

Дипломный проект защищен

С оценкой \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 2024 года

Рук учебной частью

\_\_\_\_\_

(подпись)

## ДИПЛОМНЫЙ ПРОЕКТ

*ДП «ТОП» Python-33.СТ.11/2024.*

***Тема: Разработка веб-приложение «Agile-планирование»***

*Разработал: Стеклова Дарья Дмитриевна*

Студент \_\_\_\_\_

(подпись)

(дата)

(Ф,И,О. на русском языке)

Преподаватель \_\_\_\_\_

(подпись)

(дата)

(Ф,И,О. на русском языке)

***Экзаменационная комиссия по защите дипломных проектов:***

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(дата)

\_\_\_\_\_

(Ф.И.О)

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(дата)

\_\_\_\_\_

(Ф.И.О)

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(дата)

\_\_\_\_\_

(Ф.И.О)

\_\_\_\_\_

(подпись)

\_\_\_\_\_

(дата)

\_\_\_\_\_

(Ф.И.О)

***Екатеринбург-2024***

# **ЗАДАНИЕ И КАЛЕНДАРНЫЙ ПЛАН**

*по выполнению дипломной работы*

*студентом группы*

---

*(Фамилия, Имя, Отчество)*

*Тема работы:*

*Утверждена приказом директора филиала от \_\_\_\_\_ № \_\_\_\_\_*

*Срок сдачи студентом законченной работы: \_\_\_\_\_*

*Объем дипломной работы: 50-80 машинописных страниц.*

*Перечень прилагаемого материала \_\_\_\_\_*

---

---

---

*Организации, на базе которых выполняется дипломная работа:*

*а)*

---

*б)*

---

*в)*

---

## ***Календарный план выполнения дипломной работы***

<i>Наименование этапа работ</i>	<i>Срок выполнения</i>		<i>Оценка выполненного этапа работы</i>	<i>Роспись руководителя</i>
	<i>по плану</i>	<i>фактически</i>		
<i>Написание введения</i>				
<i>Написание главы 1</i>				
<i>Написание главы 2</i>				
<i>Написание главы 3</i>				
<i>Написание заключения</i>				
<i>Оформление дипломной работы</i>				
<i>Получение отзыва руководителя</i>				
<i>Подготовка доклада и наглядных пособий</i>				
<i>Защита дипломной работы</i>				

**УТВЕРЖДАЮ**

"\_\_" \_\_\_\_\_ 2024 г.

Дипломный руководитель \_\_\_\_\_

Задание принял к исполнению:

Руководитель \_\_\_\_\_

\_\_\_\_\_  
(Подпись студента)

## *СОДЕРЖАНИЕ*

Перечень условных обозначений.....	6
Глава 1. Теоретическая часть.....	13
1.1 Описание объекта разработки .....	13
1.2 Общая задача .....	14
1.3 История вопроса .....	15
1.4 Современное состояние проблемы.....	16
1.5 Обоснование выбора технологий.....	17
Глава 2. Практическая часть.....	19
2.1 Обоснование проектирования .....	19
2.2 Анализ аналогов.....	20
2.3 Используемые инструменты разработки.....	21
2.3.1 Python.....	21
2.3.2 Django.....	24
2.3.3 Bootstrap .....	27
2.4 Описание базы данных .....	30
2.5 Этапы реализации.....	32
2.5.1 Проектирование интерфейса .....	32
2.5.2 Разработка функционала.....	38
2.5.3 Тестирование .....	40
2.6 Экономические расчёты.....	42
Глава 3. Обобщение результатов .....	43
3.1 Предложенное решение .....	43
3.2 Последствия внедрения .....	44
3.3 Рекомендации по использованию .....	45
Заключение.....	46
Список литературы.....	50

