

ТРЕБОВАНИЯ К ПРОЕКТУ ПО КУРСУ «ВЕБ-РАЗРАБОТКА»

Вариант 2

ВВЕДЕНИЕ

Требуется разработать веб-приложение, которое будет включать в себя:

- Создание таблиц в базе данных СУБД PostgreSQL.
- Реализация запросов в БД с группировкой.
- Реализация серверной логики на PHP.
- Реализация клиентской логики на JavaScript.
- Реализация как серверной, так и клиентской отрисовки HTML.
- Аутентификация по логину/пароллю и авторизация на основе ролей для доступа к контенту.
- Реализация полноценного CRUD (создание/получение/изменение/удаление записей) для одной таблицы из веб-интерфейса.

1. ТРЕБОВАНИЯ К БД

2.1 Таблица Роли

Название таблицы: roles.

Таблица 1.1 – Роли. Структура

Название	Тип данных	NOT NULL	Примечание
id	serial	Да	Первичный ключ
code	character varying(20)	Да	Уникальность на (code)
name	character varying(40)	Да	Уникальность на (name)

Таблица 1.2 – Роли. Заполнение

Название	Код
Пользователь2	USR
Администратор	ADM
Модератор2	MOD
Статистик2	STAT

Заполнить таблицу нужно в БД, меняться эти данные не будут. При желании можно добавить еще ролей кроме указанных четырех.

2.2 Таблица Пользователи

Название таблицы: users.

Таблица 1.3 – Пользователи. Структура

Название	Тип данных	NOT NULL	Примечание
id	serial	Да	Первичный ключ
name	character varying(40)	Да	
login	character varying(10)	Да	Уникальность на (login)
pass	character varying(20)	Да	

Таблица 1.4 – Пользователи. Заполнение

Имя	Логин	Пароль
Пользователь 1	user1	<задать самостоятельно>
Пользователь 2	user2	<задать самостоятельно>
Пользователь 3	user3	<задать самостоятельно>
<Ваше имя>	user_task2	<задать самостоятельно>

Заполнить таблицу нужно в БД, меняться эти данные не будут. При желании можно добавить еще пользователей кроме указанных четырех.

2.3 Таблица Роли пользователей

Название таблицы: user_roles.

Таблица 1.5 – Роли пользователей. Структура

Название	Тип данных	NOT NULL	Примечание
id	serial	Да	Первичный ключ
role_id	integer	Да	Внешний ключ на role.id
user_id	integer	Да	Внешний ключ на user.id

Таблица 1.6 – Роли пользователей. Заполнение

Логин	Роли
user1	Пользователь2
user2	Пользователь2, Модератор2
user3	Модератор2, Статистик2
user_task2	Администратор

Заполнить таблицу нужно в БД, меняться эти данные не будут.

1.4 Дополнительная таблица

Название таблицы extra

Таблица 1.7 – Дополнительная таблица. Структура

Название	Тип данных	NOT NULL	Примечание
id	serial	Да	Первичный ключ
name	character varying(50)	Да	

Примеры заполнения указаны в таблице 1.8. Необходимо придумать свой вариант из 4-10 записей. Это могут быть статусы, причины, категории, географические понятия и т.д. Заполнить таблицу нужно в БД, меняться эти данные не будут.

Таблица 1.8 – Примеры заполнения

name	name	name	name
Челябинск	В работе	В наличии	Тяжелый
Казань	На проверке	Забракован	Легкий
Уфа	Требуется доработки	Нет в наличии	Средний
Екатеринбург	Решен	Ожидается	

Должна быть как минимум одна доп. таблица. При желании можно создать еще несколько и связать их внешними ключами.

1.5 Основная таблица

Название таблицы: main.

Таблица 1.9 – Основная таблица. Структура

Название	Тип данных	NOT NULL	Примечание
id	serial	Да	Первичный ключ
extra_id	integer	Нет	Внешний ключ на extra.id
<...>	<...>	<...>	
<...>	<...>	<...>	
<...>	<...>	<...>	
<...>	<...>	<...>	<...>

Необходимо придумать названия и типы колонок самостоятельно, кроме id и extra_id. Суммарно требуется 5-7 колонок. Должны присутствовать как минимум типы данных integer и character varying(<N>).

Ниже указан пример для функционала аптеки.

Таблица 1.10 – Основная таблица. Структура

Название	Тип данных	NOT NULL	Примечание
id	serial	Да	Первичный ключ
extra_id	integer	Нет	Внешний ключ на extra.id
name	character varying(50)	Да	Название препарата
num	integer	Да	Кол-во в упаковке
producer	character varying(50)	Нет	Производитель

Таблица 1.11 – Пример заполнения таблиц

extra		main			
id	name	extra_id	num	name	producer
1	мг.	1	7	Сироп от кашля	
2	шт.	1	3	Парацетамол	ФармЛенд
3	мл.		5	Кальций Д3	Фабрика
		2	4	Витамин С	

Заполнять основную таблицу значениями через БД напрямую не обязательно, разве что для тестирования и отладки. Работа с этой таблицей будет вестись в веб-приложении.

2. ТРЕБОВАНИЯ К WEB-ИНТЕРФЕЙСУ И СЕРВЕРНОЙ ЧАСТИ

2.1 Страница авторизации

Путь к странице: `authorization.php`

Название страницы: Авторизация

Доступна неавторизированным пользователям. Если пользователь уже авторизован, перенаправлять на `/index.php`.

На странице располагается окно для ввода логина и пароля с двумя полями и кнопкой «Ок». Окно должно располагаться по центру страницы (по вертикали и горизонтали). Макет изображен на рисунке 2.1.

