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

Колледж космического машиностроения и технологий

ОТЧЕТ

по учебной практике УП.03.01

по профессиональному модулю
ПМ.03. Участие в интеграции программных модулей

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 «17» ноября 2023 г. по «23» ноября 2023 г.

Руководитель практики: преподаватель _____ О. С. Лихторенко
подпись

Итоговая оценка по практике _____

Дневник прохождения учебной практики

Дата	Содержание работ	Отметка о выполнении
	Выдача заданий на практику	выполнено
	Разработка требований и технического задания к программе в соответствии с выданной темой	выполнено
	Проектирование графического интерфейса	выполнено
	Построение USE CASE-диаграмм к программе в соответствии с выданной темой	выполнено
	Построение DFD-диаграмм к программе в соответствии с выданной темой	выполнено
	Построение инфологической и даталогической модели к программе в соответствии с выданной темой	выполнено
	Построение UML-диаграмм к программе в соответствии с выданной темой	выполнено
	Разработка программы в соответствии с выданной темой	выполнено
	Тестирование, отладка, оптимизация программы в соответствии с выданной темой	
	Подготовка отчета и презентации	
	Сдача зачета по практики	

Руководитель практики: преподаватель _____ О. С. Лихторенко
подпись

СОДЕРЖАНИЕ

Разработка требований и технического задания к программе в соответствии с выданной темой

1. **Введение**

Настоящий документ представляет введение в программный продукт с наименованием «WebNotes» — веб-приложение, спроектированное для эффективного ведения заметок и управления списками дел.

«WebNotes» представляет собой веб-приложение, разработанное с целью обеспечения удобства и эффективности в процессе ведения записей и управления задачами. Данное многофункциональное приложение призвано оптимизировать процессы организации информации, повышая при этом производительность и систематизацию рабочих задач.

«WebNotes» предназначено для широкого спектра пользователей, включая студентов, предпринимателей и специалистов, которые стремятся к эффективному ведению записей и управлению своими делами. Приложение предоставляет интуитивно понятный интерфейс для создания, редактирования и систематизации заметок и списков дел, а также обеспечивает возможность доступа к ним с любого устройства, обладающего подключением к сети Интернет.

«WebNotes» не только обеспечивает средства для структурирования информации, но также внедряет современные технологии синхронизации данных, обеспечивая тем самым максимальную удобство и гибкость в использовании приложения.

В последующих разделах данного документа рассмотрены ключевые функциональные возможности, системные требования, а также преимущества использования веб-приложения «WebNotes».

2. Основания для разработки

Документы, лежащие в основе разработки программы «WebNotes», были утверждены 17 ноября 2023 года организацией Колледж космического машиностроения и технологий «Технологического университета».

3. Назначение разработки

Программа «WebNotes» разрабатывается с целью предоставления пользователям современного, удобного и функционального веб-приложения для ведения заметок и управления списками дел. Основные функциональные задачи включают:

- Создание заметок: Пользователи могут легко создавать новые заметки в соответствии с индивидуальными потребностями;
- Управление списками дел: «WebNotes» предоставляет возможность создания списков дел, отметки выполненных задач и эффективного мониторинга текущих заданий;
- Синхронизация данных: Приложение обеспечивает современные механизмы синхронизации данных, позволяя пользователям получать доступ к своим заметкам и спискам дел с любого устройства, подключенного к сети Интернет;
- Интуитивно-понятный интерфейс: «WebNotes» обладает пользовательским интерфейсом, который легко осваивается, обеспечивая удобство в использовании даже для новых пользователей.

4. Требования к программе или программному изделию

4.1. Выполняемые функции:

- Аутентификация и авторизация пользователей через внешний сервис — Telegram;
- Управление списками дел и записными книжками, включая их добавление и удаление.

4.2. Требования к надежности:

- Устойчивость к сбоям, вводу неверной информации;
- Контроль входной информации при добавлении списков дел, записных книжек и записей в них;
- Обеспечение защиты данных от несанкционированного доступа на основе процедуры аутентификации пользователей и разграничения прав доступа к данным.

4.3. Требования к составу и параметрам технических средств:

- Интерпретатор Python (последняя версия), фреймворк Flask, СУБД MySQL (последняя версия);
- Объем хранилища не менее 5 ГБ;
- Объем ОЗУ не менее 4 ГБ;
- Частота процессора не менее 1.9 ГГц;

4.4. Требования к информационной и программной совместимости:

Приложение должно быть реализовано с использованием фреймворка Python Flask и должно быть совместимо с операционными системами Linux. База данных должна работать под управлением СУБД MySQL.

5. Требования к программной документации

Для успешной разработки, внедрения и последующей поддержки программы «WebNotes» предполагается формирование следующего предварительного состава программной документации:

- Техническое задание: Документ, содержащий основные требования к функциональности приложения, описание интерфейса пользователя и критерии приемки;
- Руководство пользователя: Инструкции по использованию «WebNotes». Включает в себя пошаговые рекомендации и советы для конечных пользователей;
- Руководство по администрированию: Документ, предоставляющий информацию для системных администраторов по установке, настройке и обслуживанию приложения.

6. Технико-экономические показатели

Технико-экономические показатели веб-приложения соответствует ожидаемым и обеспечивают выполнение следующих требований:

- Веб-приложение легко модифицируется при изменении требований;
- Система имеет дружественный пользователю интерфейс.

7. Стадии и этапы разработки

7.1. Исследование и анализ требований;

7.2. Планирование проекта;

7.3. Архитектурная стадия:

- Определение языков программирования и фреймворков;
- Выбор инструментов для разработки;
- Разработка архитектуры приложения;
- Согласование интерфейсов и базы данных;

7.4. Разработка:

- Фронтенд:
  - Создание пользовательского интерфейса;
  - Разработка клиентской логики.
- Бэкенд:
  - Реализация серверной части;
  - Интеграция с базой данных.

7.5. Тестирование и отладка;

7.6. Документирование;

7.7. Релиз и внедрение:

- Подготовка серверной инфраструктуры;
- Запуск приложения.

7.8. Внесение исправлений и обновлений.

8. Порядок контроля и приемки

Для проверки работы веб-приложения должны быть разработаны тестовые примеры. Веб-приложение должно быть протестировано на возможные реакции пользователя: запуск веб-приложения, заполнение полей ввода, выбор пункта меню, работа с базой данных, активизация кнопок.

Приемка веб-приложения должна осуществляться путем анализа его работоспособности при использовании в качестве входных данных тестовых наборов данных, при правильном выполнении возложенных на него функций и их соответствия требованиям технического задания и наличия пояснительной записки, включающей полный комплект программной документации.

Проектирование графического интерфейса

Для разработки графического интерфейса веб-приложения в соответствии с заданной темой был выбран Bootstrap. Это открытый и свободно распространяемый набор инструментов для разработки веб-приложений. Он предоставляет готовые стили, компоненты и скрипты, упрощая создание современных и адаптивных пользовательских интерфейсов. Bootstrap основан на HTML, CSS и JavaScript, что делает его идеальным выбором для разработчиков, стремящихся к быстрой и качественной реализации интерфейса своего веб-приложения.

Bootstrap предоставляет возможность легко создавать адаптивные веб-приложения, и «WebNotes» не станет исключением. Дизайн графического интерфейса будет оптимизирован для просмотра на различных устройствах, что обеспечит приятный пользовательский опыт как на компьютере с большим экраном, так и на мобильных устройствах с более мелкими экранами.

Каждая страница веб-приложения будет содержать панель навигации, в которой будут следующие разделы: «Главная», «Заметки», «Списки дел».

Главная страница веб-приложения «WebNotes» будет содержать приветственное окно, в котором показываются данные о пользователе, который использует веб-приложение на данный момент.

В разделе «Заметки» пользователь может просмотреть все свои записные книжки, удалить их или создать новую. Так же здесь можно перейти в записную книжку, и тогда пользователь увидит все заметки, что хранятся в ней.

В разделе «Списки дел» пользователь увидит все свои списки дел, и сможет отметить уже выполненные задачи, поставив галочку напротив соответствующей задачи и нажав кнопку «Сохранить». Также вверху страницы будет находиться форма для создания нового списка дел.

