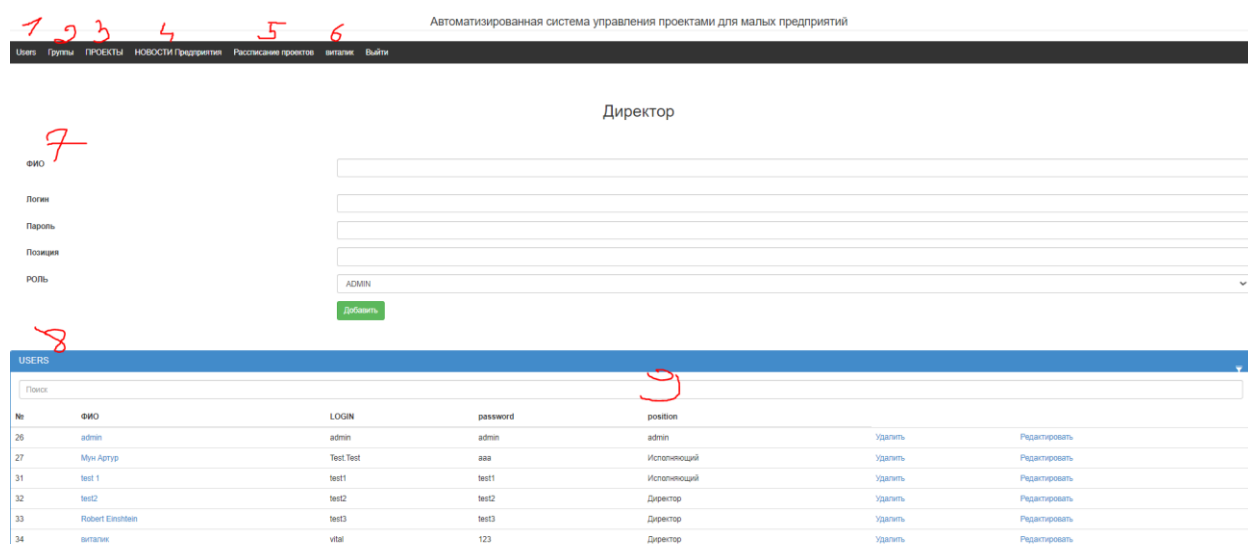


Рисунок-3.17 – Интерфейс Админа

Описание интерфейса по Рисунок 3.17:

- 1) Пользователи
- 2) Группы
- 3) Проекты
- 4) Новости
- 5) Расписания проектов
- 6) Личный кабинет
- 7) Добавление новых пользователей
- 8) Список пользователей и их логины пароли
- 9) Позиции пользователей

3.2.4 Составление кода программы

Представил пару скринов которые сделал в процессе написания кода 3.18 - 3.19

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Авторизация и регистрация</title>
  <link rel="stylesheet" href="assets/css/main.css">
</head>
<body>

<!-- Форма авторизации -->

<form action="vendor/signin.php" method="post">
  <label>Логин</label>
  <input type="text" name="login" placeholder="Введите свой логин">
  <label>Пароль</label>
  <input type="password" name="password" placeholder="Введите пароль">
  <button type="submit">Войти</button>

  <?php
  if ($_SESSION['message']) {
    echo '<p class="msg">' . $_SESSION['message'] . '</p>';
  }
  unset($_SESSION['message']);
  ?>
</form>