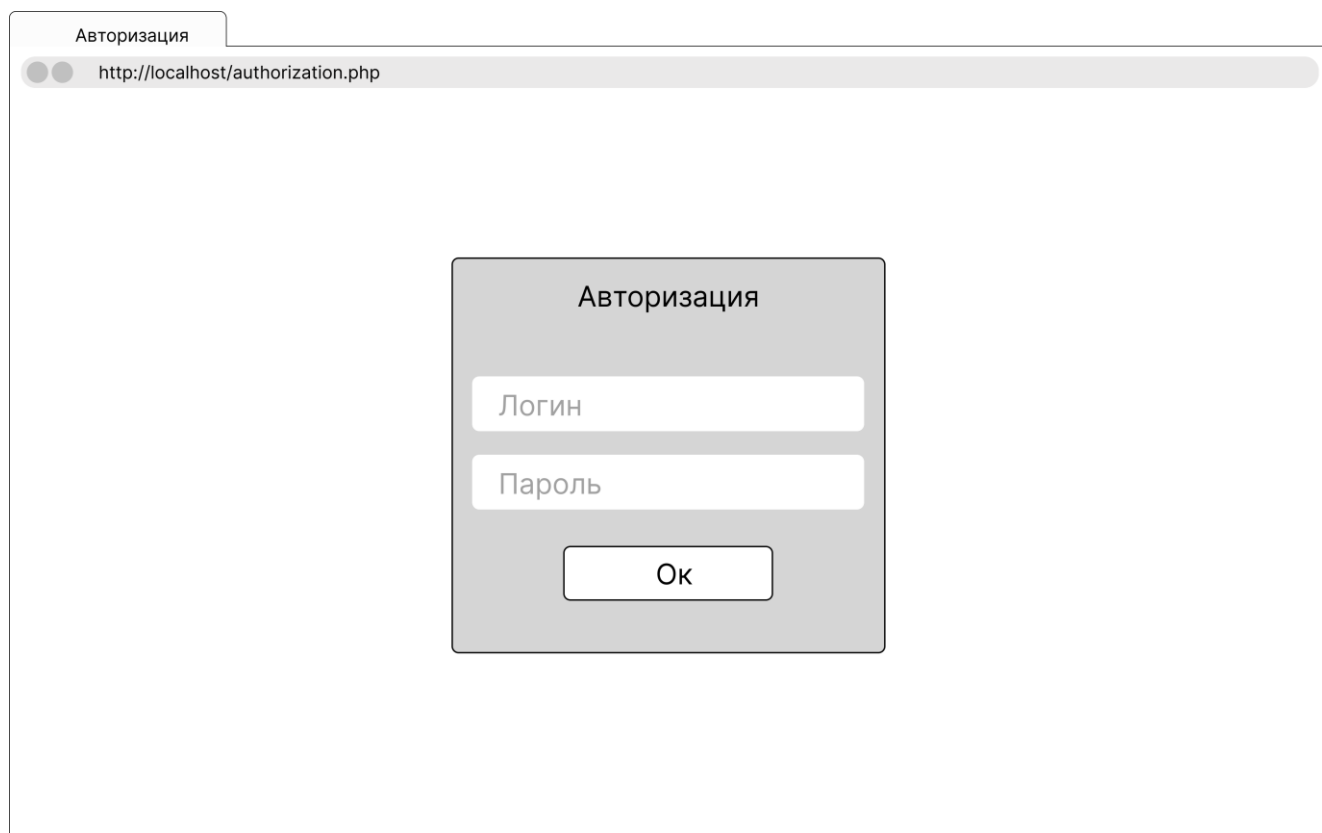


Рисунок 2.1 – Страница авторизации

При нажатии на «Ок» идет запрос к серверу. Тип запроса: POST. Если логин и пароль совпадают, то сохранить необходимые для дальнейшей логики данные в сессию и перенаправить пользователя на страницу `/index.php`.

Если пароль или логин не совпадают, то вывести сообщение с ошибкой «Неправильный логин или пароль». Отправлять запрос на сервер можно как при помощи fetch-запроса (рисунок 2.2), так и при помощи запроса с перезагрузкой страницы (рисунок 2.3). Выбрать любой способ на свое усмотрение.

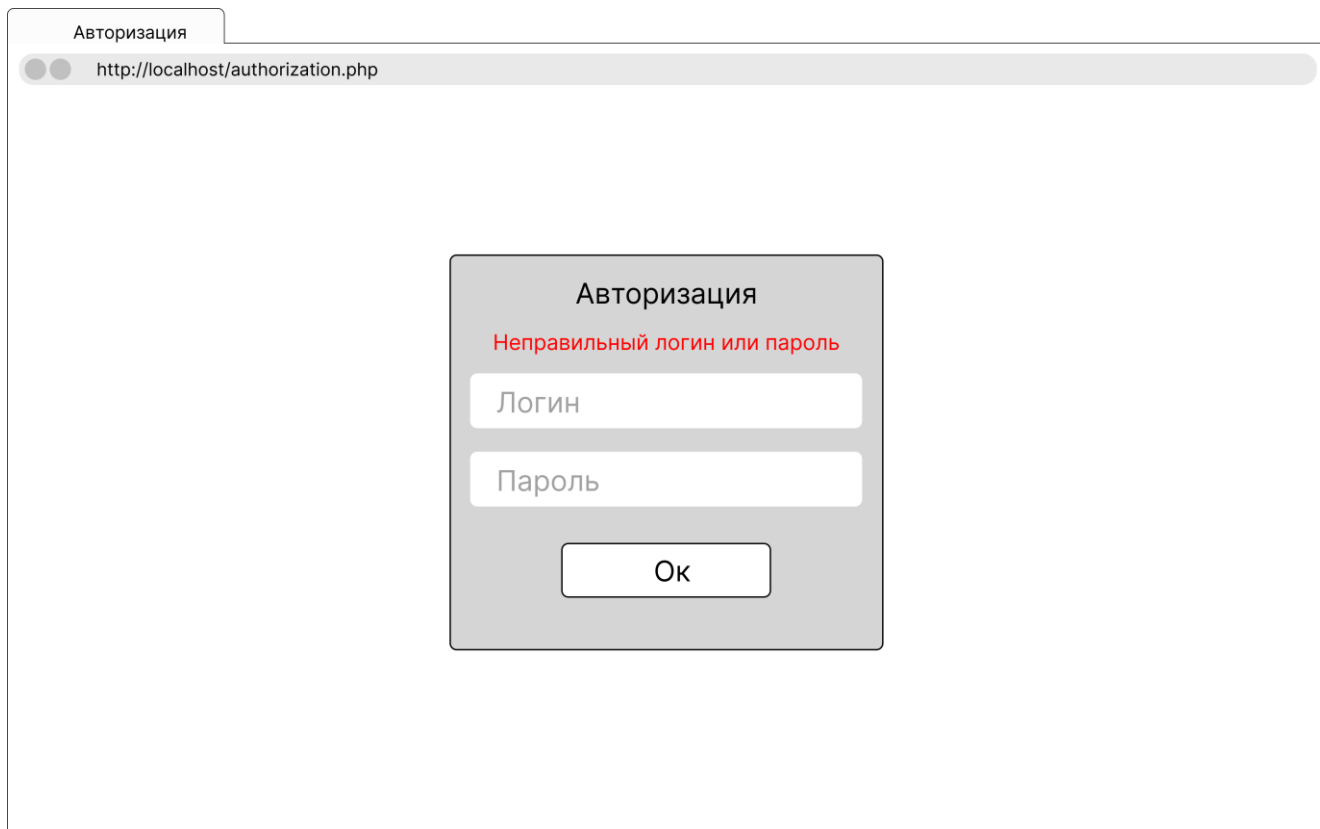


Рисунок 2.2 – Отправка логина и пароля при помощи fetch-запроса (без перезагрузки страницы)