Построение USE CASE-диаграмм к программе в соответствии с выданной темой

Перед началом разработки веб-приложения «WebNotes», необходимо построить USE CASE-диаграмму, представляющую взаимодействие с приложением гостей и авторизованных пользователей (Рисунок 1).

Первый актор диаграммы охватывает сценарии использования для гостей — тех, кто посещает приложение без предварительной авторизации. Гости могут просматривать главную страницу и выполнить вход в свой аккаунт.

Второй актор описывает взаимодействие авторизованных пользователей на диаграмме. Здесь представлены различные сценарии использования, такие как создание и удаление заметок, управление списками дел, а также выход из аккаунта. Важным аспектом является обеспечение удобства пользователя в процессе взаимодействия с веб-приложением, что включает в себя интуитивно понятные графические интерфейсы для всех функций.

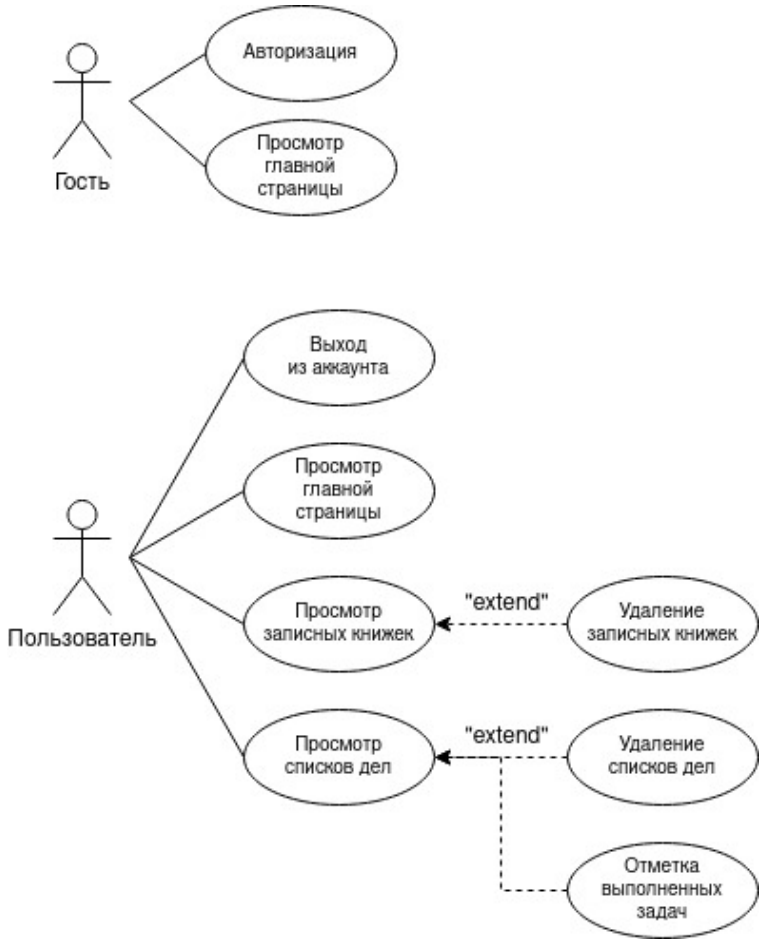

Рисунок 1. USE CASE-диаграмма

Все действия акторов спроектированы с учетом потребностей и ожиданий конечных пользователей, стремясь сделать пользовательский опыт максимально эффективным и удовлетворительным. Они служат важным инструментом для разработки, обеспечивая четкое представление об основных сценариях использования и взаимодействия пользователя с веб-приложением «WebNotes».

Построение DFD-диаграмм к программе в соответствии с выданной темой

DFD (Data Flow Diagram), или диаграмма потоков данных, представляет собой графическую модель, используемую для описания потоков данных в системе и их обработки. DFD включает в себя графические элементы, такие как процессы, данные, потоки данных и хранилища данных, а также связи между ними. Он помогает визуализировать, как данные перемещаются через систему, как они обрабатываются и какие изменения происходят на каждом этапе. DFD диаграмма данного веб-приложения представлена ниже (Рисунок 2).

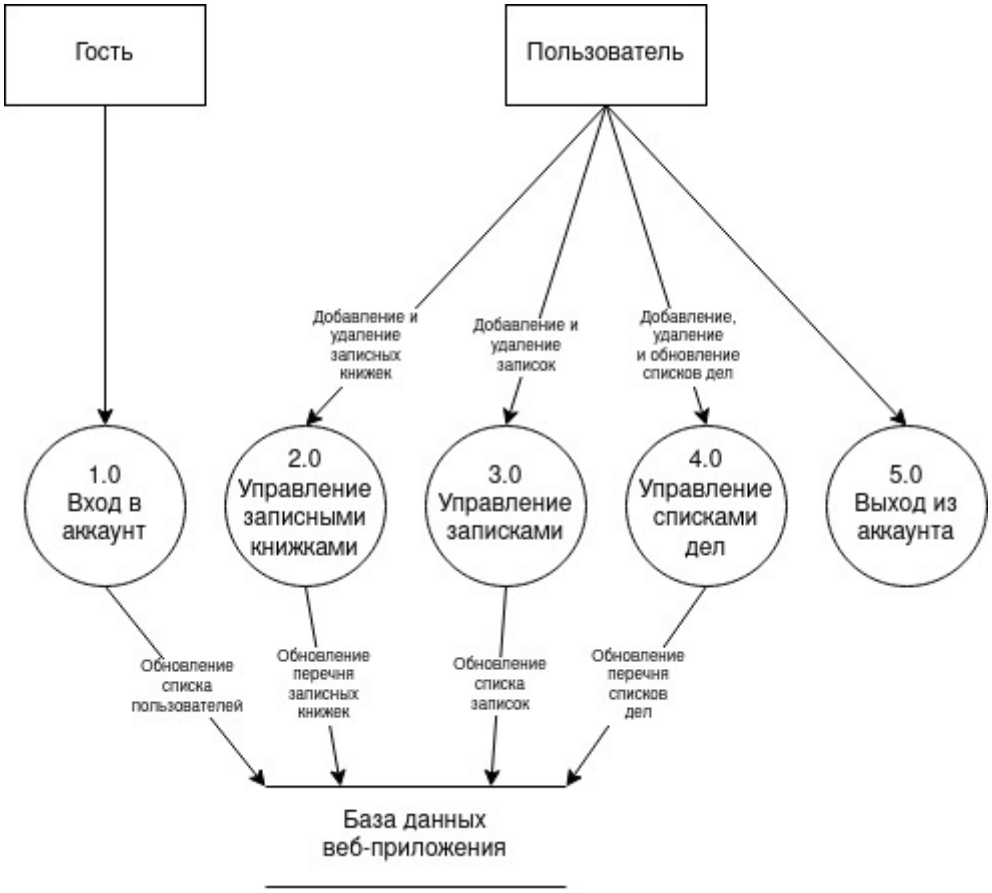

Рисунок 2. DFD-диаграмма

Построение инфологической и даталогической модели к программе в соответствии с выданной темой

Инфологическая модель описывает структуру данных независимо от конкретных технических решений или хранилищ данных. Ее цель — представить концептуальное представление данных и их взаимосвязей. Это высокоуровневая абстракция, которая фокусируется на сущностях и их взаимосвязях в предметной области, независимо от того, как они будут физически реализованы. Инфологическая модель веб-приложения «WebNotes» показана ниже (Рисунок 3).

