Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

Безопасность WEB приложений

Задание вместо второй аттестации

Выполнил студент группы IA1802:

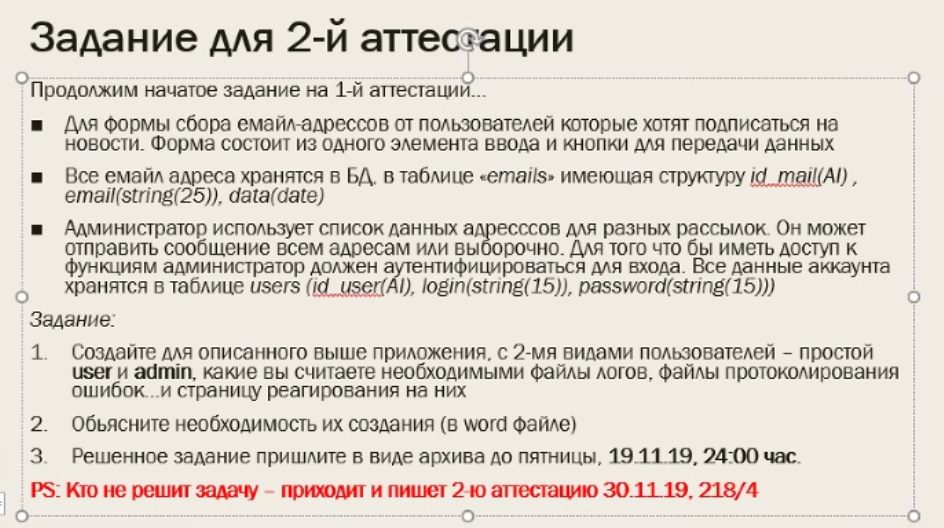
Паненко Е.

Проверил преподаватель:

Плешка Н.

Кишинёв, 2020

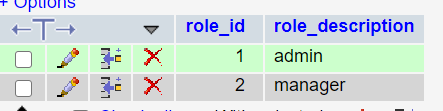
**Условие задания:**



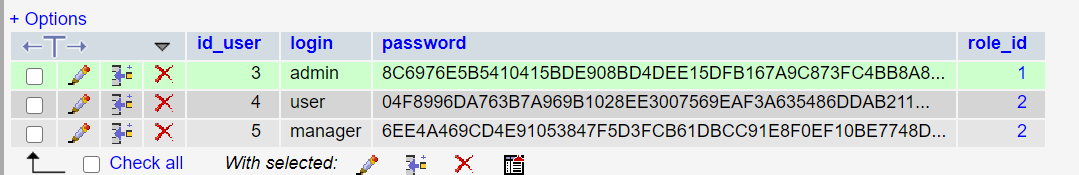
**Внедрение ролей**

Сперва изменим структуру нашей **БД** следующим образом:

Добавим таблицу **roles** с двумя ролями: **admin & manager**

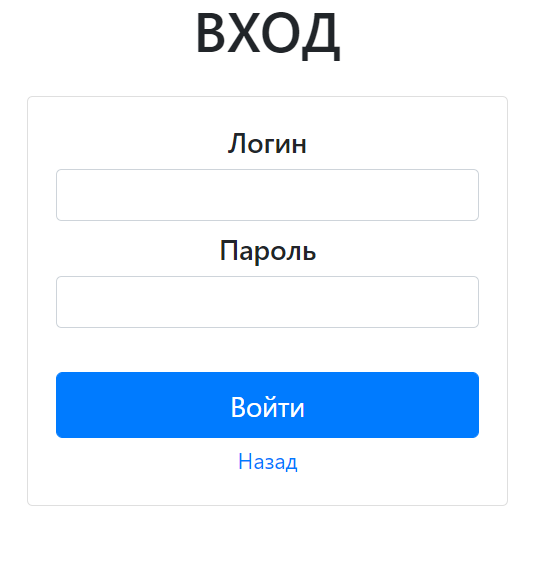


Далее свяжем эту таблицу с таблицей **users** и выдадим каждому пользователю соответствующий **role\_id**