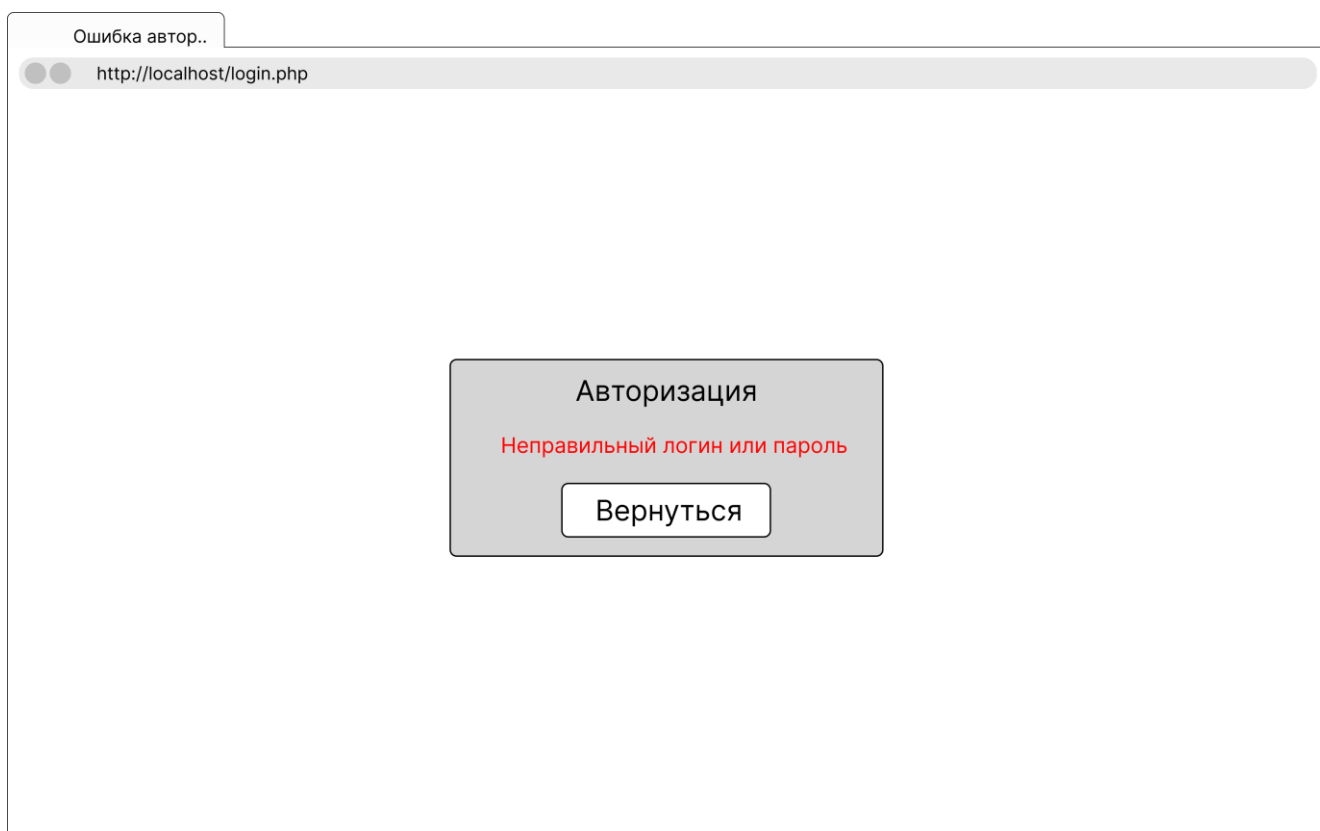


Рисунок 2.3 – Отправка логина и пароля при помощи запроса в <form>

2.2 Основная страница

Путь к странице: index.php

Название страницы: Главная

Доступ по ролям: Администратор, Статистик

Доступна авторизованным пользователям. Если пользователь не авторизован, перенаправлять на /authorization.php

Любое взаимодействие с сервером на данной странице не предполагает ее перезагрузки (кроме кнопки «Выход»). Используется клиентский рендеринг содержимого страницы при помощи fetch запросов к серверу.

На рисунке 2.4 изображен макет страницы без данных.

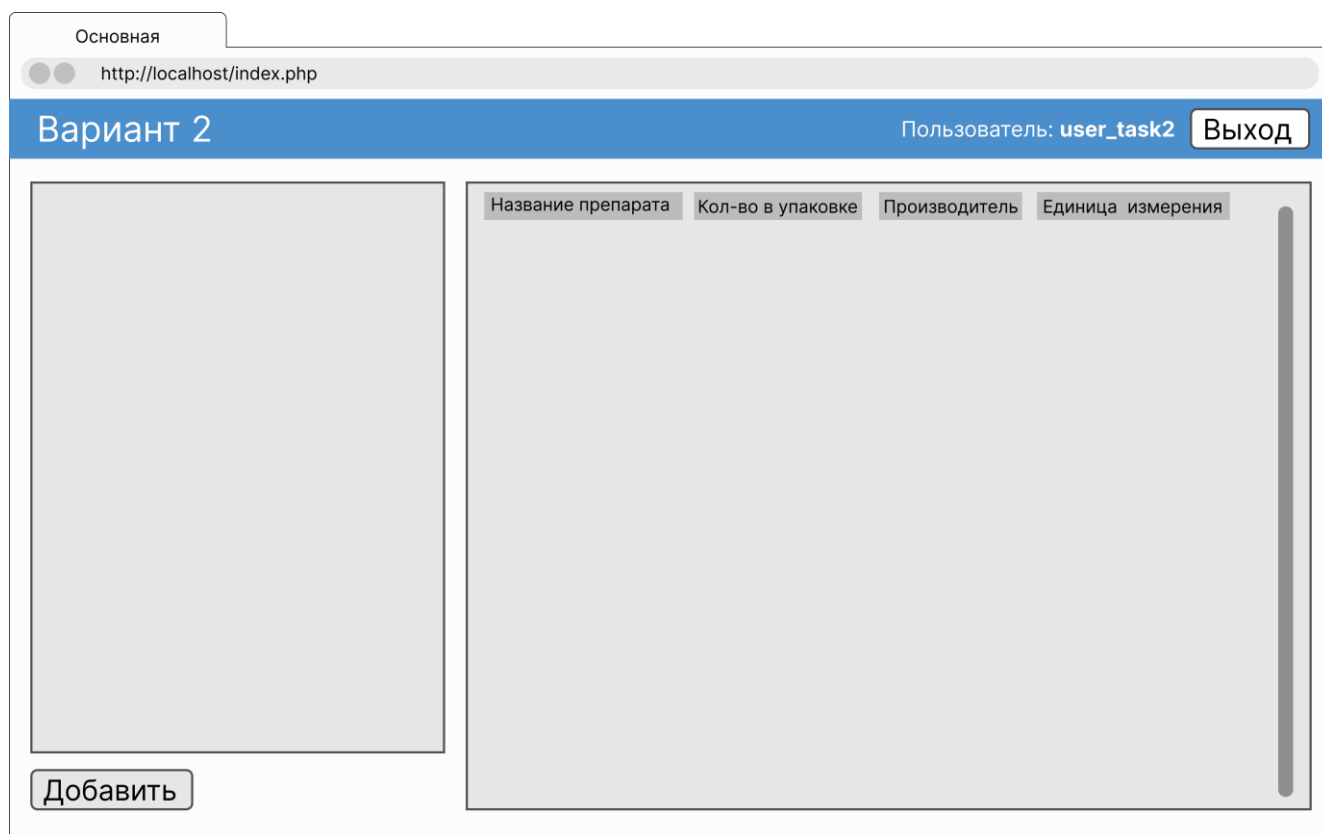


Рисунок 2.4 – Внешний вид страницы без данных

При нажатии на «Добавить» в окне справа появляются поля ввода информации основной таблицы (рисунок 2.5). Заголовок этого окна «Добавление новой записи». При нажатии на «Отмена».

