



Федеральное государственное бюджетное образовательное учреждение высшего образования  
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ  
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА–КОСМОНАВТА А.А. ЛЕОНОВА»

---

**Колледж космического машиностроения и технологий**

**ОТЧЁТ**

**по учебной практике УП.02.01**

по профессиональному модулю

ПМ.02. Разработка и администрирование баз данных

Специальность **09.02.03 «Программирование в компьютерных системах»**

Обучающегося 4 курса группы П1-20 формы обучения очной

**Демьянова Артема Александровича**

Место прохождения практики

Колледж космического машиностроения и технологий  
«Технологического университета»

Срок прохождения практики с «10» ноября 2023 г. по «16» ноября 2023 г.

Руководитель практики: преподаватель \_\_\_\_\_ Д. Н. Никонова  
подпись

Итоговая оценка по практике \_\_\_\_\_

### Дневник прохождения учебной практики

Дата	Содержание работ	Отметка о выполнении
10.11.2023	Выдача заданий на практику	выполнено
10.11.2023	Изучение предметной области индивидуального задания	выполнено
10.11.2023	Построение ER-модели базы данных	выполнено
13.11.2023	Построение ER-модели базы данных	выполнено
13.11.2023	Построение ER-модели базы данных	выполнено
13.11.2023	Разработка таблиц и связей	выполнено
13.11.2023	Разработка таблиц и связей	выполнено
14.11.2023	Разработка таблиц и связей	выполнено
14.11.2023	Разработка хранимых процедур	выполнено
14.11.2023	Разработка хранимых процедур	выполнено
15.11.2023	Разработка хранимых процедур	выполнено
15.11.2023	Разработка клиентского приложения	выполнено
15.11.2023	Разработка клиентского приложения	выполнено
15.11.2023	Разработка клиентского приложения	выполнено
16.11.2023	Разработка клиентского приложения	выполнено
16.11.2023	Разработка клиентского приложения	выполнено
16.11.2023	Подготовка отчёта и презентации	выполнено
16.11.2023	Сдача зачёта по практике	выполнено

Руководитель практики: преподаватель \_\_\_\_\_ Д. Н. Никонова  
подпись

## СОДЕРЖАНИЕ

Изучение предметной области индивидуального задания.....	2
Построение ER-модели базы данных.....	4
Разработка таблиц и связей.....	8
Разработка хранимых процедур.....	11
Разработка клиентского приложения.....	14

## **Изучение предметной области индивидуального задания**

В создании веб-сайтов и веб-приложений для Интернета применяются разнообразные технологии, языки программирования и инструменты для разработки и поддержки проектов. Каждый этап процесса разработки играет важную роль в создании полноценного и эффективного веб-проекта. Ключевыми моментами, на которые стоит обратить особое внимание, являются:

### **1. Фронтенд и бэкенд:**

- Фронтенд отвечает за визуальную составляющую веб-проекта, которую видит пользователь. Фронтенд-разработчики используют языки разметки HTML для структуры, CSS для стилей и JavaScript для взаимодействия с пользователем;
- Бэкенд занимается обработкой данных, взаимодействием с базой данных, и выполнением серверных задач. Разработчики бэкенда используют языки программирования, такие как Node.js, Python, Ruby или PHP.

### **2. Технологии и инструменты**

- Фронтенд-разработчики часто выбирают между фреймворками, такими как React, Angular, Bootstrap и Vue.js для создания динамичных пользовательских интерфейсов;
- Бэкенд-разработчики также могут использовать различные фреймворки, такие как Express.js для Node.js, Django или Flask для Python, Ruby on Rails для Ruby, чтобы ускорить процесс разработки.

### **3. Базы данных: Для хранения и управления данными в веб-разработке используются системы управления базами данных, такие как MySQL, PostgreSQL, MongoDB и другие;**

### **4. Вёрстка и дизайн: Фронтенд-разработчики работают с HTML и CSS, создавая макеты и стили, чтобы сделать веб-сайт привлекательным и функциональным;**

### **5. Развитие и тестирование: Веб-разработка — постоянный процесс обновления и улучшения, требующий тестирования для обеспечения качества;**

6. Развёртывание и хостинг: После завершения разработки веб-проект разворачивается на сервере с использованием различных услуг хостинга и платформ.

Веб-приложение для ведения личных заметок и списков дел — это онлайн платформа, предназначенная для удобной организации и управления важными для пользователя записями о том, что ему необходимо запомнить. Пользователи могут легко создавать заметки, делать списки задач и вести свой рабочий или личный план в любое время и в любом месте, где есть доступ к сети Интернет.

Существует несколько причин, почему веб-приложения для заметок и списков дел становятся всё более популярными. Во-первых, они обеспечивают простоту и удобство использования, позволяя пользователям быстро добавлять, редактировать и удалять свои записи. Отсутствие зависимости от конкретного устройства или местоположения делает их идеальным инструментом для организации жизни в движении.