Лист замечаний .....	52
----------------------	----

## *Перечень условных обозначений*

- 1. HTML (HyperText Markup Language)** – язык разметки гипертекста, используемый для создания структуры веб-страниц, включая заголовки, абзацы, ссылки и другие элементы.  
Пример: определяет, где на странице будет размещён текст, изображение или видео.
- 2. CSS (Cascading Style Sheets)** – каскадные таблицы стилей, применяемые для описания внешнего вида элементов HTML, таких как шрифты, цвета, размеры и расположение.  
Пример: отвечает за оформление кнопок, фонов и текстов.
- 3. JavaScript** – язык программирования, который позволяет добавлять интерактивность и динамику на веб-страницы.  
Пример: используется для создания выпадающих меню, слайдеров и проверки форм.
- 4. Django** – веб-фреймворк на языке Python, предназначенный для быстрого создания серверных приложений. Django предоставляет готовые компоненты для работы с базой данных, обработки запросов и управления аутентификацией.
- 5. Bootstrap** – популярный фронтенд-фреймворк, который содержит готовые CSS и JavaScript компоненты, упрощающие создание адаптивного дизайна.  
Пример: используется для построения кнопок, форм, навигационных панелей.
- 6. SQLite** – легковесная реляционная база данных, встроенная в приложение. Веб-приложения используют её для хранения данных, таких как пользовательские записи или цели.

- 7. API (Application Programming Interface)** – интерфейс программирования приложений, который позволяет приложениям взаимодействовать друг с другом.
- 8. CRUD (Create, Read, Update, Delete)** – набор операций для управления данными в приложениях.  
Пример: создаёт, читает, обновляет и удаляет записи в базе данных.
- 9. UX/UI (User Experience/User Interface)** – термины, описывающие дизайн интерфейсов и пользовательский опыт.
  - **UI** отвечает за внешний вид приложения (кнопки, формы, шрифты).
  - **UX** фокусируется на удобстве взаимодействия пользователя с приложением.
- 10. HTTP (HyperText Transfer Protocol)** – протокол передачи гипертекста, который используется для обмена данными между клиентом (браузером) и сервером.  
Пример: когда пользователь отправляет форму или загружает веб-страницу.
- 11. JSON (JavaScript Object Notation)** – текстовый формат обмена данными, который используется для передачи данных между сервером и клиентом.
- 12. Agile** – методология разработки программного обеспечения, основанная на итеративном подходе и гибком управлении изменениями.
- 13. ORM (Object-Relational Mapping)** – технология для работы с базами данных через объектно-ориентированные языки программирования.  
Пример: в Django используется встроенный ORM для управления моделями данных.
- 14. URL (Uniform Resource Locator)** – унифицированный указатель ресурса, который определяет адрес веб-страницы или API.

**15. MVC (Model-View-Controller)** – архитектурная схема, разделяющая приложение на три части:

- **Model** – отвечает за работу с данными.
- **View** – отвечает за отображение данных.
- **Controller** – управляет логикой приложения и взаимодействием между Model и View.

**16. HTTPS (HyperText Transfer Protocol Secure)** – безопасная версия протокола HTTP, которая шифрует данные при их передаче между клиентом и сервером.

**17. UI Kit** – набор готовых интерфейсных компонентов для создания дизайна приложений.

Пример: библиотека Bootstrap также является UI Kit.

**18. XSS (Cross-Site Scripting)** - Межсайтовый скриптинг — это тип уязвимости веб-приложений, который - позволяет злоумышленникам внедрять вредоносные скрипты на страницы сайта, отображаемые другим пользователям.

**19. CSRF (Cross-Site Request Forgery)** - Межсайтовые подделки запросов.

Это уязвимость, которая позволяет злоумышленникам заставить пользователя выполнять непреднамеренные действия на доверенном сайте, где он уже авторизован.



## *Введение*

Развитие современных технологий оказывает значительное влияние на различные сферы человеческой деятельности, включая личное планирование и управление временем. Дипломный проект "**Agile-планирование**" представляет собой веб-приложение, созданное для упрощения процесса постановки целей, анализа ежедневных достижений и ведения личных заметок. В основе разработки лежат передовые инструменты и подходы, такие как фреймворк Django для серверной части и библиотека Bootstrap для создания адаптивного и удобного пользовательского интерфейса.