При нажатии на «Добавить» в окне на сервер отправляется fetch-запрос. Если есть ошибки – они выводятся пользователю в alert (рисунок 2.6).

Если данные валидны и ошибок нет, то обновить данные в таблице слева, окно добавления становится пустым (Рисунок 2.7).

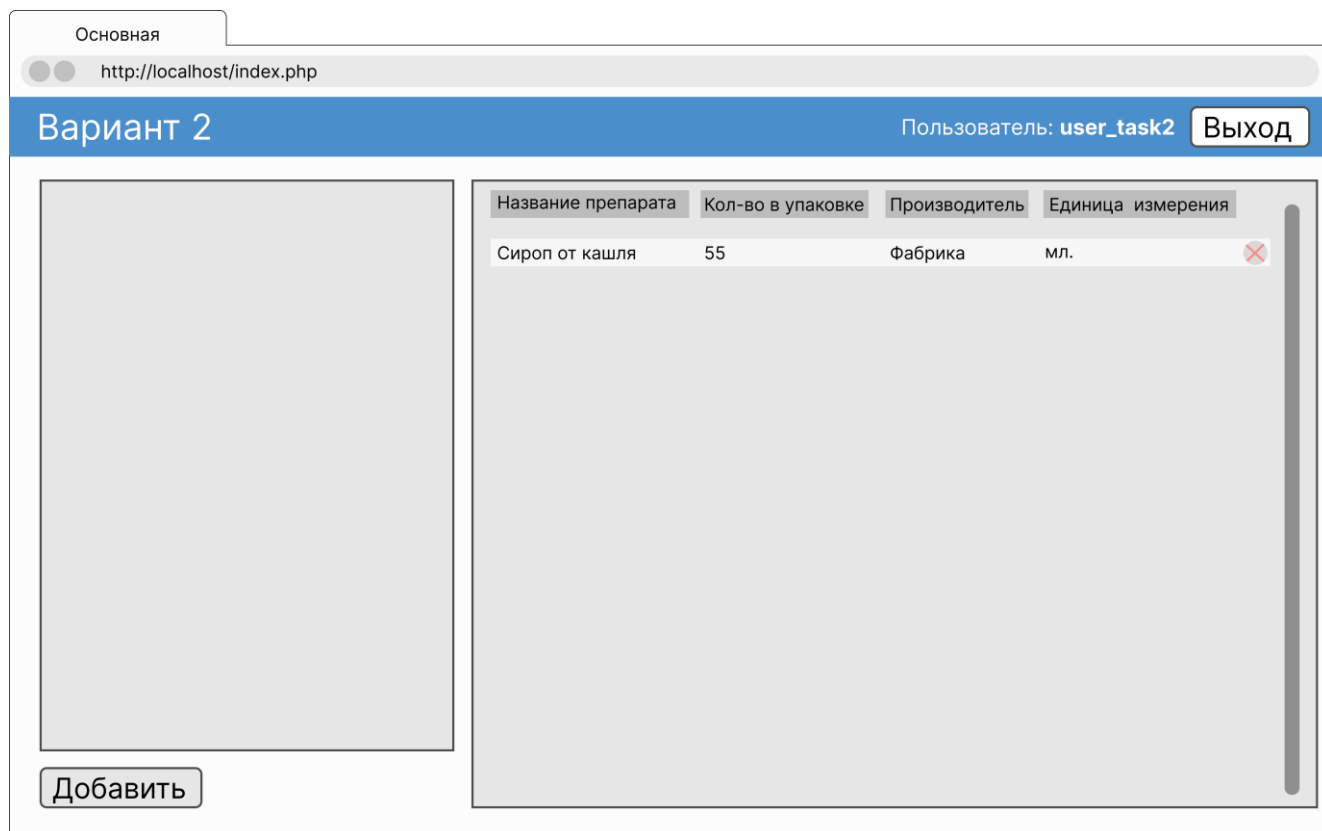


Рисунок 2.7 – Успешное добавление данных

При нажатии на элемент таблицы слева он становится активным, и строка таблицы выделяется цветом. В окно справа подтягиваются данные выбранной записи. Название окна «Изменение записи» (Рисунок 2.8)