</body>
</html>
```

Рисунок-3.18 - Создание начальной страницы входа

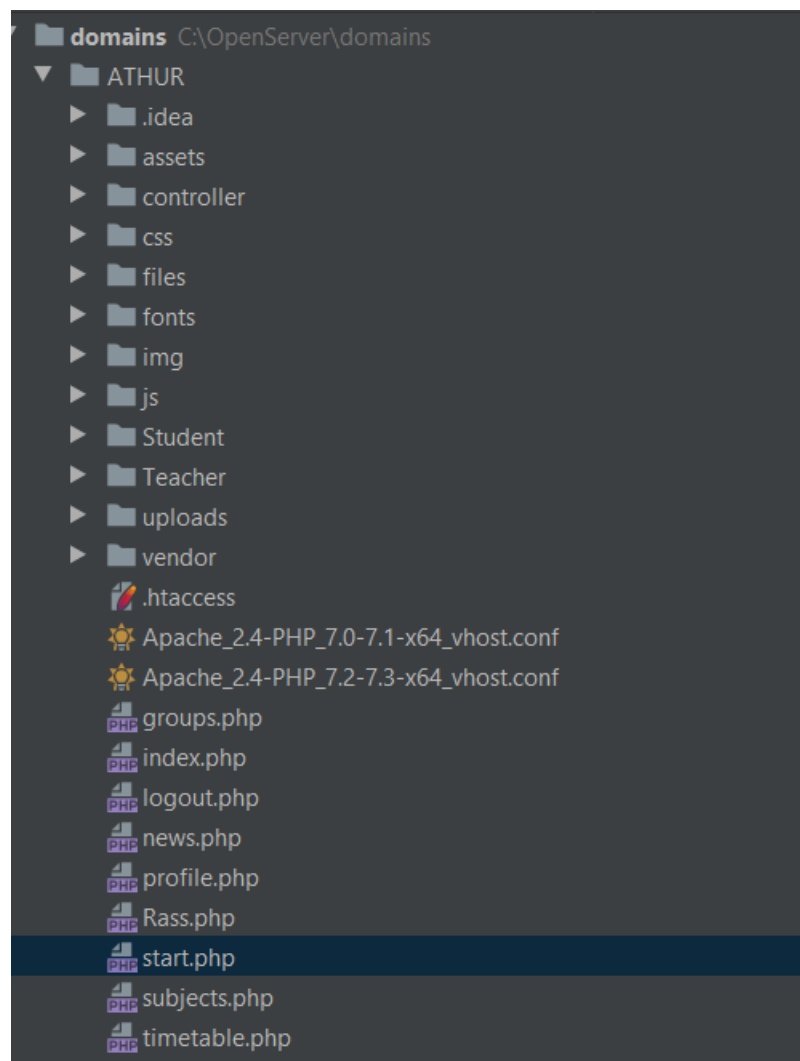


Рисунок-3.19- Полученные файлы разработанные в процессе создания сайта

ЗАКЛЮЧЕНИЕ

В данном дипломном проекте рассмотрены потребности малых организаций в автоматизации систем малых предприятий. На основе изученных материалов сделан вывод, что основные бизнес-процессы малых организаций носят проектный характер. Типом систем, наилучшим образом удовлетворяющим автоматизации данного вида процессов являются системы управления проектами.

Рассмотрены представленные на рынке системы управления проектами, сделан вывод о недостаточной реализации в них функций планирования и распределения ресурсов. Выдвинуты требования к разрабатываемой системе, обоснована необходимость разработки.

Изучены методологии управления проектами. Обоснован выбор метода критической цепи в качестве основы подсистемы распределения ресурсов.

Рассмотрены основные принципы организации разрабатываемой системы, обоснован выбор инструментария разработки системы: СУБД MySQL, языка сценариев Php, средств проектирования и разработки PHPStorm, phpMyAdmin, технологий разработки пользовательского интерфейса HTML, CSS.

На основе выдвинутых требований к функциональности системы, разработаны проект системы в виде функциональных схем. Разработаны логическая и физическая модель базы данных, приведено полное описание сущностей БД и их атрибутов. Разработаны прототипы интерфейсов для пользователей с разными привилегиями. Приведен пример их использования.

По результатам проделанной работы, можно сделать вывод, что поставленные задачи решены и цель данного дипломного проекта можно считать достигнутой.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Алан Бьюли "Изучаем SQL" (2007) <https://proglib.io/p/sql-digest>
- 2 HTML и CSS. Разработка и дизайн веб-сайтов <https://monster-book.com/html-i-css-razrabotka-i-dizayn-veb-saytov>
- 3 HTML Учебник <https://schoolsw3.com/html/>
- 4 Руководство по PHP <https://www.php.net/manual/ru/tutorial.php>
- 5 Node.js Web Development – 4 издание (2018) <https://library-it.com/web/javascript-web/>
- 6 Chatbots 101 // Электронная версия
[//www.oracle.com/us/technologies/mobile/chatbot-.pdf](https://www.oracle.com/us/technologies/mobile/chatbot-.pdf)
- 7 Архитектура SQLite // Электронная версия на
сайте <https://russianblogs.com/article/1781106567/>
- 8 Написание технического задания // Электронная версия на сайте
<https://texterra.ru/blog/kak-sostavit-gramotnoe-tekhzadanie-na-razrabotku-sayta.html>.
- 9 Язык стилей CSS // Электронная версия на сайте
https://skillbox.ru/media/code/chto_takoe_css/.
- 10 Чистая архитектура web-приложения // Электронная версия на сайте
<https://habr.com/ru/post/493430/>.