Во-вторых, веб-приложения для заметок предлагают широкий набор функций, позволяющих пользователю структурировать свои записи. Это может включать в себя возможность создавать категории, добавлять теги, устанавливать приоритеты и даже делить записи на подзадачи. Такой подход обеспечивает максимальную гибкость в организации информации в соответствии с личными потребностями.

Подобно интернет-магазинам, веб-приложения для заметок и списков дел становятся популярными благодаря своей легкости использования и богатому функционалу. Пользователи могут создавать и редактировать свои записи с удовольствием, а обширные возможности кастомизации делают подобные веб-приложения незаменимыми инструментами для эффективного планирования и управления своим временем.

## **Построение ER-модели базы данных**

База данных — это организованная коллекция данных, обычно хранящихся и обрабатываемых с использованием компьютерной системы. База данных может содержать информацию о пользователях, заказах, продуктах, финансах и т.д. Она предназначена для эффективного хранения, обновления и извлечения данных.

ER-модель, также известная как модель сущность-связь, представляет собой графическое изображение структуры базы данных, предназначенное для описания объектов и их взаимосвязей в системе. Эта модель помогает визуализировать структуру данных и логические связи между ними. Существует несколько методов представления ER-диаграмм:

- Нотация Питера Чена: Сущности изображаются прямоугольниками, а связи — ромбами. Овальные формы используются для указания атрибутов;
- Нотация Crows's foot: Сущности представляются прямоугольниками с названием, а атрибуты указываются внутри. Связи между сущностями обозначаются линиями с «птичьими ногами» на концах;
- IDEF1X: Похож на нотацию Crows's foot, но включает дополнительные правила для уточнения связей и атрибутов;
- Баркеровская нотация: Сущности изображаются прямоугольниками, а связи — между ними. Атрибуты указываются непосредственно на связях.

Для представления ER-модели базы данных веб-приложения по ведению заметок и списков дел была выбрана нотация Crows's foot (Рисунок 1).

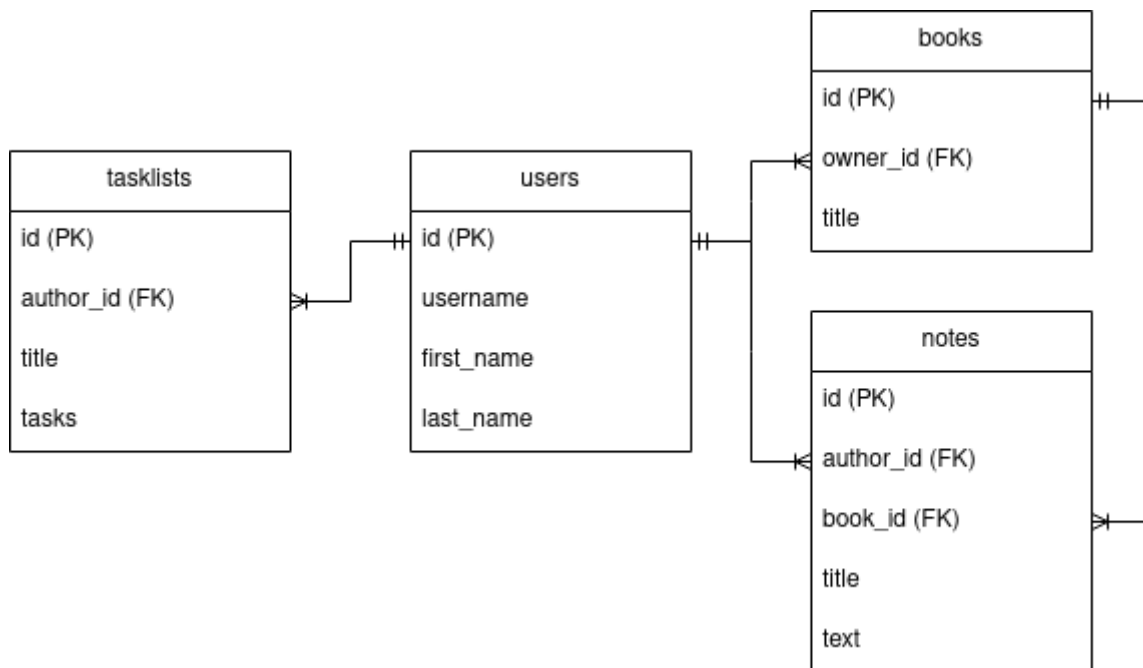


Рисунок 1. ER-модель базы данных

В таблице «users» (Таблица 1) хранится информация о пользователях веб-приложения.

Таблица 1. Users

Наименование столбца	Тип столбца	Описание	Ограничения
id	INT	Уникальный идентификатор пользователя	NOT NULL, PRIMARY KEY
username	VARCHAR(32)	Логин пользователя	
first_name	VARCHAR(64)	Имя пользователя	NOT NULL
last_name	VARCHAR(64)	Фамилия пользователя	

В таблице «books» (Таблица 2) хранится информация о записных книжках пользователей веб-приложения.