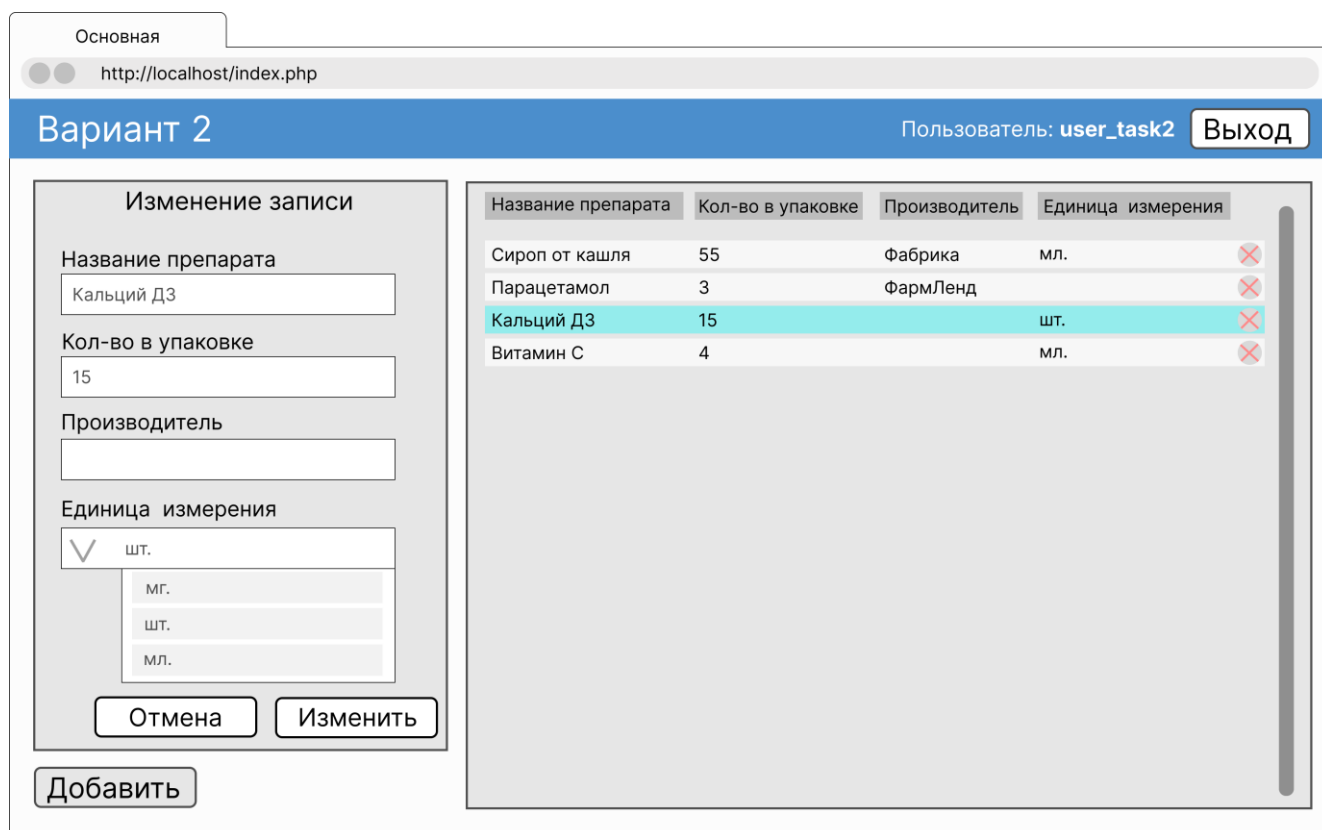


Рисунок 2.8 – Выбор строки и изменение данных

При нажатии на «Отмена» окно становится пустым, активная строка пропадает (рисунок 2.9).

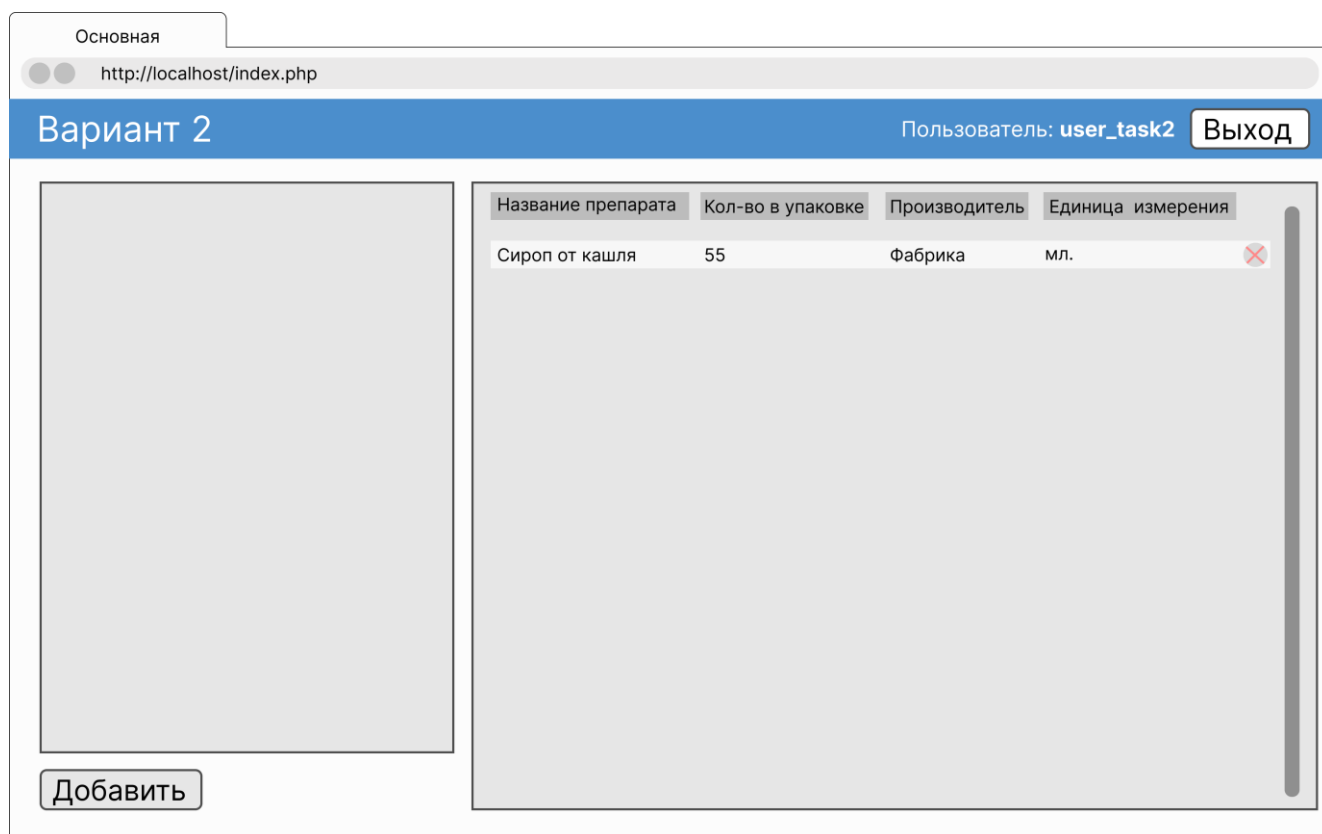



Рисунок 2.9 – Нет выделения строки

При нажатии на «Изменить» на сервер отправляется fetch-запрос. Если есть ошибки – они выводятся пользователю в alert.

Если данные валидны и ошибок нет, то обновить данные в таблице слева, окно добавления становится пустым, выделение активной строки пропадает (рисунок 2.9).

При нажатии на иконку удаления  пользователю выводится сообщение (confirm) «Вы действительно хотите удалить запись?» (Рисунок 2.10). Если пользователь нажал «Ок», то на сервер отправляется запрос на удаление, таблица слева обновляется.

Если удаляемая запись была выбрана и изменялась в окне справа, то окно должно стать пустым.