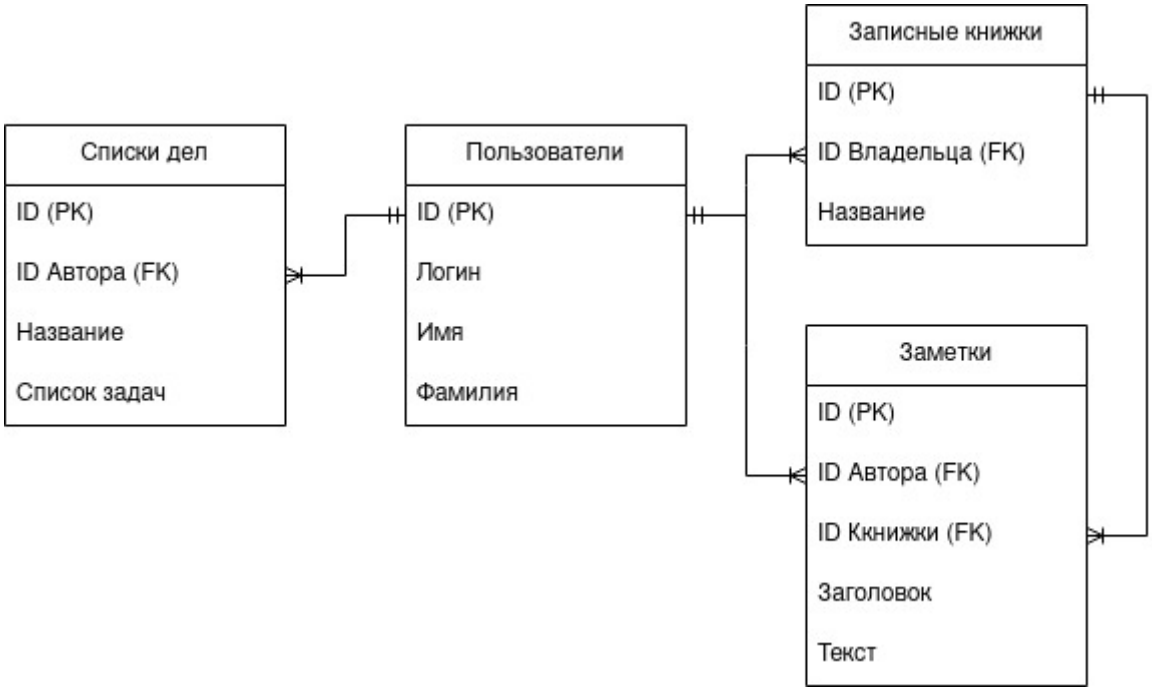

Рисунок 3. Инфологическая модель БД

Даталогическая модель более конкретна и описывает структуру данных с учетом технических аспектов, таких как типы данных, индексы и ограничения целостности. Она является промежуточным звеном между концептуальной моделью и физической реализацией. Это средний уровень абстракции, который связывает концептуальные и физические аспекты данных. Она учитывает технические детали, но все еще остается относительно независимой от конкретной реализации базы данных. Даталогическая модель веб-приложения «WebNotes» показана ниже (Рисунок 4).

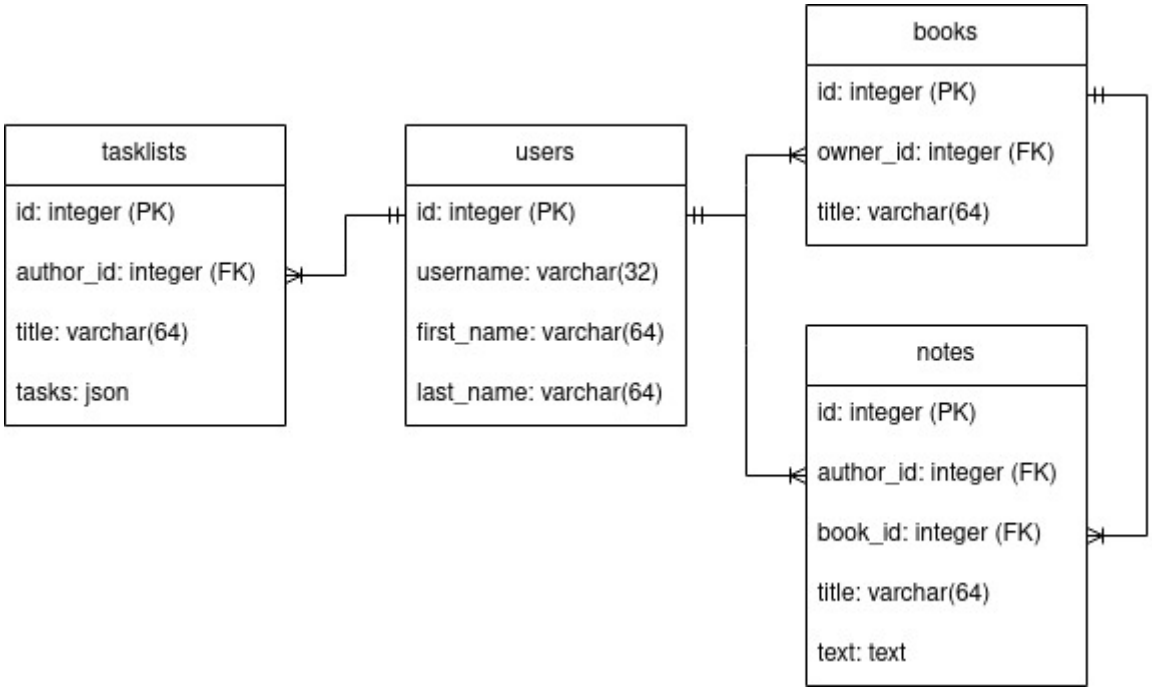

Рисунок 4. Даталогическая модель БД

Построение UML-диаграмм к программе в соответствии с выданной темой

UML-диаграммы представляют собой графические модели, используемые в области разработки программного обеспечения для визуализации, проектирования и документирования структуры и поведения системы. Аббревиатура «UML» означает Unified Modeling Language, что в переводе с английского можно интерпретировать как «Единый язык моделирования». UML предоставляет стандартизированный набор графических обозначений и правил для создания моделей, которые облегчают взаимопонимание между членами команды разработчиков, заказчиками и другими заинтересованными сторонами.

Диаграммы UML позволяют разработчикам визуализировать различные аспекты системы, такие как ее структура, взаимодействия между компонентами, поведение системы и другие аспекты. Эти модели помогают улучшить понимание требований к системе, а также облегчают коммуникацию внутри команды разработчиков. Кроме того, UML-диаграммы служат важным инструментом для создания документации, что способствует более эффективному управлению проектом и обеспечивает легкость сопровождения системы на всех этапах ее жизненного цикла.

UML-диаграмма к веб-приложению в соответствии с заданной темой изображена ниже (Рисунок 5).

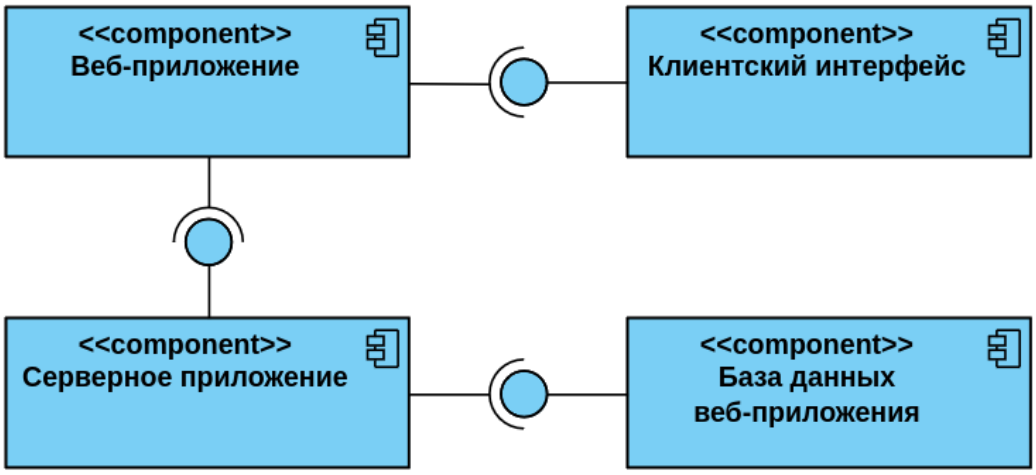

Рисунок 5. UML-диаграмма веб-приложения

Разработка программы в соответствии с выданной темой

Для разработки веб-приложения в соответствии с выданной темой был выбран язык программирования Python и фреймворк для разработки веб-приложений Flask.