- 11 <https://www.w3schools.com/css/default.asp>
- 12 https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- 13 Панель администратора // Электронная версия на сайте
<https://proproprogs.ru/django/nachinaem-rabotu-s-admin-panelyu>.
- 14 Оптимизация сайта // Электронная версия на сайте
<https://proproprogs.ru/django/optimizaciya-sayta-s-django-debug-toolbar>.
- 15 MySQL и phpMyAdmin: включение сервера и доступ к его
администрированию // Электронная версия на сайте
<https://support.qnap.ru/hc/ru/articles/360000721274>
- 16 Топ 10 лучших редакторов HTML // Электронная версия на
сайте <https://itvdn.com/ru/blog/article/top10-html>
- 17 Что такое CMS // Электронная версия на сайте
<https://www.unisender.com/ru/blog/idei/cms/>
- 18 Почему JavaScript пожирает HTML // Электронная версия на сайте
<https://habr.com/ru/company/ruvds/blog/492754>
- 19 UML диаграммы // Электронная версия на сайте
<https://evergreens.com.ua/ru/articles/uml-diagrams.html>
- 20 Платформы как сайт // Электронная версия на сайте <https://cpa-rating.ru/stati/chto-takoe-platforma-sayta>

Приложение А

Техническое задание

А1.1 Техническое задание на разработку автоматизированной системы для малых предприятий

Настоящее техническое задание распространяется на разработку автоматизированной системы работы малых предприятий. Автоматизированная система позволит улучшить все процессы работы предприятий, а так же увеличит уровень эффективности предприятия. Обеспечит быстрое получение данных о рабочих, поставленных задач, и процента выполнения заданий.

А1.1.1 Основания для разработки

Система разрабатывается на основании устного распоряжения научного руководителя для создания автоматизированной системы управления малыми предприятиями.

А.1.1.2 Назначение

Разрабатываемый проект предназначен для сбора, хранения и обработки информации рабочих в малом бизнесе.

А.1.1.3 Требования к функциональным характеристикам

Система должна обеспечить возможность выполнения следующих

Продолжение приложения А

функций:

- хранения данных;
- поиск рабочих по базе данных;
- добавление новых задач;
- добавление дедлайнов;
- добавление и удаление пользователей;
- сохранение выполненных заданий на сервер.

А.1.1.4 Требования к надежности

Обеспечить конфиденциальность данных предприятия о товарах. Предусмотреть блокировку действий пользователей по ролям.

А.1.1.5 Требования к составу и параметрам технических средств

Система должна работать на всех персональных компьютерах и различных браузерах. Минимальные требования: тип процессора – Windows XP и выше, оперативная память – 64 Мб RAM и выше.

А.1.1.6 Требования к информационной и программной совместимости

Система должна работать под управлением операционных систем семейства Microsoft Windows.

Приложение Б

Текст программы

Database:

```
CREATE TABLE `curator_group` (  
  `id` int NOT NULL,  
  `curator_id` int NOT NULL,  
  `group_id` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Структура таблицы `groups`

```
CREATE TABLE `groups` (  
  `id` int NOT NULL,  
  `group_name` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci DEFAULT NULL,  
  `courses` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `groups`

```
INSERT INTO `groups` (`id`, `group_name`, `courses`) VALUES  
(8, 'TEST GRoup 1', '3'),  
(9, 'TEST GRoup 2 ', '3'),  
(13, 'test1', '2'),  
(14, 'twitch', '1');
```

-- Структура таблицы `news`

```
CREATE TABLE `news` (  
  `id` int NOT NULL,  
  `name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
NOT NULL,  
  `time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

Продолжение приложение Б

```
`text` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `news`

```
INSERT INTO `news` (`id`, `name`, `time`, `text`) VALUES  
(8, 'СОТРУДНИКИ НОВЫЙ ПРОЕКТ ', '2022-04-22 03:55:39', 'Сегодня  
поступило новый проект , готовимся'),  
(9, 'Сотрудники новый проект ', '2022-04-22 19:05:25', 'Сегодня поступило  
новый проект , готовимся'),  
(10, 'Тестовый заголовок', '2022-04-28 11:11:22', 'Всем привет');
```

-- Структура таблицы `otchet`

```
CREATE TABLE `otchet` (  
  `id` int NOT NULL,  
  `user_id` int DEFAULT NULL,  
  `subject_id` int DEFAULT NULL,  
  `teacher_id` int DEFAULT NULL,  
  `percent` int DEFAULT NULL,  
  `date` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `file` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

-- Дамп данных таблицы `otchet`

```
INSERT INTO `otchet` (`id`, `user_id`, `subject_id`, `teacher_id`, `percent`, `date`,  
  `file`) VALUES  
(1, 31, 13, 33, 75, '2022-04-29 05:34:33', 'distance (2).sql'),  
(3, 31, 13, 32, 35, '2022-04-29 09:54:31', 'Пояснительная записка к дипломному  
проекту );
```

-- Структура таблицы `subjects`

```
CREATE TABLE `subjects` (  
  `id` int NOT NULL,
```

Продолжение приложение Б

```
`subject_name` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `subjects`

```
INSERT INTO `subjects` (`id`, `subject_name`) VALUES  
(10, 'Тест преокты 1'),  
(11, 'Тест преокты 2'),  
(12, 'PHP '),  
(13, 'Продажа');
```

-- Структура таблицы `task`

```
CREATE TABLE `task` (  
  `id` int NOT NULL,  
  `name_task` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci DEFAULT NULL,  
  `date` date DEFAULT NULL,  
  `file` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Структура таблицы `timetable`

```
CREATE TABLE `timetable` (  
  `id` int NOT NULL,  
  `subject_id` int NOT NULL,  
  `teacher_id` int NOT NULL,  
  `group_id` int NOT NULL,  
  `datetime` datetime NOT NULL,  
  `status` int DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `timetable`

Продолжение приложение Б

```
INSERT INTO `timetable` (`id`, `subject_id`, `teacher_id`, `group_id`, `datetime`,  
`status`) VALUES  
(22, 13, 33, 8, '2022-04-28 19:04:00', NULL),  
(23, 12, 32, 9, '2022-04-27 18:04:00', NULL);
```

-- Структура таблицы `tsk`

```
CREATE TABLE `tsk` (  
  `id` int NOT NULL,  
  `user_id` int NOT NULL,  
  `group_id` int NOT NULL,  
  `subject_id` int NOT NULL,  
  `doc` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
  NOT NULL,  
  `date` date NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `tsk`

```
INSERT INTO `tsk` (`id`, `user_id`, `group_id`, `subject_id`, `doc`, `date`) VALUES  
(14, 32, 8, 12, 'Образцы документов (протокола, листы голосования и т.п.)  
(1).docx', '2022-04-27');
```

-- Структура таблицы `users`

```
CREATE TABLE `users` (  
  `id` int NOT NULL,  
  `full_name` varchar(255) CHARACTER SET utf8mb4 COLLATE  
  utf8mb4_unicode_ci DEFAULT NULL,  
  `login` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
  DEFAULT NULL,  
  `email` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
  DEFAULT NULL,  
  `password` varchar(500) CHARACTER SET utf8mb4 COLLATE  
  utf8mb4_unicode_ci DEFAULT NULL,  
  `role` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci  
  DEFAULT NULL,
```