Таблица 2. Books

Наименование столбца	Тип столбца	Описание	Ограничения
id	INT	Уникальный идентификатор записной книжки	NOT NULL, PRIMARY KEY
owner_id	INT	Уникальный идентификатор создателя	NOT NULL
title	VARCHAR(64)	Заголовок	NOT NULL

В таблице «notes» (Таблица 3) хранится информация о заметках пользователей веб-приложения.

Таблица 3. Notes

Наименование столбца	Тип столбца	Описание	Ограничения
id	INT	Уникальный идентификатор заметки	NOT NULL, PRIMARY KEY
author_id	INT	Уникальный идентификатор автора	NOT NULL, FOREIGN KEY
book_id	INT	Уникальный идентификатор записной книжки, в которой эта заметка находится	NOT NULL, FOREIGN KEY
title	VARCHAR(64)	Заголовок заметки	NOT NULL
text	TEXT	Текст заметки	NOT NULL



В таблице «tasklists» (Таблица 4) хранится информация о списках дел пользователей веб-приложения.

Таблица 4. Tasklists

Наименование столбца	Тип столбца	Описание	Ограничения
id	INT	Уникальный идентификатор списка дел	NOT NULL, PRIMARY KEY
author_id	INT	Уникальный идентификатор автора	NOT NULL, FOREIGN KEY
title	VARCHAR(64)	Заголовок	NOT NULL
tasks	JSON	Список задач в формате JSON	NOT NULL

## Разработка таблиц и связей

Таблица в базе данных — это структурированный формат для хранения данных в реляционной базе данных. Она представляет собой двумерную сетку, где данные организованы в виде строк и столбцов. Каждая строка таблицы представляет собой запись, а каждый столбец - поле данных.

Связи между таблицами — в реляционных базах данных, данные часто разбиваются на несколько таблиц, которые могут быть связаны между собой. Эти связи позволяют эффективно организовывать и структурировать данные. Вот несколько типов связей:

1. Один к одному (One-to-One): Каждая запись в первой таблице связана с одной и только одной записью во второй таблице, и наоборот.
2. Один ко многим (One-to-Many): Каждая запись в первой таблице может быть связана с несколькими записями во второй таблице, но каждая запись во второй таблице связана только с одной записью в первой.
3. Многие ко многим (Many-to-Many): Множество записей в одной таблице может быть связано с множеством записей в другой таблице. Для реализации такой связи требуется третья таблица, которая содержит ключи из обеих таблиц.

Связи между таблицами обеспечивают целостность данных и позволяют эффективно извлекать и обрабатывать информацию из различных частей базы данных.

Для реализации индивидуального задания по практике была выбрана СУБД MySQL. Основными причинами, по которым она была выбрана, являются:

- Открытый исходный код: MySQL является свободной и открытой СУБД, что означает, что она доступна бесплатно и имеет открытый исходный код. Это может быть привлекательным фактором, особенно для проектов с ограниченным бюджетом.
- Широкая поддержка и сообщество: MySQL имеет обширное сообщество разработчиков и пользователей. Это обеспечивает доступ к множеству ресурсов, включая документацию, форумы, блоги и т.д. Хорошая поддержка

и наличие сообщества могут быть важными факторами при решении о выборе СУБД.

- Производительность: MySQL известен своей высокой производительностью и эффективностью работы с большими объемами данных. Если ваш проект требует быстрого доступа и обработки данных, MySQL может быть привлекательным выбором.
- Надежность и стабильность: MySQL широко используется в индустрии и долгое время существует на рынке. Это может свидетельствовать о его надежности и стабильности, что важно для проектов, где требуется высокая доступность и минимальные сбои.
- Совместимость с различными языками программирования: MySQL обладает хорошей совместимостью с различными языками программирования, такими как PHP, Python, Java и другими. Это может быть важным аспектом при разработке приложений на различных технологических стеках.
- Инструменты администрирования: MySQL поставляется с разнообразными инструментами администрирования, что упрощает управление базой данных, мониторинг и настройку.

Для реализации веб-приложения по ведению заметок и списков дел были реализованы 4 таблицы: «users», «books», «notes», «tasklists». Для создания таблиц используются запросы. Запросы представляют собой специальные команды или инструкции, которые обрабатываются системой управления базой данных (СУБД) и позволяют взаимодействовать с данными в базе.

```
CREATE TABLE IF NOT EXISTS `users` (  
  `id` INT NOT NULL,  
  `username` VARCHAR(32),  
  `first_name` VARCHAR(64) NOT NULL,  
  `last_name` VARCHAR(64),  
  PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE IF NOT EXISTS `books` (
```

```

        `id` INT NOT NULL AUTO_INCREMENT,
        `owner_id` INT NOT NULL,
        `title` VARCHAR(64) NOT NULL,
        PRIMARY KEY (`id`),
        FOREIGN KEY (`owner_id`) REFERENCES `users` (`id`) ON DELETE
CASCADE
    );
CREATE TABLE IF NOT EXISTS `notes` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `author_id` INT NOT NULL,
    `book_id` INT NOT NULL,
    `title` VARCHAR(64) NOT NULL,
    `text` TEXT NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`author_id`) REFERENCES `users` (`id`) ON DELETE
CASCADE,
    FOREIGN KEY (`book_id`) REFERENCES `books` (`id`) ON DELETE
CASCADE
    );
CREATE TABLE `tasklists` (
    id INT AUTO_INCREMENT NOT NULL,
    author_id INT NOT NULL,
    title VARCHAR(64) NOT NULL,
    tasks JSON NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (author_id) REFERENCES users(id) ON DELETE
CASCADE ON UPDATE RESTRICT
    );