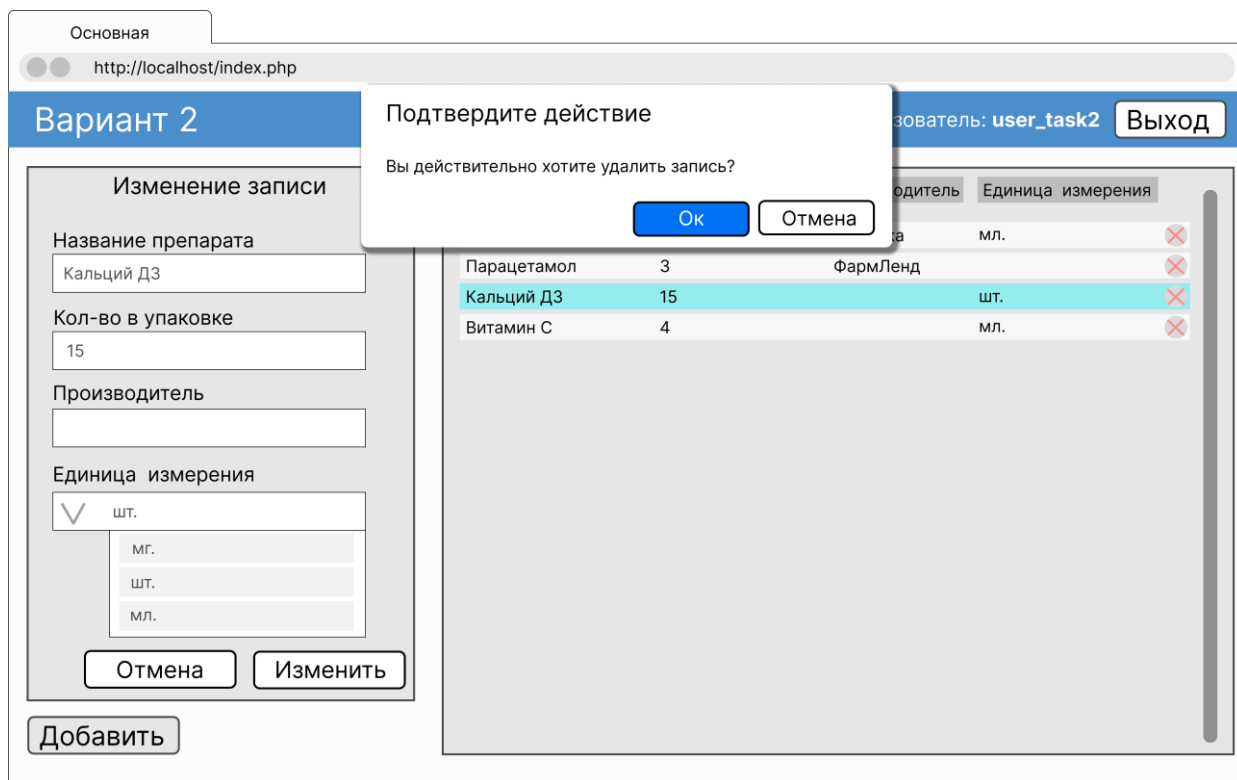


Рисунок 2.10 – Удаление записи

2.3 Требование к валидации данных

Минимум одно поле при добавлении и изменении должно быть подлежать валидации на сервере. Это может быть проверка на пустоту, проверка на тип, на равенство конкретному значению, больше/меньше и т.д. Ошибку необходимо вывести пользователю. Пример на рисунке 2.11.

Дополнительная валидация на стороне клиента – по желанию.

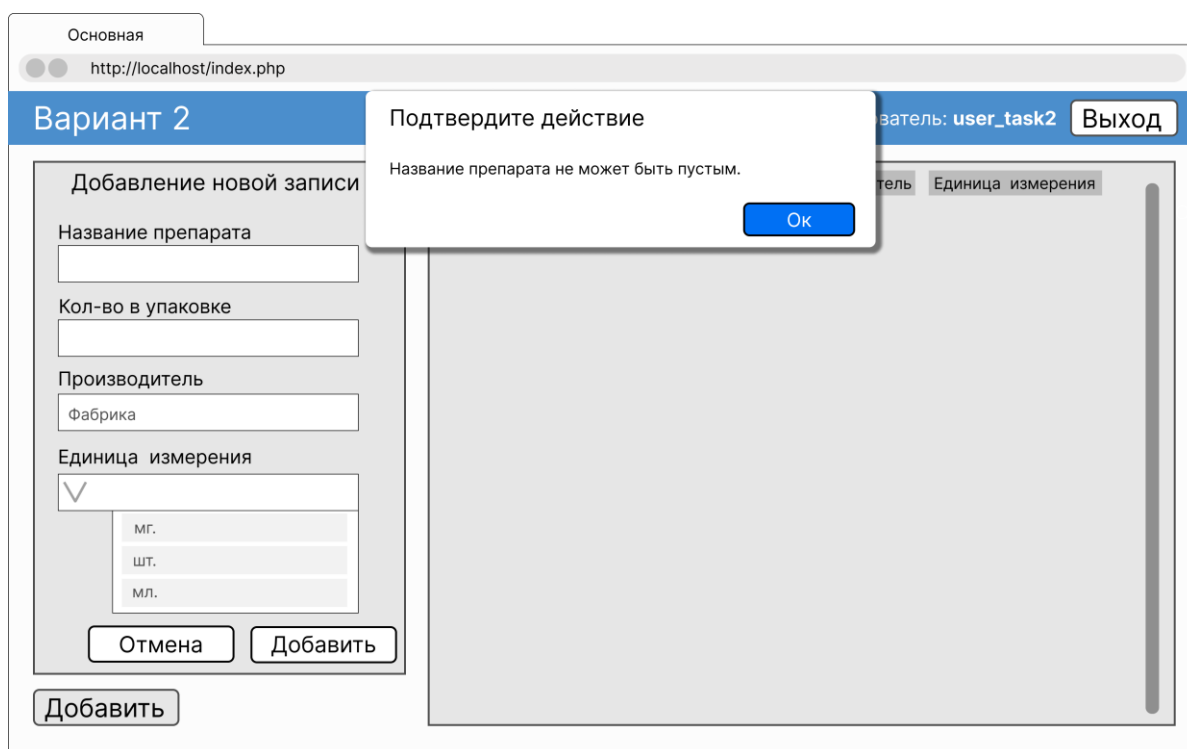


Рисунок 2.11 – Валидация данных

2.4 Требование к вычисляемому полю

Минимум одно поле при добавлении и изменении должно зависеть от значения в другом поле. Например, проверка на значение одного поля (больше/меньше/равно и т.д.) приводит к изменению значения в другом поле (установка конкретного значения, прибавка к значению и т.д.)

Если пользователь ввел значение поле самостоятельно, а при вычислении получилось другое значение, то приоритет по вычислению.

Например, задаем правило: если значение в поле «Кол-во в упаковке» больше или равно 70, то значение производителя автоматически устанавливается в «ФармЛенд» независимо от того, что ввел пользователь (рисунок 2.12).

Изменение записи

Название препарата
Кальций Д3

Кол-во в упаковке
100 1

Производитель

Единица измерения
шт.
мг.
шт.
мл.

Отмена Изменить 2

Название препарата	Кол-во в упаковке	Производитель	Единица измерения
Сироп от кашля	55	Фабрика	мл.
Парацетамол	3	ФармЛенд	шт.
Кальций Д3	15 3	ФармЛенд	шт.
Витамин С	4		мл.