Python — высокоуровневый, интерпретируемый язык программирования, изначально разработанный Гвидо ван Россумом в конце 1980-х годов. Он отличается простотой синтаксиса, читаемостью кода и обширной стандартной библиотекой, что делает его популярным выбором для различных задач, включая веб-разработку, анализ данных, искусственный интеллект и автоматизацию. Python также поддерживает различные парадигмы программирования, включая объектно-ориентированное, процедурное и функциональное программирование, что способствует его гибкости и универсальности.

Flask — легкий и гибкий веб-фреймворк для языка программирования Python. Разработанный Армином Ронхайзером, Flask обеспечивает минималистичный подход к созданию веб-приложений, предоставляя базовый набор инструментов для построения веб-сайтов и веб-приложений. Flask не навязывает строгую структуру приложения, что дает разработчикам свободу выбора инструментов и архитектуры. Он основывается на принципах Werkzeug и Jinja2, обеспечивая простоту использования и расширяемость. Flask поддерживает маршрутизацию, шаблонизацию, обработку форм, а также интеграцию с различными базами данных. Благодаря своей легкости и ясной документации Flask часто используется для создания малых и средних веб-приложений.

Пользовательский интерфейс и функциональность приложения соответствует заданным ранее требованиям. Код приложения представлен в приложении.

Тестирование, отладка, оптимизация программы в соответствии с выданной темой

В качестве метода тестирования был выбран метод «черного ящика», при котором тестирование осуществляется только через пользовательских интерфейс и при котором происходит сопоставление ожидаемых результатов с получаемыми.

Для проверки правильности работы веб-приложения должны быть использованы тестовые примеры, представленные ниже (Таблица 1).

Таблица 1. Тестовые примеры для проверки правильности работы программного средства

№ п/п	Тестовый пример	Описание тестового примера
1	2	3
1	Войти в ранее не существующий аккаунт.	Войти в аккаунт в веб-приложении через сервис Telegram впервые. Аккаунт в базе данных веб-приложения должен создаться автоматически.
2	Войти в ранее существующий аккаунт.	Войти в аккаунт в веб-приложении через сервис Telegram повторно. Данные об аккаунте должны совпадать с сохраненными в базе данных.
3	Создать новую записную книжку с пустой строкой в качестве названия.	Пользователь должен получить предупреждение о недопустимости пустого названия.
4	Создать новую записную книжку с любым не пустым названием.	Записная книжка должна создаться и отобразиться в разделе «Заметки»
5	Создать новый список дел с любым названием и задачами.	Список дел должен создаться и отобразиться в разделе «Списки дел»
6	Отметить задачу как выполненную в любом из созданных списков дел и активизировать кнопку «Сохранить».	Задача должна остаться отмеченной даже после обновления страницы.
7	Выйти из аккаунта.	Активизировать кнопку «Выход» на верхней панели веб-приложения. После этого пользователь должен выйти из своего аккаунта.

В процессе испытаний были получены следующие результаты:

1. На главной странице веб-приложения активизируем кнопку «Войти через Telegram» (Рисунок 6) в появившемся окне вводим номер телефона от аккаунта в Telegram. Затем страница веб-приложения обновится, и пользователю будут доступны разделы «Заметки» и «Списки дел» (Рисунок 7). Так как аккаунт еще не был добавлен в базу данных веб-приложения, то при первом входе данные о пользователе берутся из его профиля в Telegram и заносятся в базу данных веб-приложения.
2. На главной странице веб-приложения, также как и в предыдущем тесте, активизируем кнопку «Войти через Telegram» (Рисунок 6) и в появившемся окне вводим номер телефона от аккаунта в Telegram. После этого страница веб-приложения также обновится, и пользователю будут доступны разделы «Заметки» и «Списки дел» (Рисунок 7). Но в данном случае, так как аккаунт ранее был зарегистрирован в базе данных приложения, данные о нем берутся уже из базы данных, а не из Telegram.
3. Переходим в раздел «Заметки» и в секции «Создать новую записную книжку» поле «Название» оставляем пустым и активизируем кнопку «Создать». Должно появиться предупреждение о том, что нельзя оставить данное поле для ввода пустым (Рисунок 8).
4. Переходим в раздел «Заметки» и в секции «Создать новую записную книжку» поле «Название» заполняем любым текстом и активизируем кнопку «Создать» (Рисунок 9). Страница обновится и созданная записная книжка появится под формой для их создания (Рисунок 10).
5. Переходим в раздел «Списки дел» и в секции «Создать новый список дел» поле «Название» заполняем любым текстом и добавляем несколько задач в список и активизируем кнопку «Создать» (Рисунок 11). Страница обновится и созданный список дел появится под формой для их создания (Рисунок 12).
6. Переходим в раздел «Списки дел» и выбираем любой список. В выбранном списке нажать на чекбокс около любой из задач в нем, затем активизировать

кнопку «Сохранить». Страница обновится и отмеченная задача в выбранном списке останется отмеченной (Рисунок 13).

7. На верхней панели справа активизировать кнопку «Выход». Страница обновится и пользователя перенаправит на главную страницу, после чего ему не будут доступны вкладки «Заметки» и «Списки дел» до тех пор, пока он снова не войдет в свой аккаунт (Рисунок 14).