```

## Разработка хранимых процедур

Хранимые процедуры в MySQL представляют собой набор инструкций SQL, объединенных в один блок, который сохраняется в базе данных для последующего выполнения. Они позволяют создавать многократно используемые и параметризованные блоки кода, что упрощает обслуживание и улучшает безопасность базы данных.

Для удобства работы администратора были реализованы следующие хранимые процедуры:

1. Добавление записной книжки:

```
DELIMITER //  
  
CREATE PROCEDURE AddBook(  
    IN owner_id_param INT,  
    IN book_title VARCHAR(64)  
)  
BEGIN  
    INSERT INTO books (owner_id, title)  
    VALUES (owner_id_param, book_title);  
END //  
DELIMITER ;
```

2. Создание заметки:

```
DELIMITER //  
  
CREATE PROCEDURE CreateNote(  
    IN author_id_param INT,  
    IN book_id_param INT,  
    IN note_title VARCHAR(64),  
    IN note_text TEXT  
)  
BEGIN  
    INSERT INTO notes (author_id, book_id, title, text)  
    VALUES (author_id_param, book_id_param, note_title, note_text);
```

```
END //
```

```
DELIMITER ;
```

3. Создание списка дел:

```
DELIMITER //
```

```
CREATE PROCEDURE AddTasklist(
```

```
    IN author_id_param INT,
```

```
    IN tasklist_title VARCHAR(64),
```

```
    IN tasklist_tasks JSON
```

```
)
```

```
BEGIN
```

```
    INSERT INTO tasklists (author_id, title, tasks)
```

```
    VALUES (author_id_param, tasklist_title, tasklist_tasks);
```

```
END //
```

```
DELIMITER ;
```

4. Обновление информации о пользователе:

```
DELIMITER //
```

```
CREATE PROCEDURE UpdateUserInfo(
```

```
    IN user_id_param INT,
```

```
    IN new_username VARCHAR(32),
```

```
    IN new_first_name VARCHAR(64),
```

```
    IN new_last_name VARCHAR(64)
```

```
)
```

```
BEGIN
```

```
    UPDATE users
```

```
    SET username = new_username, first_name = new_first_name, last_name =
```

```
new_last_name
```

```
    WHERE id = user_id_param;
```

```
END //
```

```
DELIMITER ;
```

5. Удаление книги и связанных заметок:

```
DELIMITER //
CREATE PROCEDURE DeleteBookAndNotes(
    IN book_id_param INT
)
BEGIN
    DELETE FROM books WHERE id = book_id_param;
END //
DELIMITER ;
```

6. Получение списка задач пользователя:

```
DELIMITER //
CREATE PROCEDURE GetUserTasklist(
    IN user_id_param INT
)
BEGIN
    SELECT tasks FROM tasklists WHERE author_id = user_id_param;
END //
DELIMITER ;
```

7. Поиск пользователей по имени:

```
DELIMITER //
CREATE PROCEDURE SearchUsersByName(
    IN partial_name VARCHAR(64)
)
BEGIN
    SELECT * FROM users WHERE first_name LIKE CONCAT('%',
partial_name, '%') OR last_name LIKE CONCAT('%', partial_name, '%');
END //
DELIMITER ;
```

## Разработка клиентского приложения

Для создания веб-приложения был выбран язык программирования Python, фреймворк Flask для бэкенда и Bootstrap для фронтенда. Python является высокоуровневым и интерпретируемым языком программирования, с акцентом на читаемость кода и простоту синтаксиса. Flask, в свою очередь, представляет собой легковесный инструмент для создания веб-приложений на языке Python, предоставляя базовые средства для этой цели. Bootstrap предоставляет набор инструментов, содержащих HTML, CSS и JavaScript компоненты, которые упрощают создание современных и отзывчивых пользовательских интерфейсов для веб-приложений и сайтов.

На главной странице пользователю рассказывается о возможностях данного приложения и предлагается войти в свою учётную запись (Рисунок 2).