**Целью данного проекта** является создание удобного и функционального инструмента для управления личными задачами, анализа повседневной продуктивности и повышения осознанности. Приложение ориентировано на широкую аудиторию, включая студентов, специалистов, а также всех, кто стремится повысить эффективность своей деятельности и структурировать ежедневные задачи.

### *Оценка современного состояния вопроса.*

В условиях интенсивного информационного потока проблема организации личного времени и контроля задач становится всё более актуальной. Современные пользователи часто сталкиваются с трудностями, связанными с планированием своих дел, постановкой достижимых целей и регулярной рефлексией. Это особенно важно в профессиональной среде, где требуется балансировать между рабочими и личными обязанностями.

Существующие популярные решения, такие как Trello, Notion, Google Keep, и другие, хотя и предлагают широкий функционал, нередко оказываются избыточными и сложными для индивидуального использования. Эти инструменты чаще ориентированы на командную работу или управление большими проектами, что снижает их удобство в повседневной жизни. Кроме

того, отсутствие персонализированных настроек и избыточная сложность интерфейса создают барьеры для их эффективного использования обычными пользователями.

На этом фоне становится очевидной необходимость разработки простого и интуитивного инструмента, который мог бы сочетать в себе гибкость в постановке задач и анализе результатов, но при этом оставался доступным и понятным для всех категорий пользователей.

#### *Тенденции решения поставленных задач.*

В последние годы популярность методологий Agile, ранее применяемых исключительно в управлении проектами, заметно возросла в контексте личной эффективности. Принципы Agile, такие как итеративное планирование, гибкость и регулярный анализ результатов, всё чаще адаптируются для решения индивидуальных задач. Это связано с тем, что такой подход позволяет пользователям сосредоточиться на самых важных приоритетах, вовремя корректировать свои планы и оценивать прогресс.

Сочетание аналитики и гибкости в планировании открывает новые возможности для разработки инструментов самоорганизации. В частности, внедрение функционала для регулярной рефлексии (например, анализ достижений и благодарностей за день) помогает пользователям улучшать свою продуктивность, выявлять ключевые успехи и формировать положительный настрой на будущее.

#### *Актуальность работы и используемые технологии.*

Разработка веб-приложения "**Agile-дневник**" актуальна для всех, кто стремится эффективно управлять своими целями и задачами. В основе системы лежит идея совмещения инструментов планирования и рефлексии в одном интуитивном интерфейсе. Это делает приложение универсальным и

доступным для широкой аудитории, включая студентов, офисных работников и фрилансеров.

Для реализации проекта использованы современные технологии:

- **Django** – мощный и гибкий фреймворк для создания серверной части приложения;
- **Bootstrap** – библиотека для упрощённой разработки адаптивного дизайна;
- **HTML и CSS** – стандартные инструменты веб-разработки для построения клиентской части.

Выбор этих технологий обусловлен их стабильностью, обширным сообществом и широкими возможностями для настройки, что позволяет достичь высокой производительности и удобства приложения.

#### *Ценность достигнутых результатов и отрасль применения.*

Разработанное веб-приложение обладает высокой практической ценностью. Его основные преимущества заключаются в простоте использования, гибкости функционала и возможности адаптации под индивидуальные потребности пользователей. Оно может быть полезно в различных сферах, таких как:

- **Образование** – для планирования учебного процесса и анализа достигнутых результатов;
- **Бизнес** – для постановки и контроля личных и профессиональных задач;
- **Личная жизнь** – для ведения ежедневных записей, постановки целей и анализа их выполнения.

Кроме того, приложение способствует формированию позитивного настроя благодаря функционалу благодарностей и рефлексии, что делает его особенно ценным инструментом для повышения личной эффективности и продуктивности.