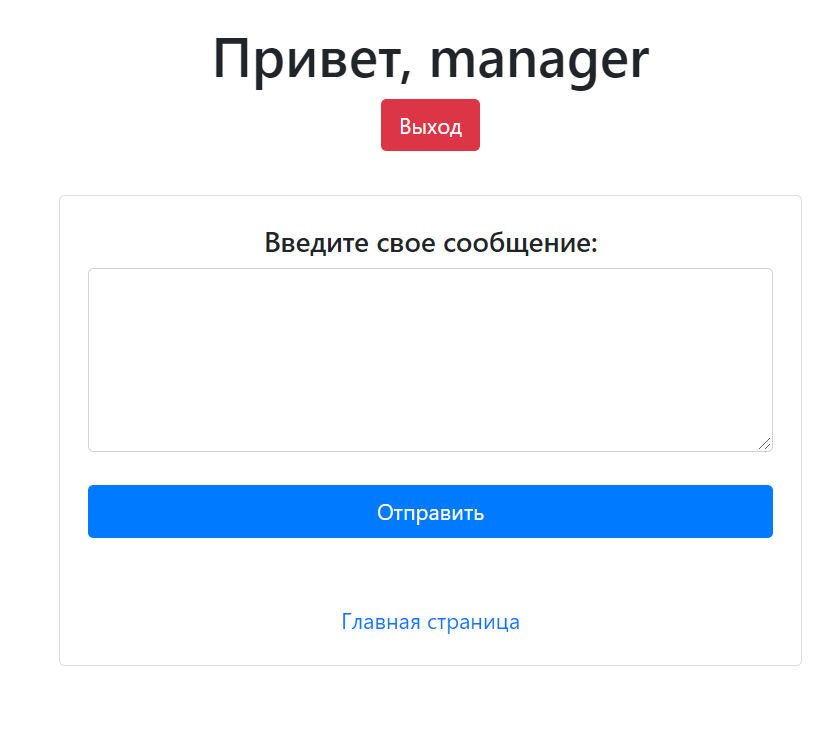


Вначале продемонстрируем функционал

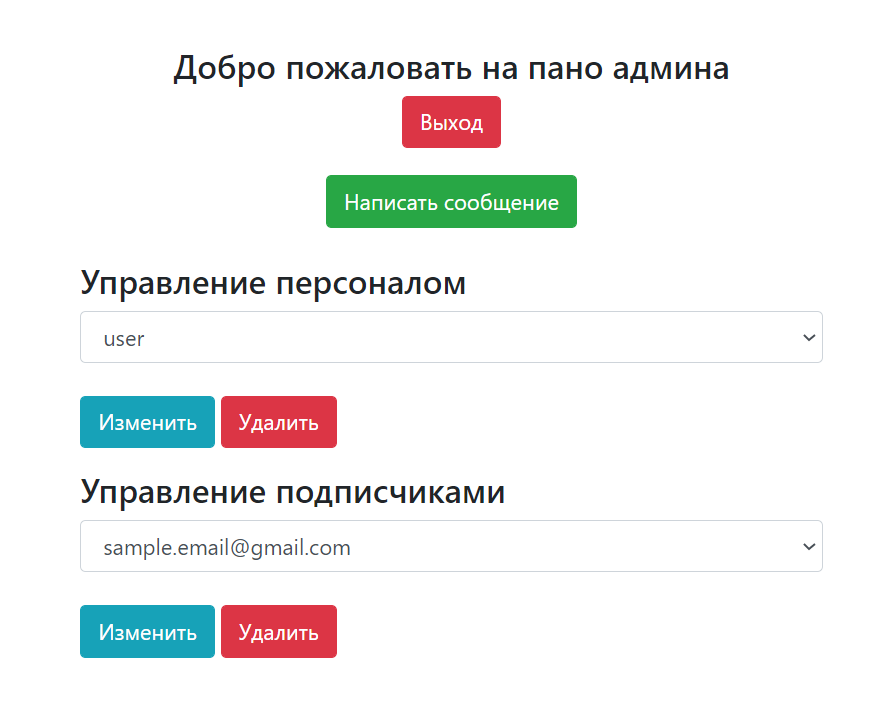
При попытке входа в система, в зависимости от типа пользователя он будет перенаправлен на разные страницы.