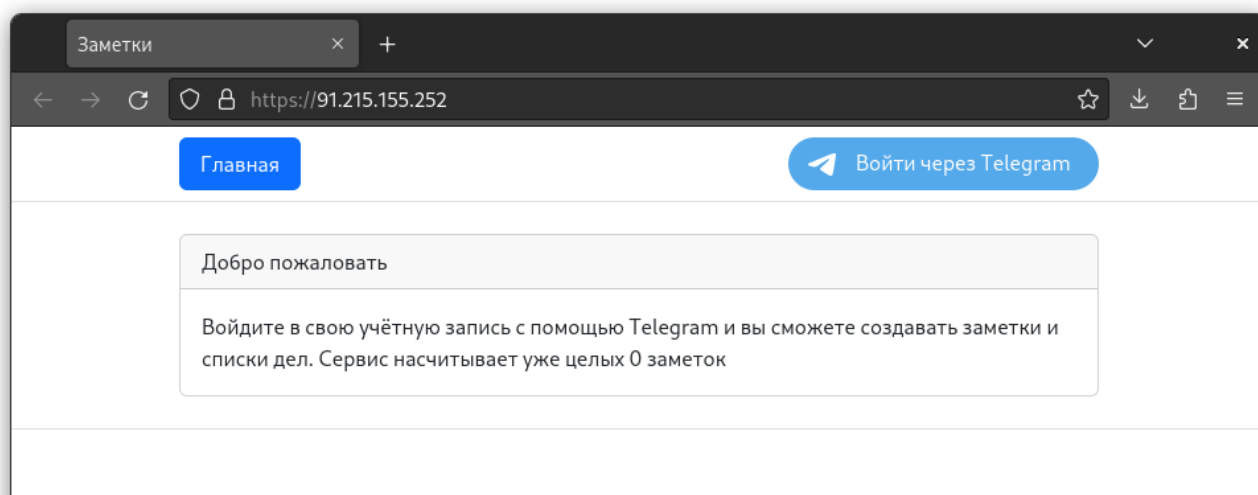


Рисунок 2. Главная страница



После входа в аккаунт пользователю доступны вкладки «Заметки» (Рисунок 3), «Списки дел» (Рисунок 4) и «Выход».

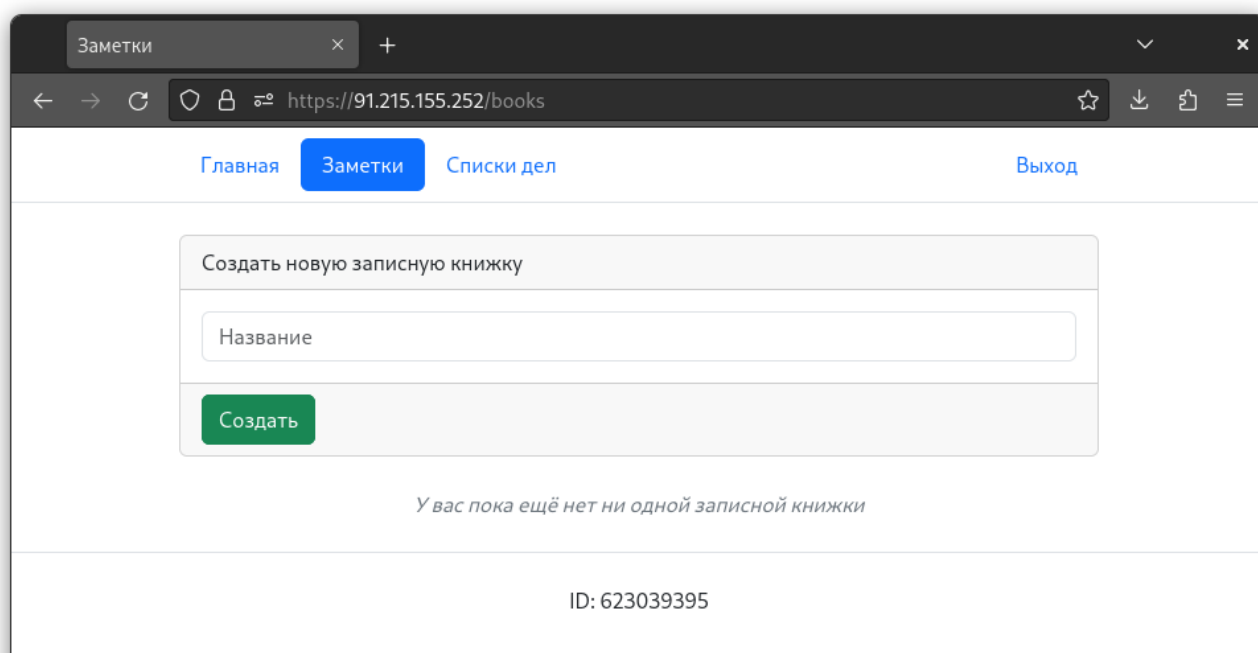


Рисунок 3. Страница с перечислением всех имеющихся у пользователя записных книжек