*Рекомендации по использованию.*

Разработанное приложение рекомендуется для использования как инструмент самоорганизации, который помогает пользователям структурировать свои задачи, отслеживать прогресс и вести осознанный образ жизни. В перспективе приложение может быть расширено дополнительными функциями, такими как интеграция с календарями, автоматическое составление отчётов или разработка мобильной версии.

Таким образом, проект "Agile-дневник" отвечает актуальным потребностям пользователей, сочетая в себе простоту и функциональность, а также демонстрирует возможности современных технологий для решения задач самоорганизации.

## *Глава 1. Теоретическая часть*

### *1.1 Описание объекта разработки*

Веб-приложение "Agile-дневник" представляет собой современный инструмент для управления целями, анализа достижений и ведения личных заметок. В условиях ускоряющегося темпа жизни многие люди сталкиваются с необходимостью структурировать свои дела, фиксировать важные достижения и выделять время на самоанализ. "Agile-дневник" отвечает этим требованиям, предоставляя пользователю возможность управлять повседневными задачами и рефлексировать над их выполнением.

Основные функциональные возможности приложения включают:

- **Регистрацию и авторизацию пользователей.** Каждому пользователю предоставляется личный доступ к системе, где хранятся его данные.
- **Ведение ежедневных записей ("Мой день").** Эти записи включают три ключевых элемента:
  - **Благодарности** – пользователю предлагается отметить три вещи, за которые он благодарен уходящему дню.
  - **Достижения** – пользователь фиксирует три результата, которых он достиг за день.
  - **Мысли** – свободный текст для выражения идей, размышлений или наблюдений.
- **Управление целями.** Пользователь может добавлять свои цели, отслеживать их выполнение, а также архивировать завершённые задачи.
- **Просмотр архивных записей.** Все выполненные цели и старые записи сохраняются в архиве, что позволяет анализировать результаты за длительный период.

- **Редактирование данных.** Приложение даёт возможность корректировать введённую информацию.

Удобство и простота использования обеспечиваются адаптивным интерфейсом, что делает приложение доступным для работы с любых устройств – настольных компьютеров, планшетов и смартфонов.

"Agile-дневник" ориентирован на индивидуальное использование, что делает его полезным для студентов, специалистов и всех, кто стремится повысить продуктивность и осознанность.

## *1.2 Общая задача*

Целью разработки веб-приложения "Agile-дневник" является создание системы, которая совмещает возможности планирования, анализа и рефлексии, сохраняя при этом простоту и интуитивность интерфейса.

Общая задача проекта включает реализацию следующих аспектов:

### **1. Создание функционала персонализации.**

Пользователь должен иметь возможность зарегистрироваться, войти в свою учётную запись и работать с индивидуальными данными. Это требует разработки системы авторизации, которая обеспечивает безопасное хранение пользовательских данных.

### **2. Интеграция модуля для ежедневных записей.**

Одной из ключевых функций "Agile-дневника" является запись благодарностей, достижений и мыслей. Такая структура помогает пользователю акцентировать внимание на позитивных аспектах жизни и результатах своей деятельности.

### **3. Планирование и управление целями.**

В приложении реализуется возможность добавления целей с последующим изменением их статуса (например, "Выполнено"). Это

позволяет пользователям фиксировать как краткосрочные, так и долгосрочные задачи.

#### 4. Формирование архива.

Все завершённые задачи и старые записи сохраняются в архиве, что упрощает анализ выполненной работы.

#### 5. Обеспечение адаптивного дизайна.

Приложение должно быть удобным для использования на различных устройствах, обеспечивая корректное отображение интерфейса на экранах разного размера.

Важным аспектом разработки является интеграция функций, повышающих вовлечённость пользователя, таких как кнопки "Выполнено" и "В архив" для целей.

### *1.3 История вопроса*

Современный рынок веб-приложений предлагает широкий выбор инструментов для управления задачами, заметками и целями. Среди них можно выделить такие популярные решения, как Trello, Notion и Todoist. Однако, несмотря на их востребованность, данные инструменты не всегда удовлетворяют потребности индивидуального пользователя.

