# Техническая документация для проекта "Astra v2.0"

*Разработано студентом 5-го курса:*

*Свистун Н.О, ВКБ-52, ДГТУ*

*г.Ростов-на-Дону*

Оглавление

[Техническая документация для проекта "Astra v2.0" 1](#_Toc167832394)

[Введение 3](#_Toc167832395)

[Цели проекта 4](#_Toc167832396)

[Задачи проекта 6](#_Toc167832397)

[Архитектура проекта 6](#_Toc167832398)

[Используемые технологии 8](#_Toc167832399)

[Установка и настройка 10](#_Toc167832400)

[Заключение 11](#_Toc167832401)

## Введение

Проект "Astra v2.0" разработан с целью создания единой централизованной базы данных, которая включает в себя информацию о кейсах, заданиях и тестах, связанных с продуктами группы компаний "Астра". Центральной идеей проекта является предоставление пользователям удобного и эффективного инструмента для хранения, поиска и использования данных, что существенно улучшает рабочие процессы и повышает общую эффективность компании.

*Проблемы, решаемые проектом*

В процессе работы с продуктами группы компаний "Астра" возникает необходимость управления большим количеством данных, включая кейсы, задания и тесты. Без единой централизованной системы эти данные могут быть разбросаны по различным источникам, что затрудняет их использование и управление. Проект "Astra v2.0" решает следующие ключевые проблемы:

1. Разрозненность данных: отсутствие единой системы для хранения и управления информацией о кейсах, заданиях и тестах.
2. Сложность поиска и фильтрации информации: затрудненный доступ к нужной информации из-за отсутствия эффективных инструментов поиска и фильтрации.
3. Низкая производительность и эффективность работы: увеличение временных затрат на поиск и обработку данных, что снижает общую производительность сотрудников.
4. Отсутствие интеграции и автоматизации процессов: необходимость ручного управления данными без возможности автоматизации рутинных задач.

## Цели проекта

Основная цель проекта заключается в обеспечении надежного хранения, удобного управления и эффективного использования информации. Для достижения этой цели проект фокусируется на нескольких ключевых аспектах.

1. Централизованное хранение данных

Централизованное хранение данных предполагает создание единой базы данных, где будут собраны все кейсы, задания и тесты, связанные с продуктами ГК "Астра". Это обеспечивает несколько преимуществ:

* Надежное и безопасное хранение информации: Все данные хранятся в одном месте, что позволяет применять высокие стандарты безопасности и обеспечивать их целостность.
* Удобство доступа: Пользователи имеют доступ к информации из любого места и в любое время, что облегчает работу и ускоряет процессы.
* Упрощенное управление данными: Администраторам и пользователям проще управлять данными, обновлять их и поддерживать в актуальном состоянии.

1. Интуитивно понятный интерфейс

Разработка удобного пользовательского интерфейса для управления данными играет важную роль в проекте. Основные задачи, решаемые интерфейсом:

* Упрощение выполнения основных задач: Пользователи могут легко добавлять, редактировать, удалять и искать данные благодаря интуитивно понятным формам и элементам управления.
* Снижение обучающих затрат: Понятный интерфейс снижает время на обучение пользователей работе с системой, что повышает общую производительность.
* Эстетичность и эргономичность: Приятный визуальный дизайн и логичная структура элементов управления делают работу с системой более комфортной.

1. Поддержка автоматизированного и ручного добавления данных

Возможность как автоматизированного, так и ручного добавления данных обеспечивает гибкость и точность в управлении информацией:

* Автоматизация процессов: Внедрение инструментов для автоматического сбора и добавления данных позволяет значительно сократить временные затраты и минимизировать ошибки.
* Ручное управление: Администраторы и пользователи могут вручную добавлять и корректировать информацию, что позволяет учесть все нюансы и детали, которые могут быть упущены при автоматизации.
* Обновление данных: Легкость обновления информации способствует поддержанию базы данных в актуальном состоянии, что важно для точности и релевантности данных.

1. Эффективный поиск и фильтрация

Реализация системы тегов и инструментов поиска играет ключевую роль в обеспечении быстрого доступа к нужной информации:

* Система тегов: Позволяет категоризировать данные по различным параметрам, что упрощает их сортировку и поиск.
* Инструменты поиска: Мощные и гибкие инструменты поиска позволяют пользователям быстро находить нужную информацию по ключевым словам, тегам и другим критериям.
* Фильтрация данных: Возможность фильтрации результатов поиска по различным параметрам обеспечивает более точное и быстрое нахождение нужной информации.