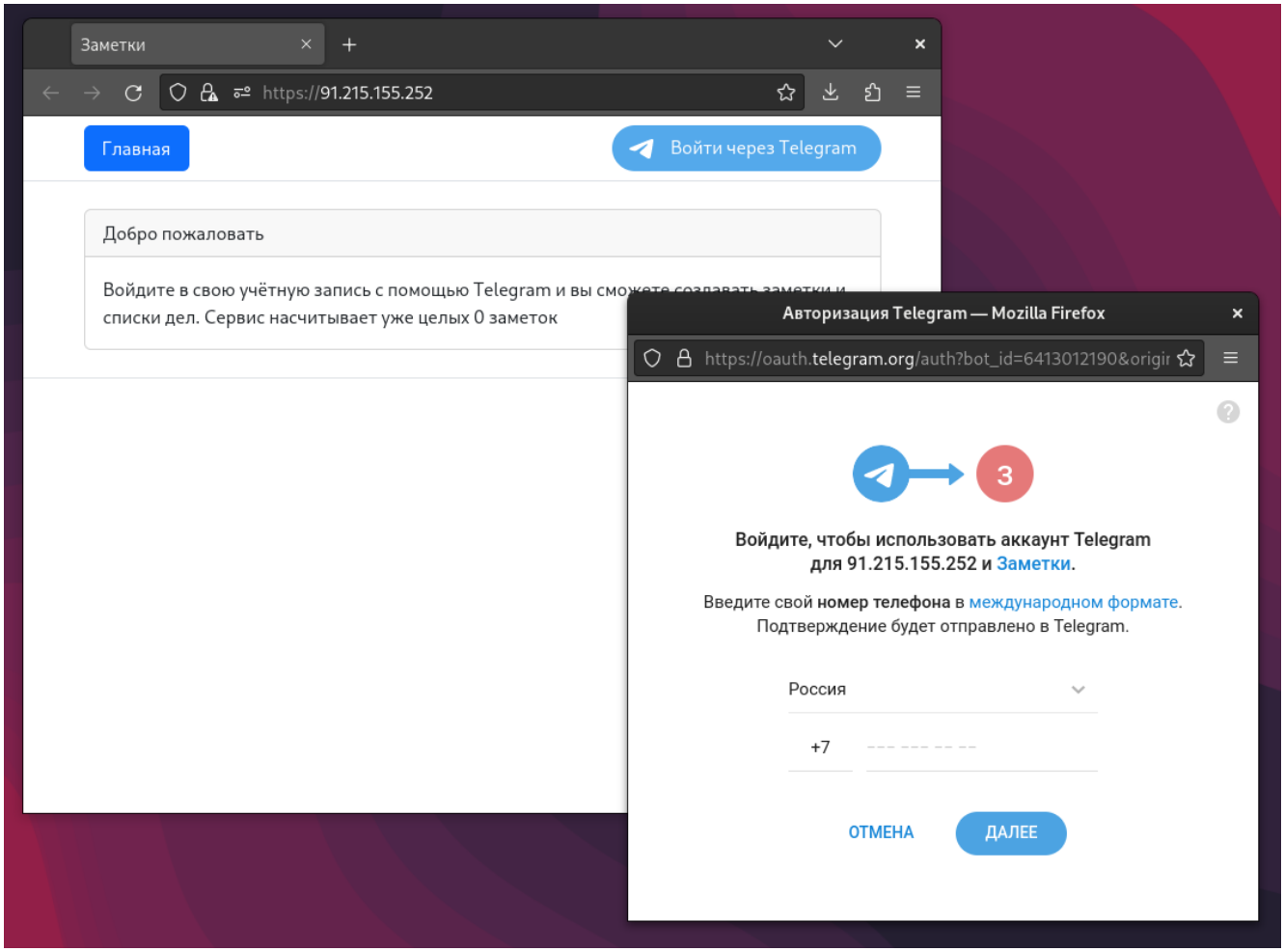

Рисунок 6. Вход в аккаунт

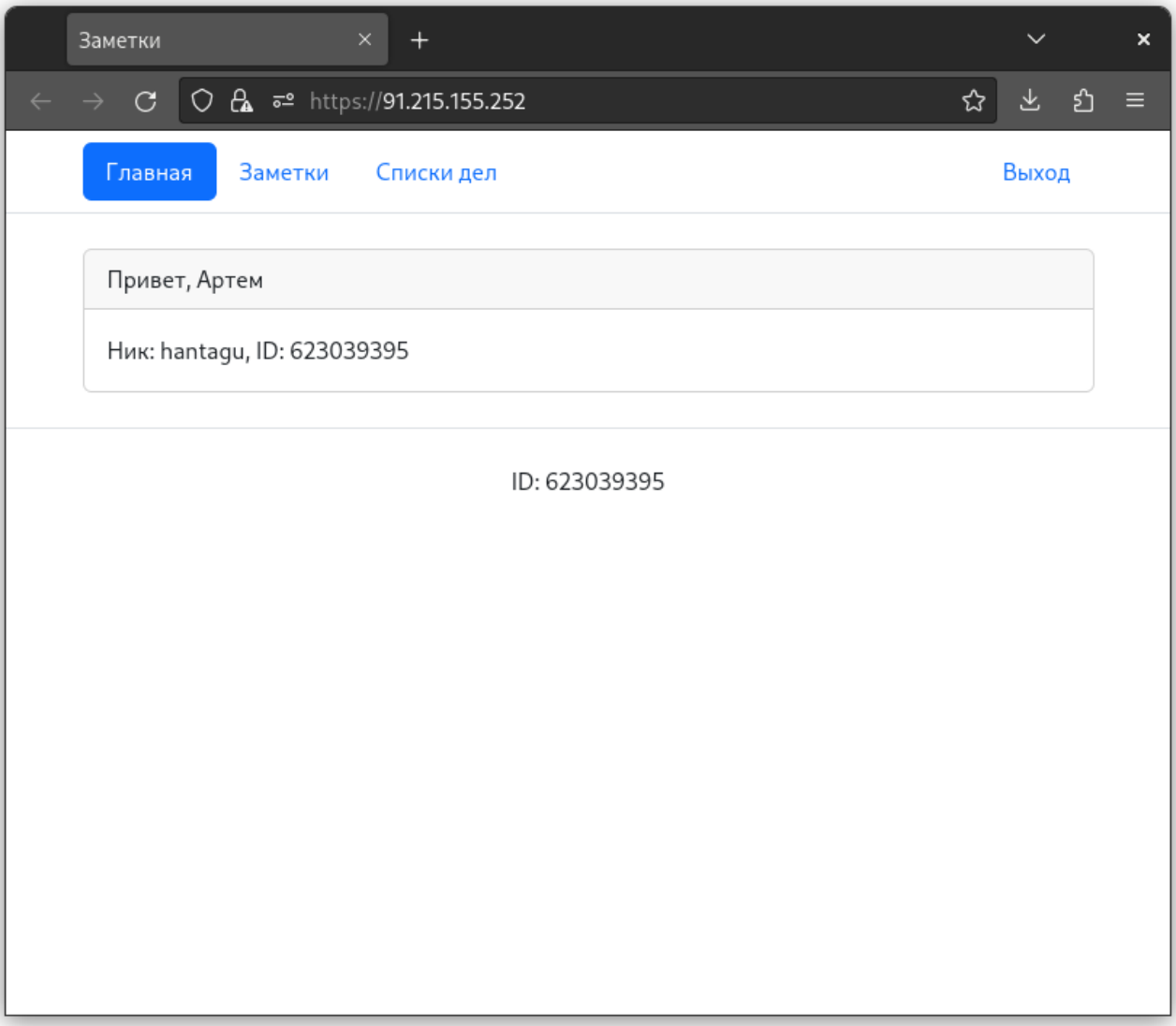

Рисунок 7. Главная страница

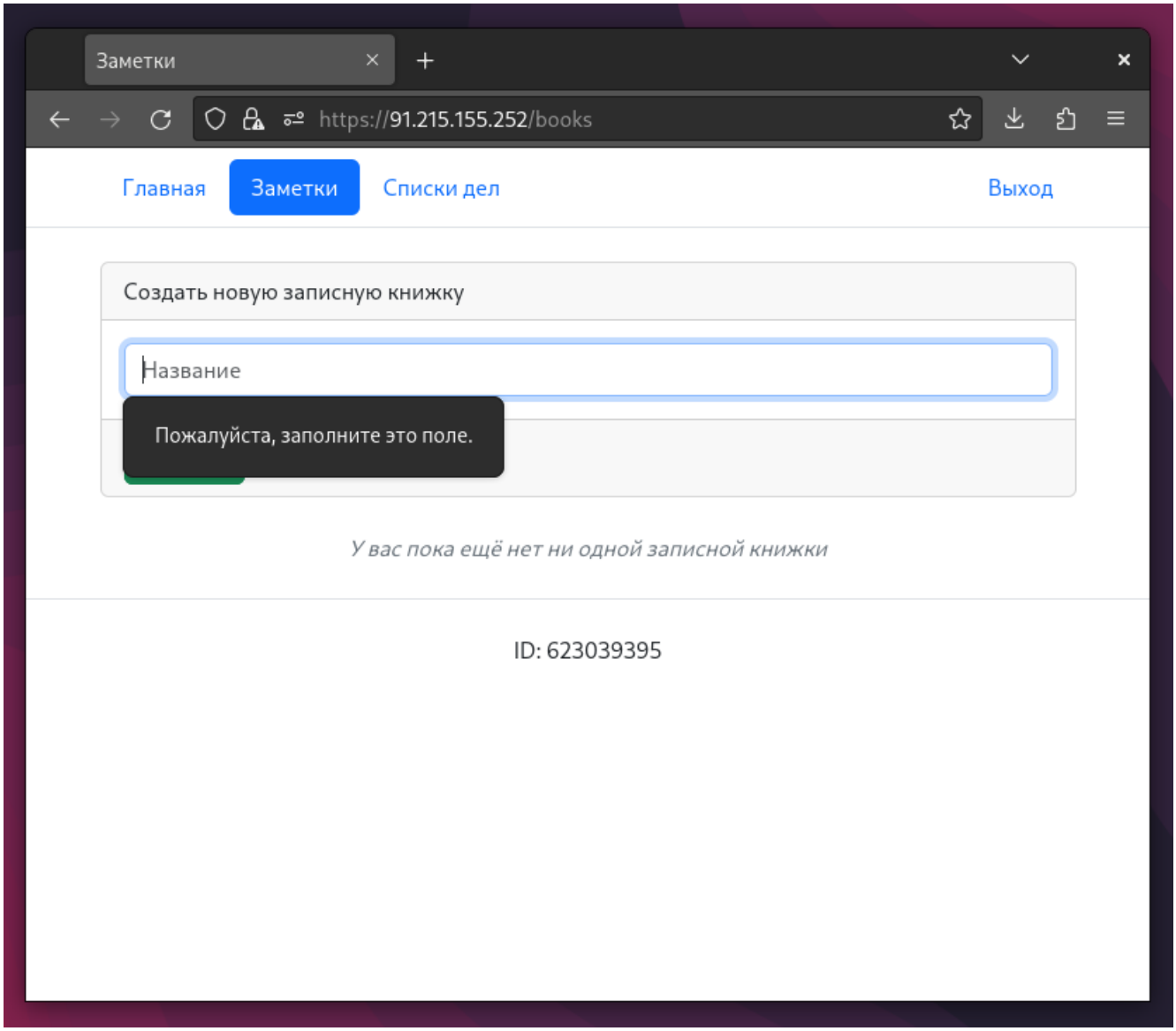