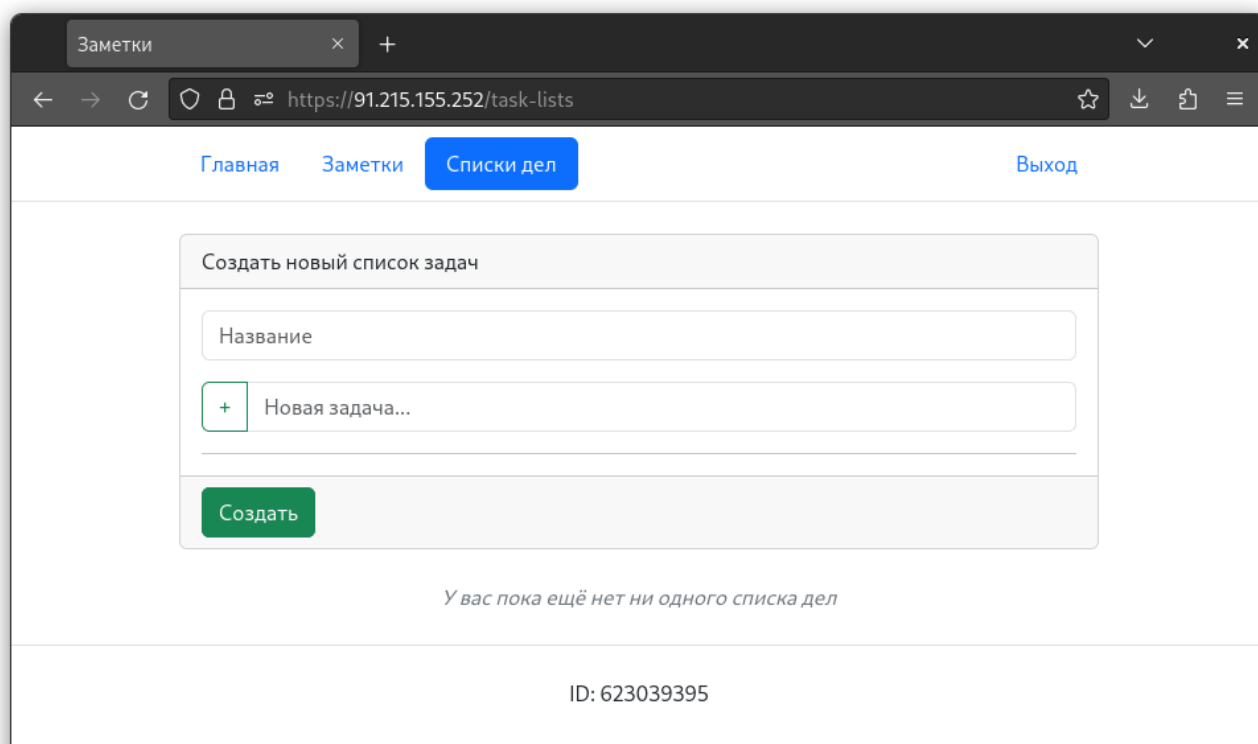


Рисунок 4. Страница с перечислением всех имеющихся у пользователя списков дел

Для создания новой заметки пользователю необходимо создать новую записную книжку. Для этого необходимо перейти на вкладку «Заметки», ввести название записной книжки и нажать на кнопку «Создать». После этого новая записная книжка отобразится на этой же странице (Рисунок 5 и Рисунок 6).