1. Обеспечение безопасности и контроля доступа

Поддержка аутентификации и авторизации пользователей необходима для защиты данных и контроля доступа:

* Аутентификация пользователей: Гарантирует, что доступ к системе получают только авторизованные пользователи, что предотвращает несанкционированный доступ.
* Авторизация: Определяет права и уровни доступа пользователей к различным частям системы и данным, что позволяет контролировать, кто и какие действия может выполнять.
* Защита данных: Обеспечивает высокие стандарты безопасности для защиты информации от утечек, взломов и других угроз.

## Задачи проекта

Разработка интуитивно понятного интерфейса для управления вопросами и документами. Простой и удобный интерфейс позволит пользователям легко выполнять свои задачи.

Реализация системы тегов для эффективной фильтрации контента. Теги помогут быстро находить и сортировать информацию по различным категориям.

Обеспечение возможности экспорта данных в различных форматах (CSV, PDF). Экспорт данных позволит использовать информацию вне системы и делиться ею с другими.

Поддержка аутентификации и авторизации пользователей. Защита данных и контроль доступа обеспечат безопасность информации.

Реализация системы уведомлений о важных событиях и изменениях в базе данных. Уведомления помогут пользователям своевременно получать информацию о важных обновлениях.

## Архитектура проекта

Архитектура проекта "Astra v2.0" представляет собой многоуровневую структуру, состоящую из нескольких взаимосвязанных компонентов, обеспечивающих функциональность системы. В проекте используются современные технологии и подходы для обеспечения надежности, масштабируемости и безопасности.

1. Структура проекта

Проект состоит из нескольких приложений, каждое из которых отвечает за определенную функциональность. Основные приложения включают:

* accounts: управление пользователями, аутентификация и авторизация.
* dashboard: управление вопросами и документами, система тегов, экспорт данных.
* home: главный интерфейс, общедоступные страницы.

1. Основные компоненты

***Приложение accounts***

* admin.py: регистрация моделей пользователей и управление административной частью.
* apps.py: конфигурация приложения.
* models.py: модели данных для пользователей, аутентификационные данные.
* views.py: представления для входа, выхода и управления учетными записями.
* urls.py: маршрутизация URL для приложения.
* templates/accounts/: шаблоны для страниц входа и выхода (login.html, logged\_out.html).

***Приложение dashboard***

* admin.py: регистрация моделей вопросов и документов.
* apps.py: конфигурация приложения.
* models.py: модели данных для вопросов, ответов и документов.
* forms.py: формы для создания и редактирования вопросов и документов.
* views.py: представления для управления вопросами, документами и тегами.
* urls.py: маршрутизация URL для приложения.
* templates/dashboard/: шаблоны для различных страниц управления (edit\_question.html, preview\_document.html, questions.html, tags.html, upload\_document.html, view\_document.html, base\_dash.html, create\_question.html, documents.html, edit\_document.html).

***Приложение home***

* admin.py: административные настройки для главной страницы.
* apps.py: конфигурация приложения.
* models.py: модели данных для главной страницы.
* views.py: представления для главной страницы.
* urls.py: маршрутизация URL для главной страницы.
* templates/home/: шаблоны для главной страницы (base.html, index.html).

1. Конфигурационные файлы

* settings.py: основные настройки проекта, включая базы данных, приложения, аутентификацию, статические файлы и другие параметры конфигурации.
* urls.py: глобальная маршрутизация URL для всего проекта.
* wsgi.py: конфигурация для запуска приложения с помощью WSGI сервера.
* asgi.py: конфигурация для запуска приложения с помощью ASGI сервера.

1. Статические файлы и шаблоны

***Статические файлы***

* static/css/: стили для различных страниц приложения (dashboard.css, styles.css).
* static/img/: изображения для использования на страницах (asta\_logo.png, login.png).

***Шаблоны***

Шаблоны используются для создания HTML страниц на основе данных из представлений:

* templates/accounts/: шаблоны для страниц входа и выхода.
* templates/dashboard/: шаблоны для страниц управления вопросами и документами.
* templates/home/: шаблоны для главной страницы.

1. База данных

Проект использует базу данных SQLite (db.sqlite3) для хранения информации о пользователях, вопросах, ответах и документах. Модели данных определяются в соответствующих файлах models.py каждого приложения.

1. Мидлвары и контекстные процессоры

Middleware: настройки мидлваров в settings.py, обеспечивающие обработку запросов, безопасность, управление сессиями и другие функции.