Рисунок 8. Создание записной книжки с пустым названием

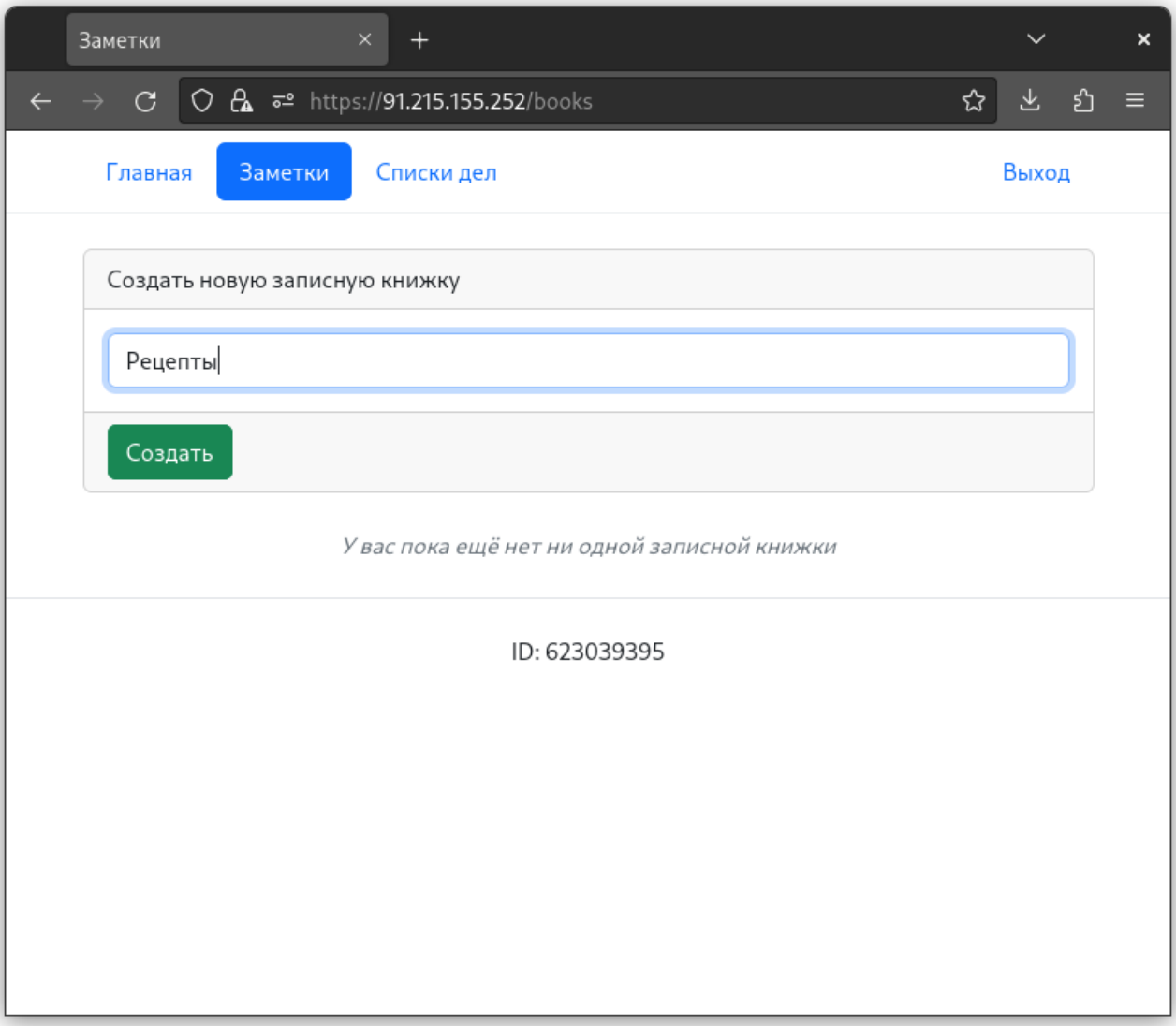

Рисунок 9. Создание записной книжки

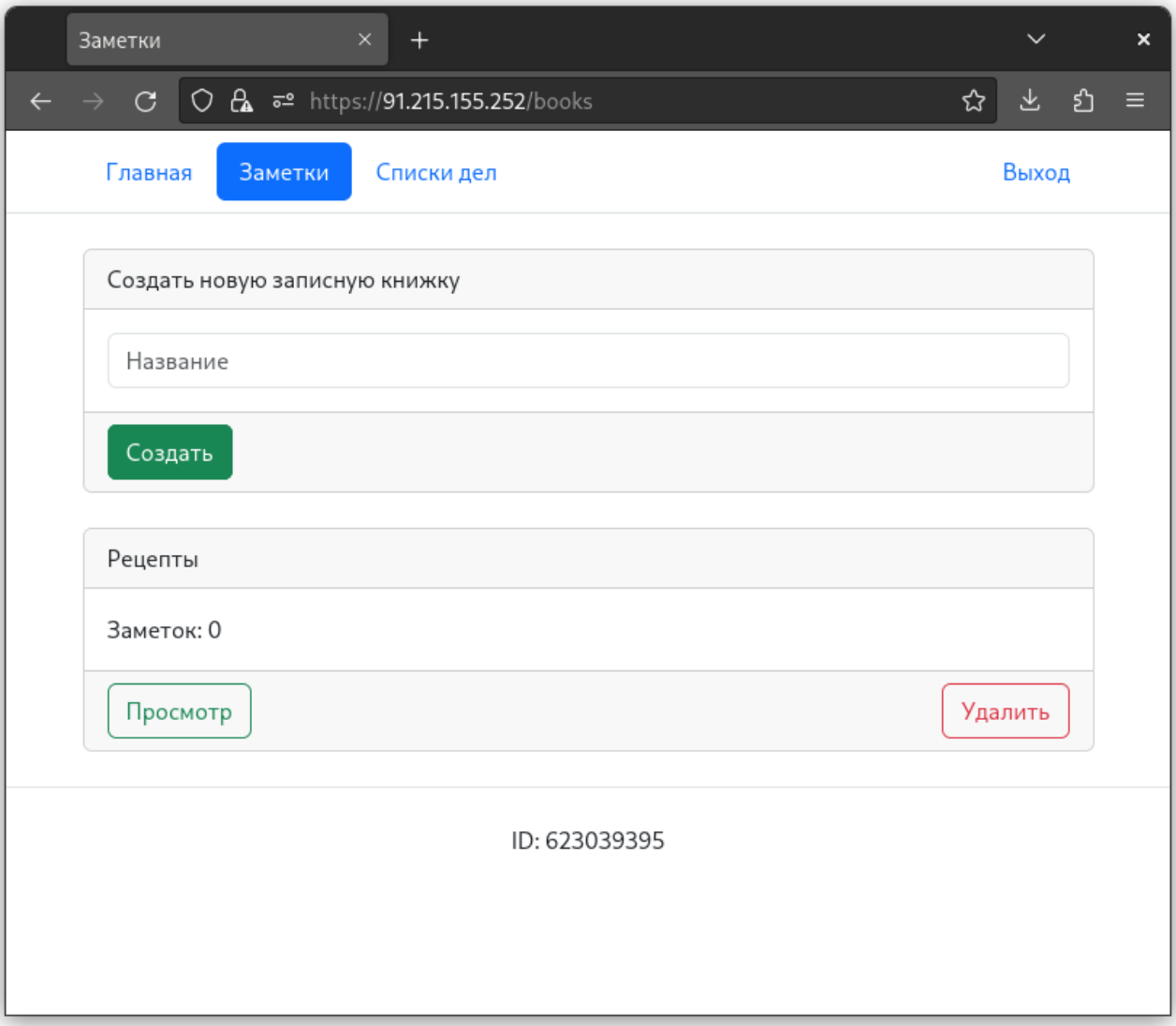