Продолжение приложение Б

```
`position` varchar(500) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `users`

```
INSERT INTO `users` (`id`, `full_name`, `login`, `email`, `password`, `role`,  
`position`) VALUES  
(26, 'admin', 'admin', 'admin@gmail.com', 'admin', '1', 'admin'),  
(27, 'Мун Артур', 'Test.Test', NULL, 'aaa', '2', 'Исполняющий'),  
(31, 'test 1', 'test1', NULL, 'test1', '2', 'Исполняющий'),  
(32, 'test2', 'test2', NULL, 'test2', '3', 'Директор'),  
(33, 'Robert Einshtein', 'test3', NULL, 'test3', '3', 'Директор'),  
(34, 'виталик', 'vital', NULL, '123', '1', 'Директор');
```

-- Структура таблицы `user\_group`

```
CREATE TABLE `user_group` (  
  `id` int NOT NULL,  
  `user_id` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NOT NULL,  
  `group_id` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

-- Дамп данных таблицы `user\_group`

```
INSERT INTO `user_group` (`id`, `user_id`, `group_id`) VALUES  
(42, '27', '9'),  
(43, '31', '8'),  
(44, '32', '13'),  
(47, '34', '14');
```

-- Индексы таблицы `curator\_group`

```
ALTER TABLE `curator_group`
```

Продолжение приложение Б

```
ADD PRIMARY KEY (`id`);

-- Индексы таблицы `groups`
--
ALTER TABLE `groups`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `news`
ALTER TABLE `news`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `otchet`

ALTER TABLE `otchet`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `subjects`

ALTER TABLE `subjects`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `task`

ALTER TABLE `task`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `timetable`

ALTER TABLE `timetable`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `tsk`

ALTER TABLE `tsk`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `users`
```

Продолжение приложение Б

```
ALTER TABLE `users`  
  ADD PRIMARY KEY (`id`);
```

-- Индексы таблицы `user\_group`

```
ALTER TABLE `user_group`  
  ADD PRIMARY KEY (`id`);
```

-- AUTO_INCREMENT для таблицы `curator\_group`

```
ALTER TABLE `curator_group`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;
```

-- AUTO_INCREMENT для таблицы `groups`

```
ALTER TABLE `groups`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;
```

-- AUTO_INCREMENT для таблицы `news`

```
ALTER TABLE `news`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

-- AUTO_INCREMENT для таблицы `otchet`

```
ALTER TABLE `otchet`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
```

-- AUTO_INCREMENT для таблицы `subjects`

```
ALTER TABLE `subjects`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;
```

-- AUTO_INCREMENT для таблицы `task`

Продолжение приложение Б

```
ALTER TABLE `task`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;  
  
-- AUTO_INCREMENT для таблицы `timetable`  
  
ALTER TABLE `timetable`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=24;  
  
-- AUTO_INCREMENT для таблицы `tsk`  
  
ALTER TABLE `tsk`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;  
  
-- AUTO_INCREMENT для таблицы `users`  
  
ALTER TABLE `users`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=35;  
  
-- AUTO_INCREMENT для таблицы `user_group`  
  
ALTER TABLE `user_group`  
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=48;  
COMMIT;
```

Index

```
<?php  
session_start();  
  
if ($_SESSION['user']) {  
    header('Location:start.php');
```

Продолжение приложение Б

?>

```
<!doctype html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <title>Авторизация и регистрация</title>
  <link rel="stylesheet" href="assets/css/main.css">
</head>
<body>
```

```
<!-- Форма авторизации -->
```

```
<form action="vendor/signin.php" method="post">
  <label>Логин</label>
  <input type="text" name="login" placeholder="Введите свой логин">
  <label>Пароль</label>
  <input type="password" name="password" placeholder="Введите пароль">
  <button type="submit">Войти</button>
```

```
<?php
if ($_SESSION['message']) {
  echo '<p class="msg"> ' . $_SESSION['message'] . ' </p>';
}
unset($_SESSION['message']);
?>
</form>
```

```
</body>
</html>
```