Контекстные процессоры: настройки контекстных процессоров, которые добавляют переменные в контекст шаблонов.

*Таким образом, архитектура проекта "Astra v2.0" организована таким образом, чтобы обеспечить модульность, масштабируемость и удобство управления. Каждый компонент и приложение выполняет свою роль, обеспечивая общую функциональность системы. Эта архитектура позволяет легко добавлять новые функции, масштабировать проект и поддерживать его в актуальном состоянии.*

## Используемые технологии

1. Django: серверная часть и администрирование

Django - это высокоуровневый веб-фреймворк на языке Python, который облегчает создание безопасных и поддерживаемых веб-приложений. Он включает в себя множество встроенных функций и компонентов, которые позволяют быстро и эффективно разрабатывать серверную часть приложений.

* Модели данных: Django ORM (Object-Relational Mapping) позволяет легко определять модели данных и взаимодействовать с базой данных, используя Python-код вместо SQL-запросов.
* Административная панель: Django предоставляет мощную административную панель, которая автоматически генерируется на основе моделей данных. Это позволяет администраторам управлять пользователями, вопросами, документами и другими данными через удобный веб-интерфейс.
* Безопасность: Django включает в себя встроенные механизмы безопасности, такие как защита от CSRF (Cross-Site Request Forgery), защита от SQL-инъекций, управление паролями и другие.
* Маршрутизация: Django использует систему маршрутизации URL, что позволяет легко определять маршруты для различных частей приложения.
* Шаблоны: Django шаблонизатор позволяет создавать динамические HTML-страницы, используя шаблоны и данные из представлений.
* Мидлвары: Django мидлвары обеспечивают обработку запросов и ответов, управление сессиями, обработку ошибок и другие функции на уровне промежуточного слоя между сервером и приложением.

1. HTML, CSS, JavaScript: фронтенд и пользовательский интерфейс

HTML (HyperText Markup Language), CSS (Cascading Style Sheets) и JavaScript являются основными технологиями для разработки фронтенда веб-приложений. Они обеспечивают создание и управление пользовательским интерфейсом.

* HTML: используется для создания структуры веб-страниц. С помощью HTML определяются элементы страницы, такие как заголовки, абзацы, списки, формы и другие компоненты.
  + Примеры файлов: шаблоны страниц (login.html, base.html, questions.html и другие).
* CSS: используется для стилизации веб-страниц, управления их внешним видом и компоновкой. С помощью CSS задаются цвета, шрифты, отступы, расположение элементов и другие стили.
  + Примеры файлов: стили для различных страниц приложения (dashboard.css, styles.css).
* JavaScript: используется для добавления интерактивности и динамического поведения на веб-страницах. С помощью JavaScript можно реализовывать проверку форм на стороне клиента, обработку событий, работу с AJAX-запросами и другие функции.
  + Примеры использования: реализация интерактивных элементов управления, таких как выпадающие меню, модальные окна, фильтрация и сортировка данных на клиентской стороне.

1. SQLite: база данных для хранения информации

SQLite - это легковесная встроенная база данных, которая используется для хранения данных в локальных файлах. Она идеально подходит для небольших и средних приложений, где не требуется сложная серверная инфраструктура.

* Хранение данных: SQLite используется для хранения информации о пользователях, вопросах, ответах, документах и других данных приложения.
  + Пример файла: db.sqlite3 - основной файл базы данных проекта.
* Простота настройки и использования: SQLite не требует отдельной установки или настройки сервера базы данных. Все данные хранятся в одном файле, что упрощает процесс развертывания и администрирования базы данных.
* Поддержка Django: Django имеет встроенную поддержку SQLite, что позволяет легко интегрировать и использовать эту базу данных в проектах на Django. Модели данных автоматически создают соответствующие таблицы в базе данных.
* Производительность: Несмотря на легковесность, SQLite обеспечивает высокую производительность и надежность для большинства задач, связанных с хранением и управлением данными.

## Установка и настройка

***Установка зависимостей***

Создайте и активируйте виртуальное окружение (опционально):

python -m venv venv

source venv/bin/activate # для Linux/macOS

venv\Scripts\activate # для Windows

Установите зависимости из файла requirements.txt:

pip install -r requirements.txt

***Настройка базы данных***

Примените миграции для настройки базы данных:

python manage.py migrate

***Создание суперпользователя***

Создайте суперпользователя для доступа к админке Django:

python manage.py createsuperuser

***Запуск проекта***

Для запуска сервера разработки используйте команду:

python manage.py runserver

Откройте браузер и перейдите по адресу http://127.0.0.1:8000/, чтобы увидеть работающий проект.