#### Trello

Trello основан на методологии Kanban и предоставляет пользователю возможность организовывать задачи в виде досок и карточек. Основные преимущества:

- Интуитивно понятный интерфейс.
- Возможность создавать списки и настраивать карточки задач.
- Интеграция с популярными инструментами, такими как Slack, Google Drive и другие.

Однако, несмотря на популярность, Trello ориентирован прежде всего на командную работу. Индивидуальные пользователи сталкиваются с избыточной сложностью настроек и недостаточной гибкостью для персонализированного использования.

### Notion

Notion является многофункциональной платформой, которая сочетает элементы заметок, баз данных и управления проектами. Его преимущества включают:

- Высокую степень настройки.
- Возможность структурировать данные под любые задачи.
- Расширенные функции интеграции с внешними сервисами.

Тем не менее, Notion также имеет свои недостатки. Для новичков платформа может оказаться слишком сложной, что делает её неудобной для быстрого освоения. Кроме того, избыточный функционал отвлекает от простых задач.

### Todoist

Todoist – это минималистичный инструмент для управления задачами, который выделяется своей простотой. Пользователи могут:

- Быстро добавлять задачи с указанием приоритетов и сроков.
- Интегрировать приложение с календарями и электронной почтой.
- Получать уведомления о предстоящих событиях.

Однако Todoist не предоставляет возможности ведения личных записей или анализа достижений. Это ограничивает его применение для пользователей, которые хотят не только управлять задачами, но и фиксировать результаты.

## *1.4 Современное состояние проблемы*



Проблема эффективного управления временем и задачами остаётся актуальной в современном обществе. С увеличением количества информации и растущими требованиями к продуктивности многие люди испытывают трудности с организацией своих дел.

Одним из перспективных подходов к решению этой проблемы является применение элементов Agile-методологии. Agile, изначально разработанный для управления проектами в сфере IT, нашёл широкое применение в личной и профессиональной жизни.

Принципы Agile включают:

- Постановку реалистичных целей с последующей оценкой их выполнения.
- Постоянную адаптацию планов к текущим обстоятельствам.
- Регулярную рефлексию для выявления проблем и определения достижений.

Эти принципы позволяют людям быть более гибкими и осознанными в управлении своими задачами. Однако существующие инструменты на рынке, такие как Trello и Notion, не адаптированы для простого и интуитивного применения индивидуальными пользователями.

### *1.5 Обоснование выбора технологий*

Для реализации "Agile-дневника" были выбраны следующие технологии:

- **Django.** Этот фреймворк для веб-разработки на Python предоставляет мощные инструменты для создания серверной части приложения. Его преимущества:
  - Наличие встроенных инструментов для работы с базами данных.

- Простота настройки системы авторизации и управления пользователями.
- Быстрая разработка благодаря модульности.
- **Bootstrap.** CSS-библиотека, которая позволяет создавать адаптивный и стильный интерфейс. Bootstrap обеспечивает:
  - Поддержку адаптивного дизайна.
  - Использование готовых компонентов, таких как кнопки, формы и меню.
  - Простоту стилизации интерфейса.
- **HTML, CSS и JavaScript.** Эти технологии были использованы для создания клиентской части приложения, обеспечивая:
  - Интерактивность интерфейса.
  - Корректное отображение страниц на устройствах с разными характеристиками.

Выбор данных технологий обусловлен их популярностью, стабильностью и доступностью документации, что упрощает разработку и поддержку приложения.

## *Глава 2. Практическая часть*

### *2.1 Обоснование проектирования*

#### *Цели и задачи проекта*

Проект направлен на решение трёх ключевых задач:

#### **1. Управление целями.**

Система предоставляет пользователю возможность добавлять цели, задавать их описание, контролировать процесс выполнения и архивировать завершённые задачи. Это упрощает отслеживание прогресса и помогает пользователю сосредотачиваться на приоритетах.