Рисунок 10. Новая записная книжка

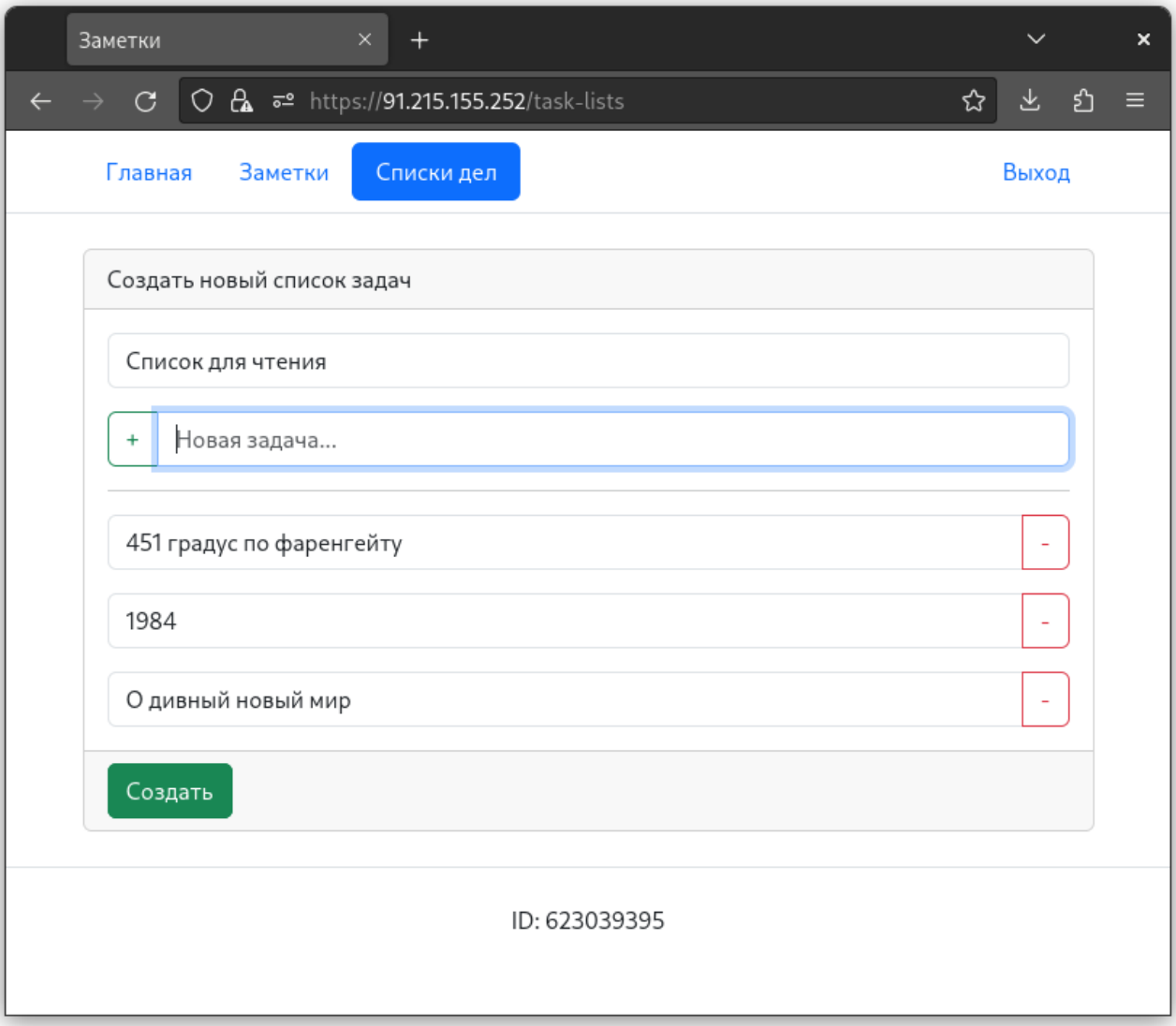

Рисунок 11. Создание списка дел

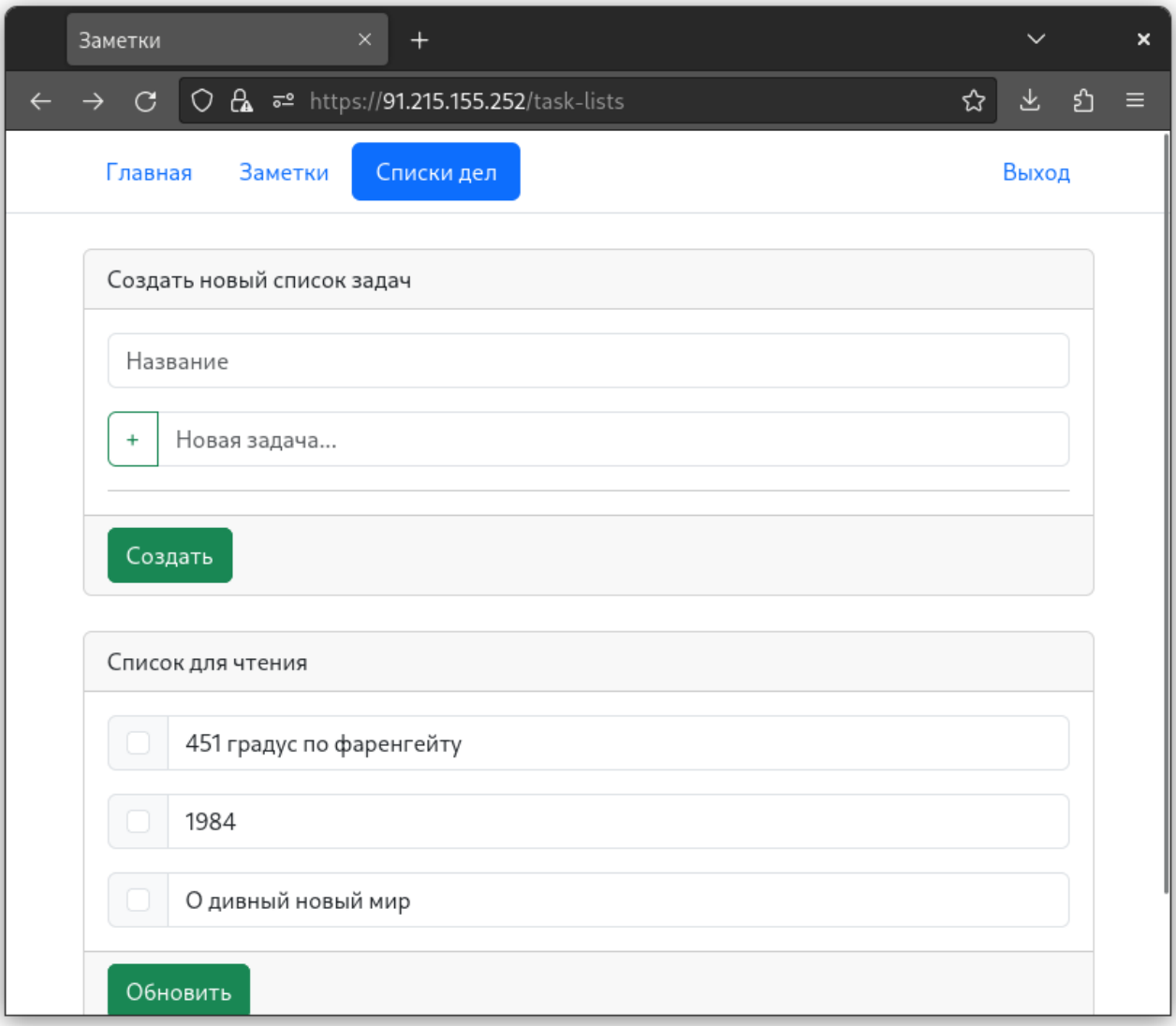