Например, при сценарии входа пользователя manager он будет перенаправлен на страницу рассылки сообщений



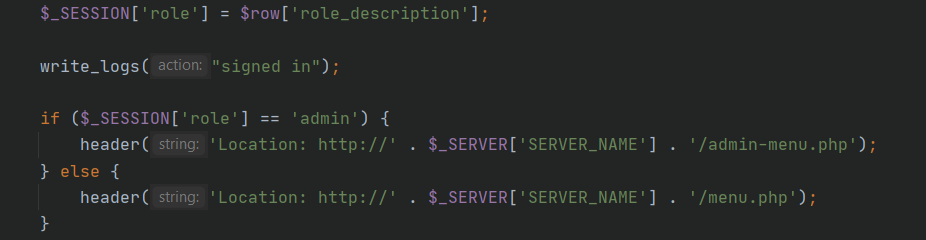
В случае входа администратора он будет переправлен на свою панель



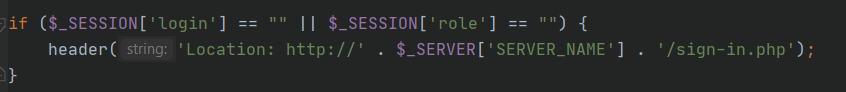
Отсюда он может перейти на панель управления рассылками, тем самым он имеет доступ к функционалу менеджера, но менеджер не имеет доступа к панели администратора.

**Реализация**:

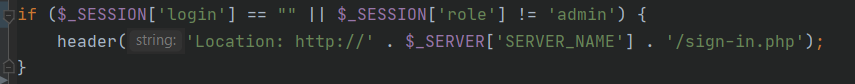
Вначале будем сохранять роль в переменной сессии и производить редирект на нужную страницу в зависимости от роли пользователя.



Далее в файле menu.php будем просто проверять, если пользователь зашел, так как к этому пано имеют доступ все аутентифицированные пользователи.



К файлу admin-menu.php имеет доступ только пользователь с ролью admin



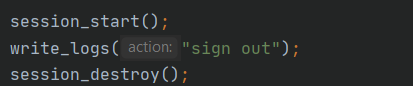
Внедрение данного механизма позволяет нам расширить функционал приложения и разделить пользователей по правам.

Далее приступим к журнализации действий пользователя. Начнем с создания скрипта, который будет логировать действия пользователей.

function write\_logs($action) {  
 $filename = "..\\log.txt";  
 $fieldsSeparator = "\t\t";  
  
 if (!file\_exists($filename)) {  
 $file = fopen($filename, "w") or die("Error");  
  
 $logLine = "Время".$fieldsSeparator.$fieldsSeparator.  
 "Файл".$fieldsSeparator.$fieldsSeparator.  
 "User IP".$fieldsSeparator.$fieldsSeparator.  
 "Session ID".$fieldsSeparator.$fieldsSeparator.  
 "Username".$fieldsSeparator.  
 "Role".$fieldsSeparator.  
 "Action";  
  
 fwrite($file, $logLine."\r\n");  
 fclose($file);  
 }  
  
 $logLine = date("d/m/y H:i:s").$fieldsSeparator.  
 $\_SERVER['PHP\_SELF'].$fieldsSeparator.  
 $\_SERVER['REMOTE\_ADDR'].$fieldsSeparator.  
 (("".SESSION\_ID() != "") ? SESSION\_ID() : "[-]").$fieldsSeparator.  
 (("".$\_SESSION['login'] != "") ? $\_SESSION['login'] : "[-]").$fieldsSeparator.  
 (("".$\_SESSION['role'] != "") ? $\_SESSION['role'] : "[-]").$fieldsSeparator.  
 $action;  
  
 $file = fopen($filename, "a") or die("Error");  
 fwrite($file, $logLine."\r\n");  
 fclose($file);  
}

Использовать эту функцию будем следующим образом – в точках, где необходима будем вызывать наш метод передавая ему в качестве параметра действие которое производит пользователь.

Пример такого вызова:

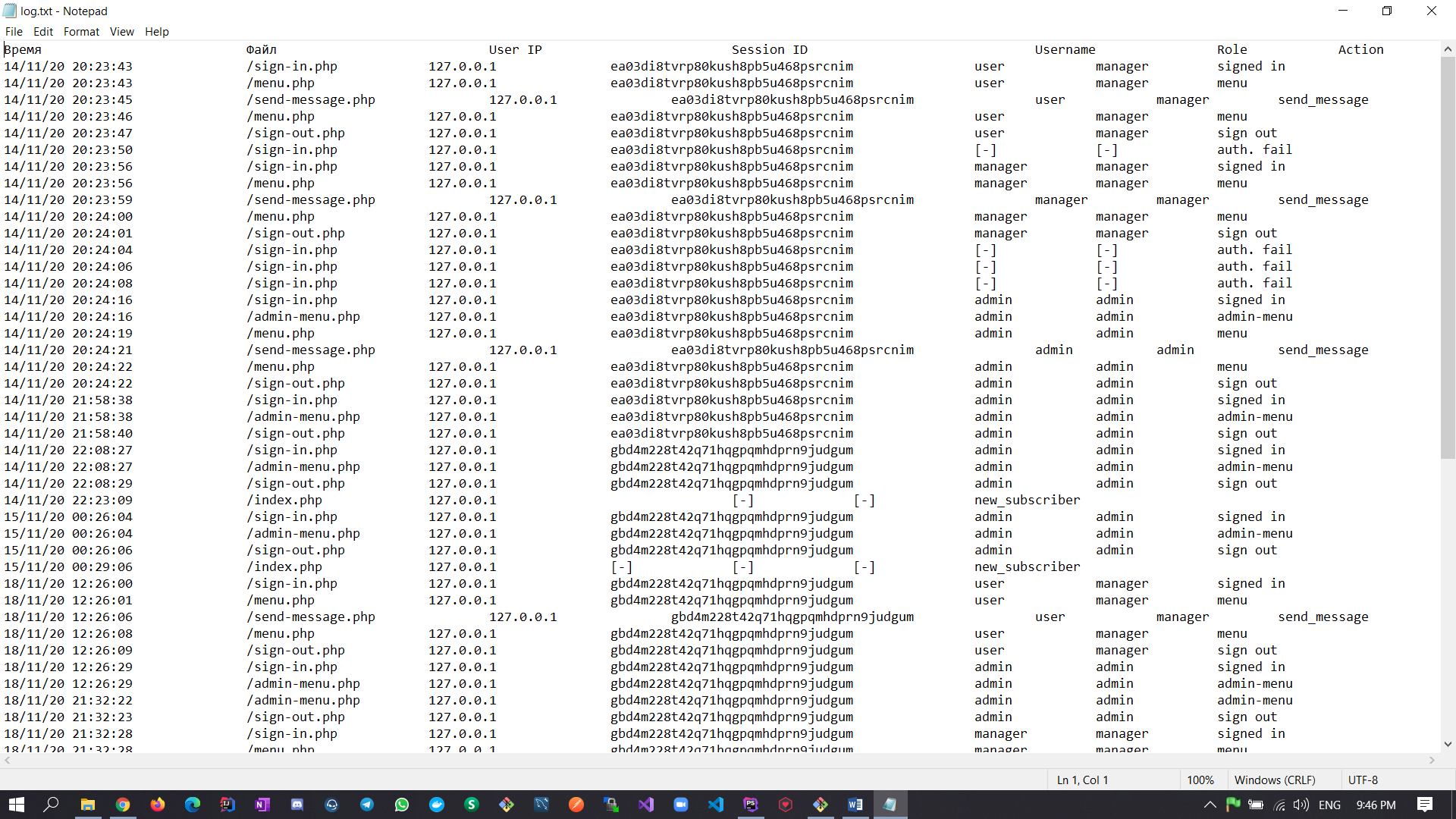


При выходе любого пользователя мы будем это регистрировать в логах.

Действия, которые логируются:

* Вход пользователя в систему
* Переход на страницу отправки сообщения
* Переход на пано администратора
* Отправка сообщения
* Добавление email к рассылке
* Выход из системы

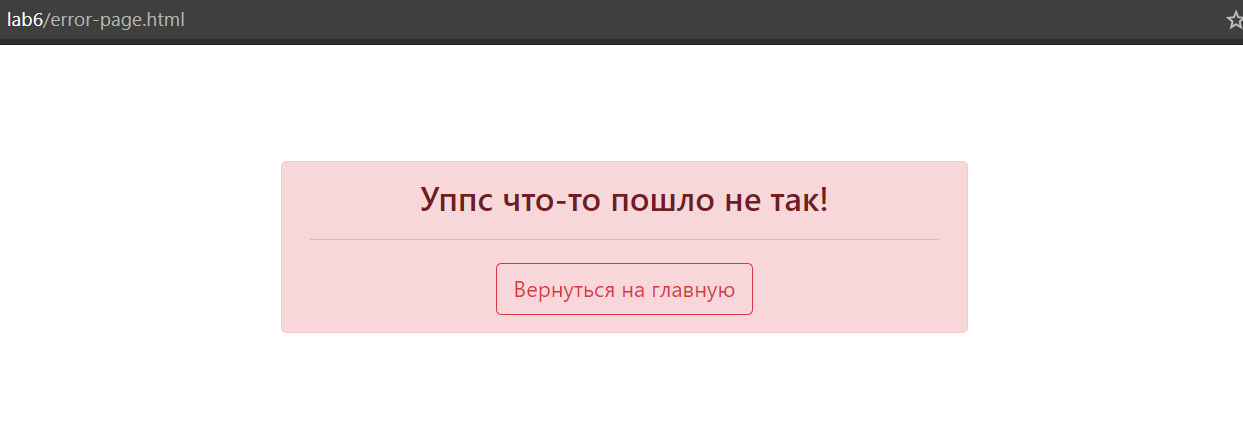
Пример логов:



Файл хранится не в директории с проектом, а на одну директорию выше.

Далее перейдем к обработке ошибок и их журнализацию.

Для начала создадим страницу, которую будем показывать пользователю в случае возникновения ошибки.



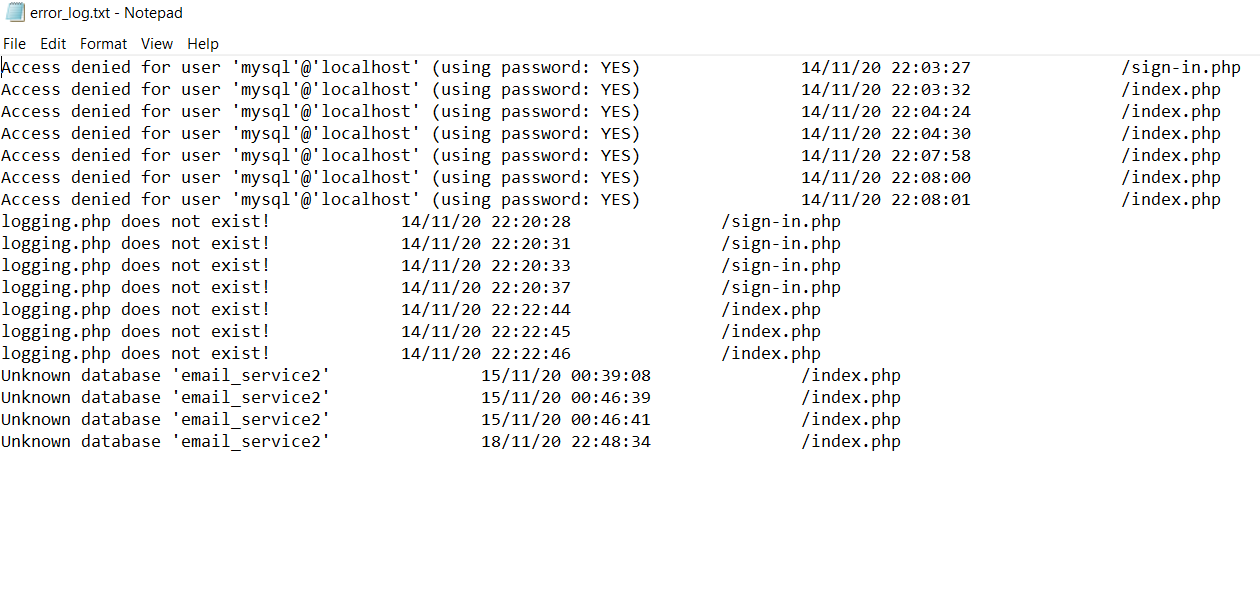
Обрабатывать и логировать ошибки будем с помощью конструкции try catch и встроенной функции error\_log()

Пример обработки наличия соединения с БД.



Так же будем обрабатывать корректное подключение файла logging.php

Пример файла error\_log.txt, который по аналогии с первым хранится на одну директорию выше.



**Анализ проделанной работы:**

В ходе выполнения задания вместо второй аттестации мы реализовали следующие механизмы:

* Внедрили контроль сессий
* Внедрили роли в наше приложение
* Разделили функционал администратора и менеджера
* Внедрили журнализацию действий пользователя
* Внедрили обработку и журнализацию ошибок

Внедренная журнализация позволит нам в будущем:

* Вести контроль за определёнными пользователями
* Засекать многочисленные неудачные попытки входа
* Разрешать возможные конфликты с персоналом
* Обработка ошибок поможет быстрее находить причину неполадки