#### **2. Анализ достижений.**

Ведение дневника с ежедневной фиксацией благодарностей и достижений помогает развивать привычку рефлексии, что, согласно исследованиям в области психологии, способствует повышению осознанности и удовлетворённости жизнью.

#### **3. Ведение личных заметок.**

Пользователь может записывать свои мысли, идеи и наблюдения. Это создаёт условия для творческого самовыражения и анализа.

Зачем нужен "Agile-дневник"

В условиях высокой информационной нагрузки и стремительного ритма жизни людям необходимо оптимизировать управление временем. Большинство существующих инструментов ориентированы на командную работу или обладают сложным интерфейсом, что делает их неудобными для индивидуального использования. "Agile-дневник" решает эти проблемы, предлагая простой и интуитивный интерфейс, адаптированный для личных целей.

## 2.2 Анализ аналогов

Сравним ключевые приложения: Notion, Trello, Todoist и Google Keep.

Notion

Преимущества:

- Инструмент объединяет заметки, базы данных, списки задач.
- Высокая гибкость и масштабируемость: можно создавать

структуры любой сложности.

Недостатки:

- Высокий порог входа для начинающих пользователей.
- Отсутствие функции рефлексии, что делает инструмент менее

подходящим для личного анализа.

Trello

Преимущества:

- Простота управления через карточки.
- Возможность создания визуальных досок для планирования.

Недостатки:

• Фокус на командной работе; для индивидуальных пользователей функционал ограничен.

- Нет функций для записи благодарностей или анализа эмоций.

Todoist

Преимущества:

- Удобный интерфейс для работы с задачами.
- Интеграция с календарями и напоминаниями.

Недостатки:

- Отсутствие функций для ведения дневника или рефлексии.
- Лишён гибкости в плане визуального оформления.

Google Keep

Преимущества:

- Простота интерфейса.

- Интеграция с Google Workspace.

Недостатки:

- Ограниченность функционала: подходит только для создания коротких заметок.

- Не предназначен для долгосрочного планирования и анализа. Ни одно из рассмотренных решений не удовлетворяет полностью потребности в управлении целями, анализе достижений и ведении заметок. "Agile-дневник" направлен на устранение этих пробелов.

## *2.3 Используемые инструменты разработки*

### *2.3.1 Python*

Python — это высокоуровневый язык программирования общего назначения, известный своей простотой и читаемостью. Он был создан в 1991 году Гвидо ван Россумом и с тех пор стал одним из самых популярных языков благодаря своей универсальности и богатой экосистеме библиотек. Python широко используется в веб-разработке, анализе данных, искусственном интеллекте, машинном обучении, автоматизации и других областях.

Ключевые особенности Python:

#### **1. Читаемость кода:**

Синтаксис Python интуитивно понятен, что снижает порог входа для новичков и позволяет разработчикам легко работать с кодом друг друга.

#### **2. Кроссплатформенность:**

Python работает на большинстве операционных систем, включая Windows, macOS и Linux, что делает его универсальным для разработки.

### 3. Интерпретируемый язык:

Python не требует компиляции, что ускоряет процесс разработки и упрощает тестирование и отладку.

### 4. Многообразие библиотек и фреймворков:

Python имеет огромное количество библиотек, таких как NumPy и pandas для анализа данных, TensorFlow и PyTorch для машинного обучения, Django и Flask для веб-разработки.

### 5. Активное сообщество:

Широкая база пользователей обеспечивает большое количество документации, учебных материалов и поддержку.

Преимущества Python:

#### 1. Простота и удобство:

- Легкость изучения делает Python идеальным для новичков.
- Простые конструкции кода позволяют сосредоточиться на логике решения задач, а не на синтаксисе.

#### 2. Гибкость:

- Python поддерживает несколько парадигм программирования, включая объектно-ориентированное, функциональное и процедурное программирование.
- Подходит для задач любой сложности: от скриптов для автоматизации до сложных систем искусственного интеллекта.