Рисунок 12. Созданный список дел

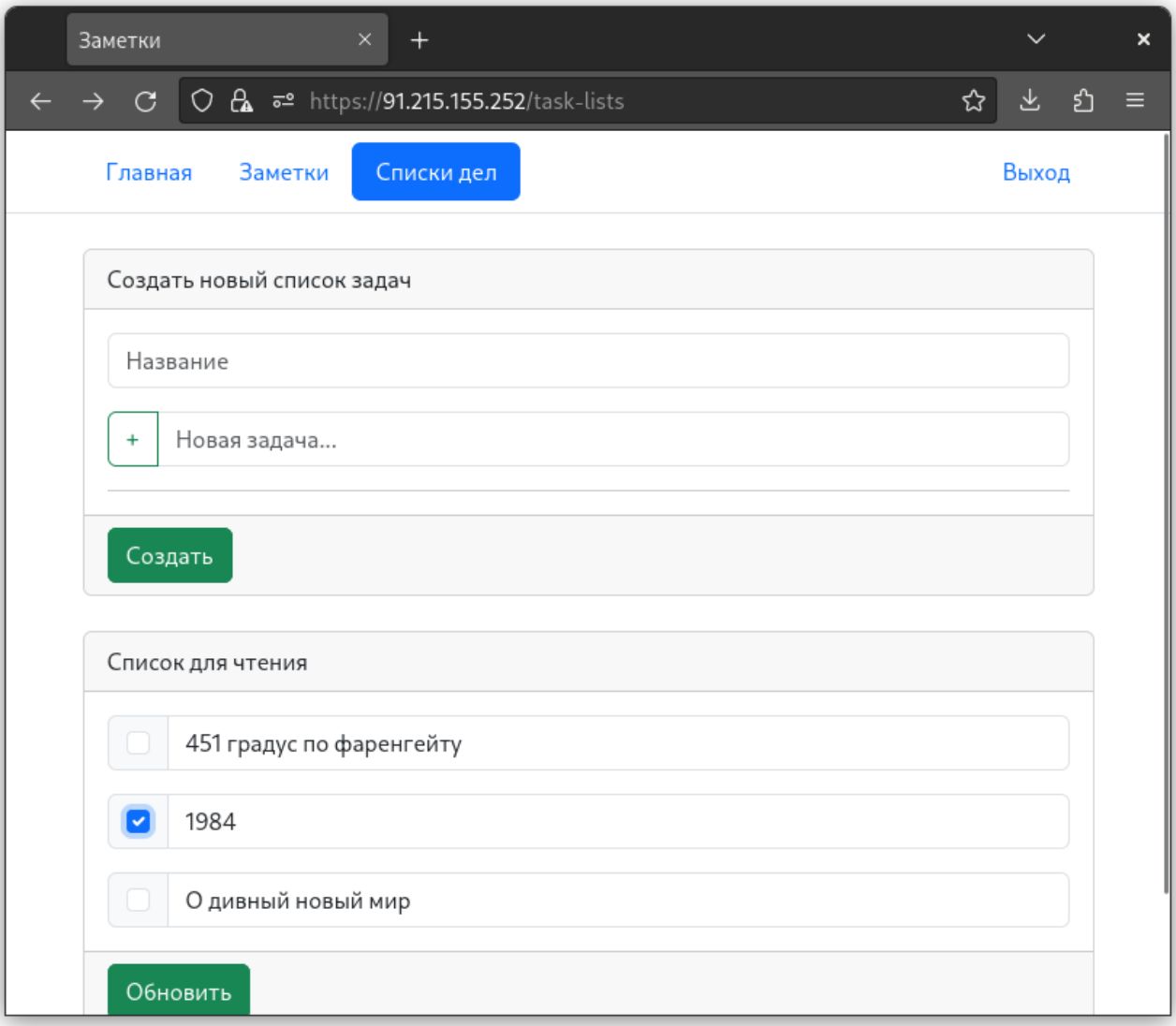

Рисунок 13. Выполненная задача в списке дел

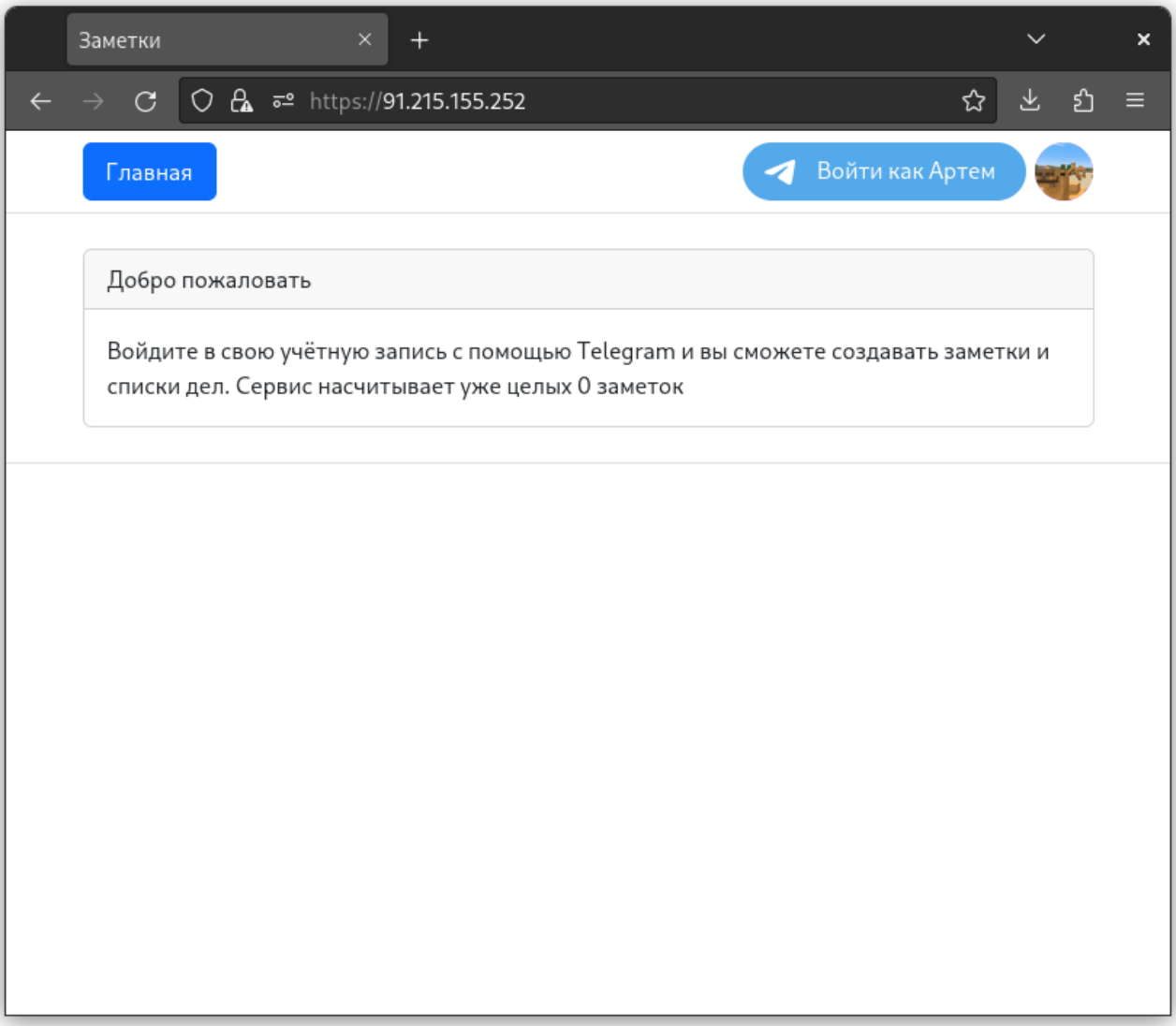

Рисунок 14. Выход из аккаунта