## Заключение

Проект "Astra v2.0" разработан для решения ряда задач, связанных с управлением данными о кейсах, заданиях и тестах, связанных с продуктами группы компаний "Астра".

Основной целью проекта было создание единой, централизованной и систематизированной базы данных, которая позволила бы значительно упростить процессы работы с информацией, улучшить ее качество и обеспечить удобство использования.

***Основные достижения проекта***

1. Централизованное хранение данных:

* Проект "Astra v2.0" обеспечивает надежное хранение всех данных в единой базе данных, что исключает разрозненность информации и облегчает ее управление.
* Централизованное хранение позволяет легко находить и использовать нужную информацию, сокращая время на поиск и обработку данных.

1. Интуитивно понятный интерфейс:

* Созданный интерфейс управления данными позволяет пользователям легко добавлять, редактировать и удалять вопросы и документы.
* Визуально привлекательный и логично организованный интерфейс способствует снижению времени на обучение пользователей и повышает их продуктивность.

1. Поддержка автоматизированного и ручного добавления данных:

* Возможность автоматизации процессов добавления и обновления данных минимизирует количество ошибок и ускоряет работу с системой.
* Ручное добавление данных позволяет учитывать все нюансы и детали, обеспечивая высокую точность информации.

1. Эффективный поиск и фильтрация:

* Реализована система тегов, которая позволяет быстро и легко находить нужную информацию по различным параметрам.
* Инструменты поиска и фильтрации обеспечивают быстрый доступ к данным, что значительно повышает эффективность работы пользователей.

1. Обеспечение безопасности и контроля доступа:

* Внедрена система аутентификации и авторизации, которая гарантирует, что доступ к системе получают только авторизованные пользователи.
* Контроль доступа позволяет задавать различные уровни прав для пользователей, обеспечивая безопасность и защиту данных.

***Влияние на эффективность деятельности***

1. Повышение производительности:

* Удобный и интуитивно понятный интерфейс, эффективные инструменты поиска и фильтрации данных, а также автоматизация рутинных задач способствуют значительному повышению производительности сотрудников.
* Снижение времени на поиск и обработку информации позволяет сотрудникам сосредоточиться на более важных задачах, что повышает общую эффективность деятельности компании.

1. Улучшение качества данных:

* Централизованное хранение и автоматизация процессов добавления и обновления данных обеспечивают высокое качество и точность информации.
* Регулярное обновление данных и возможность ручного управления позволяют поддерживать базу данных в актуальном состоянии, что является критически важным для принятия обоснованных решений.

1. Обеспечение безопасности и конфиденциальности данных:

* Система аутентификации и авторизации, а также контроль доступа обеспечивают высокую степень защиты данных от несанкционированного доступа и утечек.
* Ведение журналов действий пользователей позволяет отслеживать изменения и оперативно реагировать на потенциальные угрозы безопасности.

1. Удобство использования и взаимодействия с данными:

* Интуитивно понятный интерфейс и инструменты управления данными делают работу с системой удобной и эффективной.
* Экспорт данных в различных форматах (CSV, PDF) позволяет легко делиться информацией с коллегами и использовать ее вне системы, что упрощает взаимодействие и сотрудничество.

***Перспективы развития проекта***

1. Интеграция с другими системами:

* Возможность интеграции с другими корпоративными системами и инструментами для обеспечения более комплексного управления данными.
* Расширение функциональности системы за счет добавления новых модулей и интеграции с внешними сервисами.

1. Расширение функциональности:

* Внедрение новых функций и возможностей, таких как расширенные инструменты аналитики, отчетности и мониторинга.
* Улучшение пользовательского интерфейса и добавление новых инструментов для управления данными.

1. Улучшение UX/UI:

* Постоянное совершенствование дизайна и интерфейса системы на основе отзывов пользователей и анализа их потребностей.
* Обеспечение адаптивности интерфейса для удобного использования на различных устройствах, включая мобильные и планшеты.

Основное внимание в проекте уделено качеству данных, удобству их добавления и использования, а также повышению эффективности деятельности компании. В результате реализации проекта сотрудники компании получают мощный инструмент для управления информацией, который способствует улучшению процессов работы, повышению производительности и обеспечению безопасности данных.