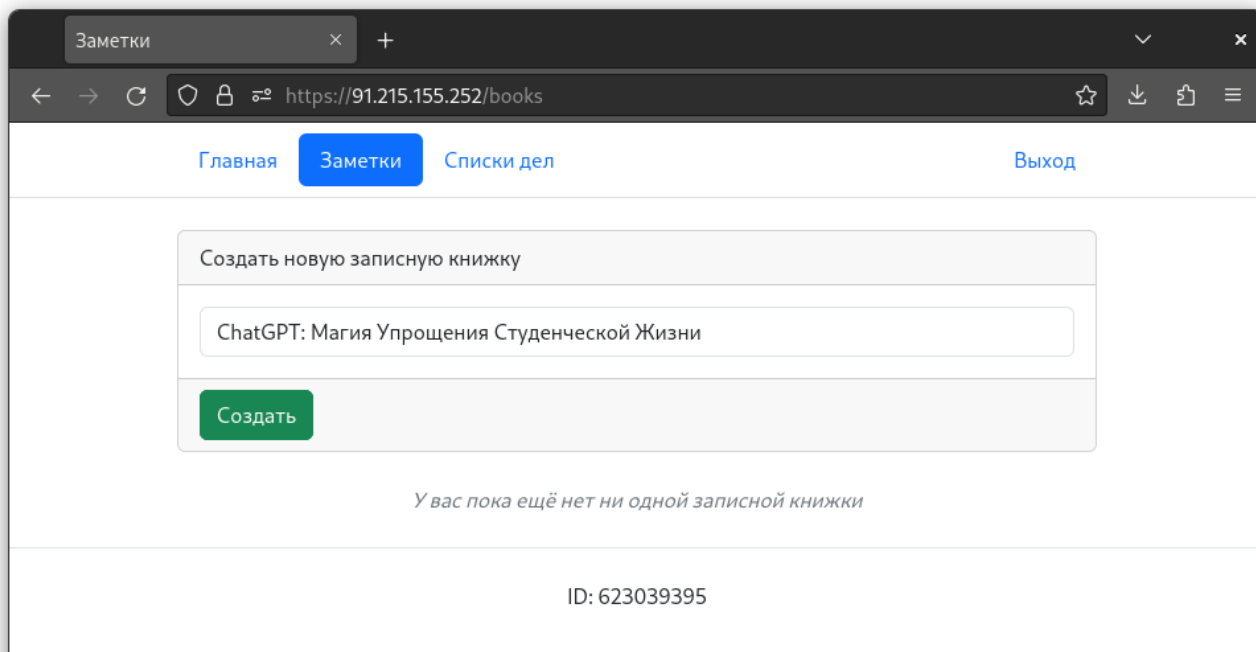


Рисунок 5. Создание новой записной книжки

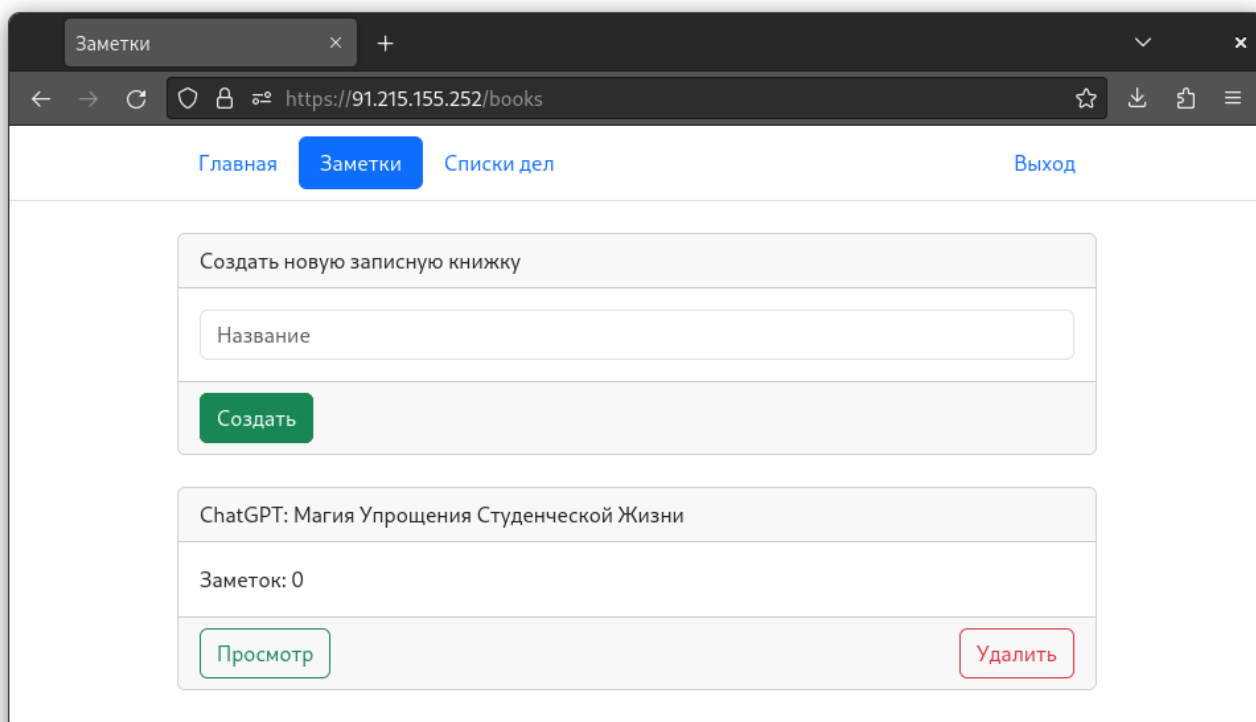


Рисунок 6. Новая записная книжка

По нажатию на кнопку «Просмотр» пользователь перейдёт в выбранную записную книжку (Рисунок 7). В ней отображаются все заметки, которые пользователь оставил ранее, а в самом верху страницы есть форма для создания новой заметки в этой записной книжке.