Изменение записи

Название препарата
Кальций Д3

Кол-во в упаковке
100 1

Производитель
Свое значение 123 2 Будет проигнорировано

Единица измерения
шт.
мг.
шт.
мл.

Отмена Изменить 3

Название препарата	Кол-во в упаковке	Производитель	Единица измерения
Сироп от кашля	55	Фабрика	мл.
Парацетамол	3	ФармЛенд	шт.
Кальций Д3	15 4	ФармЛенд	шт.
Витамин С	4		мл.

Рисунок 2.12 – Требование к вычисляемому полю

Можно реализовать на JS, на PHP, на SQL, на выбор.

2.4 Страница статистики

Путь к странице: admin.php

Название страницы: Статистика

Доступ по ролям: Администратор, Статистик2

Доступна авторизованным пользователям. Если пользователь не авторизован, перенаправлять на /authorization.php

Страница формируется полностью на сервере методами PHP.

Сначала выводить список ролей пользователя по связи:

role.name, где

role.id = user_roles.role_id и user_roles.user_id = user_id из сессии пользователя. Сортировка по role.name по алфавиту. Можно сохранить роли в сессию сразу при авторизации.

Каждая роль выводится с новой строки, выводить в формате «- Название роли». Если название роли = «Администратор», то вывести жирным.

Далее выводить таблицу со всеми пользователями и их ролями. Стиль таблицы произвольный, главное – видимые границы строк и столбцов. В БД использовать запрос с группировкой. Должно быть 3 столбика: Логин, Имя и Роли. В столбец «Роли» вывести название всех ролей данного пользователя через запятую. Список должен совпасть со требованиями к БД. Сортировать на выбор либо по логину либо по имени.

Далее вывести таблицу с двумя столбцами: Поле и Кол-во. Стиль таблицы произвольный, главное – видимые границы строк и столбцов. В данную таблицу вывести список записей extra.name дополнительной таблицы extra.name и кол-во записей из основной таблицы по ссылкам main.extra_id = extra.id. Использовать запрос с группировкой. Сортировать по столбику «Кол-во».

Макеты с примерами на рисунках 2.13 и 2.14.