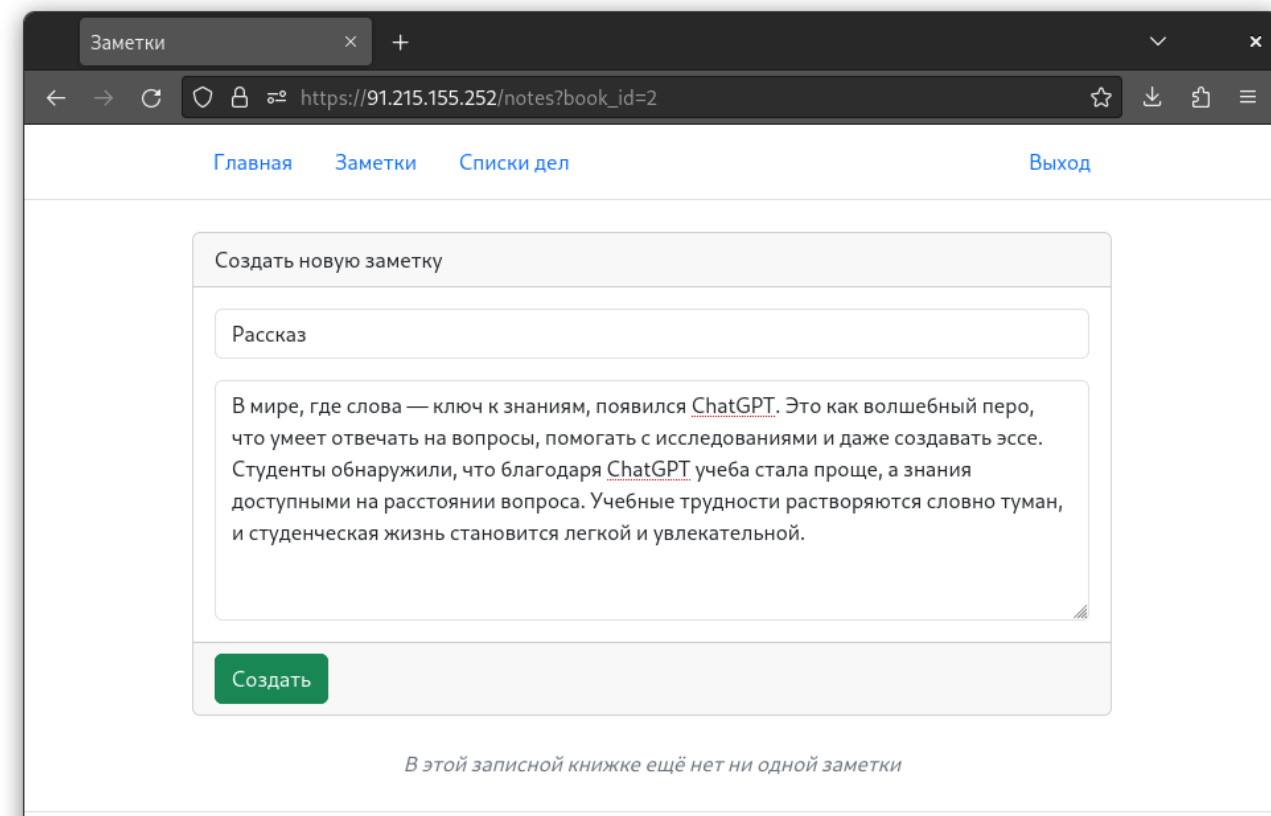


Рисунок 7. Создание новой заметки

По нажатию на кнопку «Создать» новая заметка будет добавлена в записную книжку (Рисунок 8).

The screenshot shows a web browser window with the title 'Заметки' (Notes). The address bar shows the URL 'https://91.215.155.252/notes?book\_id=2'. The page has a navigation bar with links: 'Главная' (Home), 'Заметки' (Notes), 'Списки дел' (To-do lists), and 'Выход' (Logout). The main content area is titled 'Создать новую заметку' (Create a new note). It contains two input fields: 'Заголовок' (Title) and 'Текст' (Text). Below the text field is a green button labeled 'Создать' (Create). Below this is a section titled 'Рассказ' (Story) containing a paragraph of text about ChatGPT. At the bottom of this section is a red button labeled 'Удалить' (Delete). The footer of the page displays the ID: 623039395.

Рисунок 8. Новая заметка в записной книжке

По нажатию на кнопку «Создать» эта заметка будет сохранена и в дальнейшем будет отображаться у пользователя, пока он её не удалит.

Для создания нового списка дел пользователю необходимо перейти на вкладку «Списки дел», ввести название списка и добавить задачи (Рисунок 9).

The screenshot shows a web browser window with the address bar displaying `https://91.215.155.252/task-lists`. The page has a dark-themed header with navigation links: «Главная», «Заметки», «Списки дел» (highlighted in blue), and «Выход». Below the header, there is a form titled «Создать новый список задач». The form contains a text input field with the placeholder «Список для чтения», a green «+» button, and a text input field with the placeholder «Новая задача...». Below these, there is a list of tasks, each with a text input field and a red «-» button for deletion. The tasks listed are: «1984 (Джордж Оруэлл)», «Мы (Евгений Замятин)», «451 градус по Фаренгейту (Рэй Брэдбери)», and «О дивный новый мир (Олдос Хаксли)». At the bottom of the form is a green «Создать» button. Below the form, there is a message: «У вас пока ещё нет ни одного списка дел». At the very bottom of the page, the ID «ID: 623039395» is displayed.

Рисунок 9. Создание нового списка дел

По нажатию на кнопку «Создать» этот список дел будет сохранён и в дальнейшем будет отображаться у пользователя, пока он его не удалит.