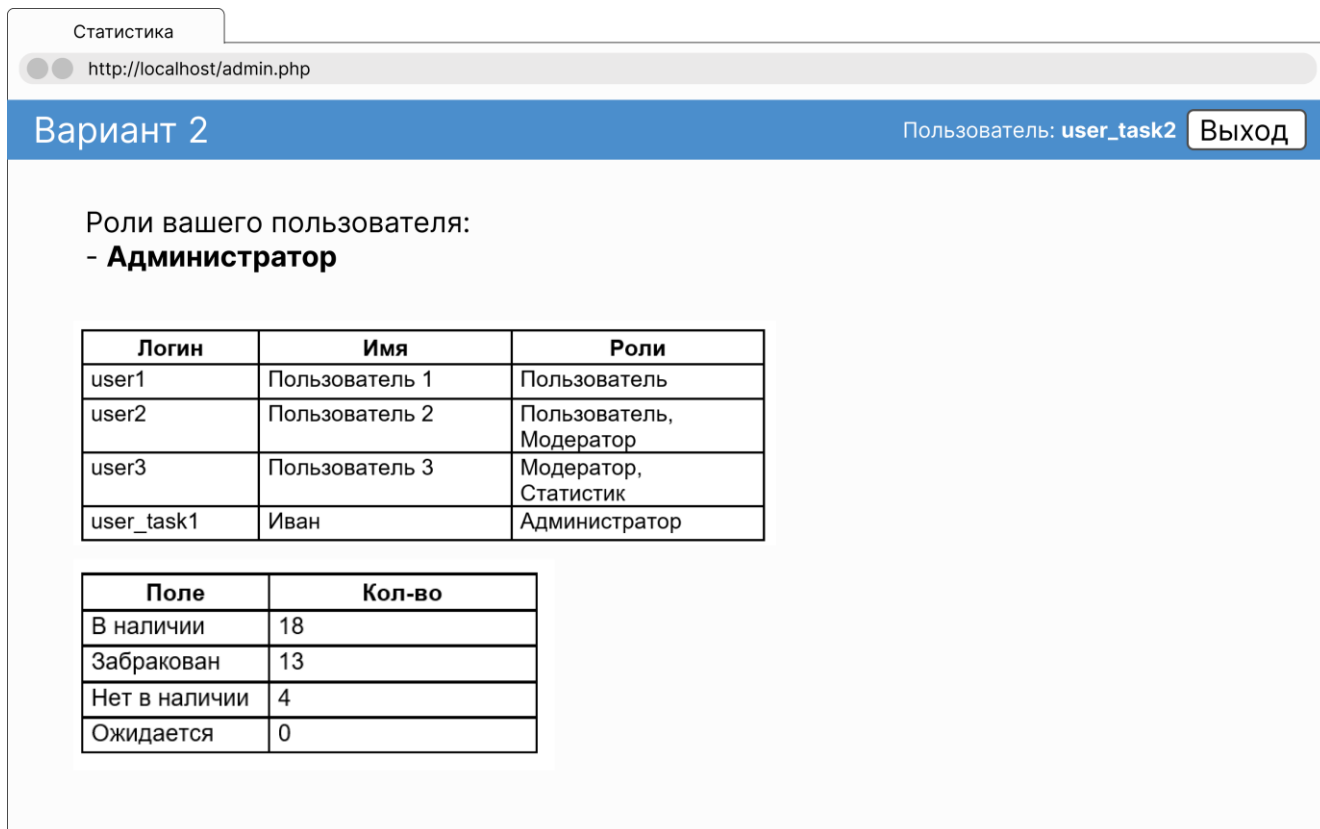


Рисунок 2.13 – Макет с примером страницы

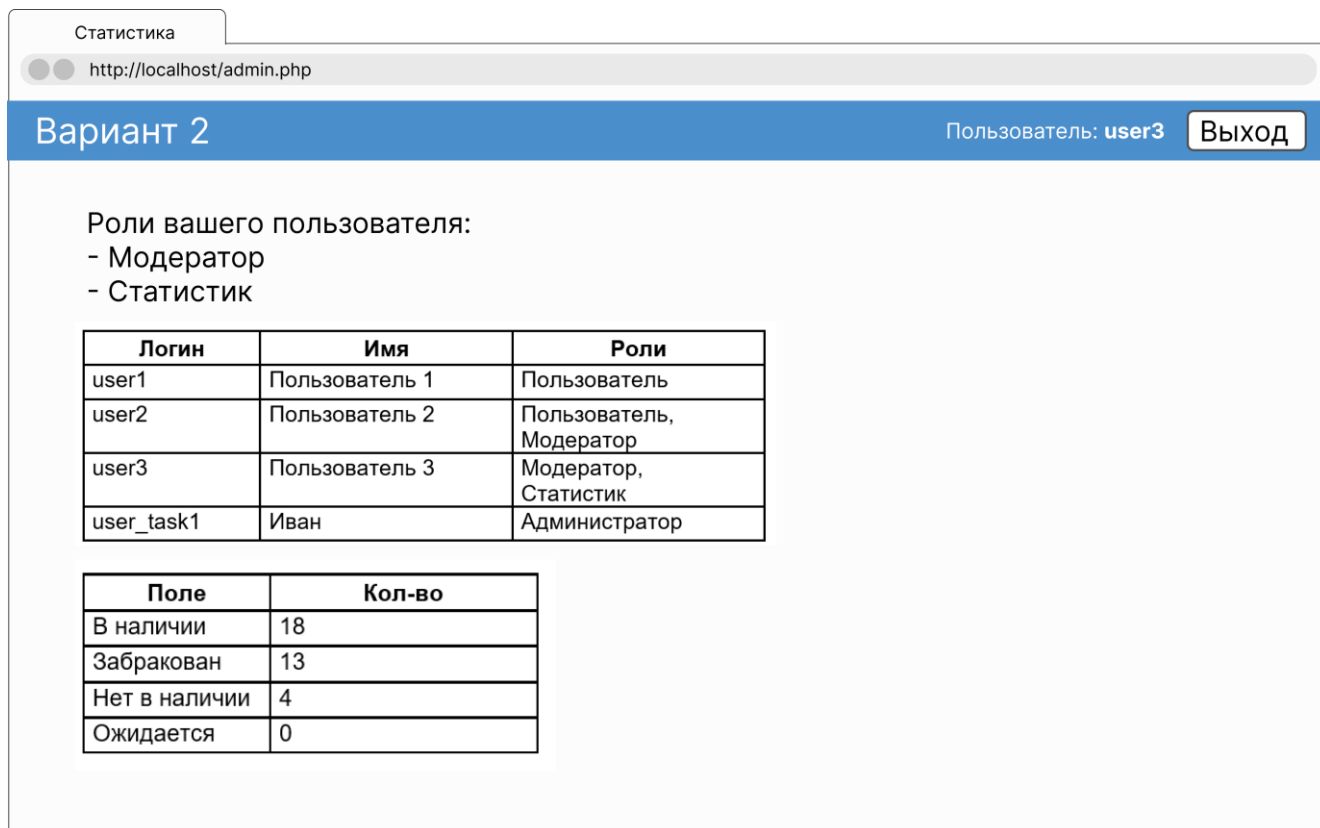


Рисунок 2.14 – Макет с примером страницы

Если у пользователя нет нужной роли для доступа к admin.php, то выводить список ролей пользователя и сообщение «У вас недостаточно привилегий для просмотра данной страницы» согласно макету на рисунке 2.15.

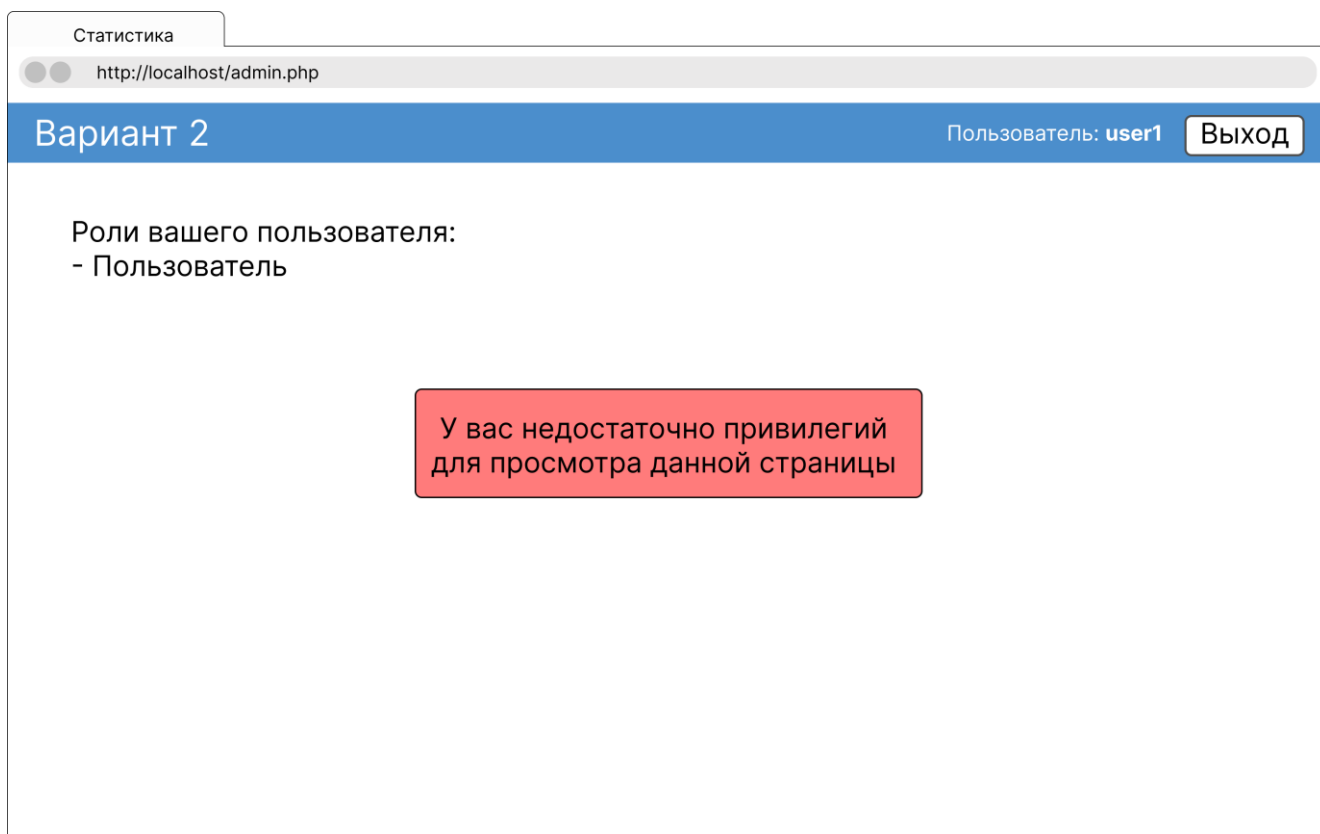


Рисунок 2.15 – Недостаточно привилегий

2.5 Шапка страницы

На страницах `index.php` и `admin.php` присутствует шапка в самом верху страницы.

Ширина – 100% страницы.

Слева вывести номер своего варианта.

Справа выводиться логин пользователя `user.name` (сохранить в сессию и доставать оттуда) с жирным выделением.

Кнопка «Выход» сбрасывает сессию пользователя и перенаправляет на `authorization.php` (рисунок 2.16).

На странице `authorization.php` шапку выводить не нужно.



Рисунок 2.16 – Шапка страницы