#### 3. Обширная стандартная библиотека:

- Стандартная библиотека Python предоставляет широкий спектр модулей и функций для работы с текстом, числами, временем, сетью, базами данных и многим другим.
- Это позволяет выполнять многие задачи без установки дополнительных библиотек.

#### 4. Быстрое прототипирование:

- Простота кода и наличие готовых решений сокращают время разработки прототипов.
5. Популярность в индустрии:
- Python широко используется в ведущих компаниях, таких как Google, Facebook, Amazon и Netflix.
  - Большое количество вакансий для разработчиков Python делает его привлекательным для изучения.
6. Широкий спектр применения:
- Веб-разработка: Django, Flask.
  - Наука о данных: pandas, NumPy, SciPy.
  - Машинное обучение и ИИ: TensorFlow, PyTorch.
  - Автоматизация: скрипты для обработки данных, тестирования систем.
7. Развитая интеграция:
- Python легко интегрируется с другими языками программирования и системами, такими как C/C++, Java и .NET.
8. Поддержка динамической типизации:
- Python не требует явного указания типов переменных, что упрощает код и делает его более лаконичным.

Недостатки Python (для объективности):

- Более низкая производительность по сравнению с языками, такими как C++ или Java, из-за интерпретируемой природы. Однако это компенсируется применением высокопроизводительных библиотек.
- Ограничения в мобильной разработке: Python менее популярен для создания мобильных приложений.

- Потенциальные проблемы с многопоточностью из-за глобальной блокировки интерпретатора (GIL).

Python является мощным и удобным инструментом, который сочетает в себе простоту, функциональность и богатую экосистему. Эти качества делают его одним из лучших выборов для реализации как небольших скриптов, так и сложных систем.

### 2.3.2 Django

Django — это высокоуровневый фреймворк для веб-разработки на языке Python, который позволяет создавать надежные и масштабируемые веб-приложения. Он был разработан в 2005 году разработчиками газеты *Lawrence Journal-World*, чтобы облегчить создание новостных сайтов. Django основывается на архитектуре *Model-View-Template (MVT)*, что помогает организовать код структурированно и чисто.

Цель Django — ускорить процесс разработки, предоставляя разработчикам множество инструментов и компонентов "из коробки", таких как системы аутентификации, маршрутизация URL, работа с базами данных и формами. Django активно поддерживается сообществом, имеет обширную документацию и использует принцип "Don't Repeat Yourself" (DRY), минимизируя дублирование кода.

#### Преимущества Django

##### 1. "Из коробки" функционал

- Django предоставляет встроенные компоненты, которые покрывают основные потребности разработки:
  - Система аутентификации пользователей.
  - Административный интерфейс для управления данными.
  - Механизмы защиты от угроз (XSS, CSRF, SQL-инъекции).



- Поддержка ORM (Object-Relational Mapping) для работы с базами данных.

## 2. Принцип DRY (Don't Repeat Yourself)

- Django поддерживает повторное использование кода, что делает процесс разработки более продуктивным и организованным.

## 3. Архитектура MVT (Model-View-Template)

- Разделение логики приложения, управления данными и представления интерфейса упрощает работу с кодом, его поддержку и тестирование.

## 4. Безопасность

- Django обеспечивает высокий уровень защиты, предотвращая распространённые угрозы:
  - SQL-инъекции.
  - Межсайтовый скриптинг (XSS).
  - Межсайтовые подделки запросов (CSRF).
  - Кликджекинг.
- Встроенные механизмы аутентификации и управления правами доступа.

## 5. Масштабируемость

- Django подходит для создания как небольших проектов, так и крупных корпоративных систем с высокой нагрузкой. Его масштабируемость делает его популярным выбором для приложений с миллионами пользователей (например, Instagram).

## 6. Удобство работы с базами данных

- Django использует встроенный ORM (Object-Relational Mapper), который позволяет работать с базами данных на уровне Python-кода, не требуя написания SQL-запросов.
- Поддерживает множество баз данных, включая PostgreSQL, MySQL, SQLite и Oracle.

## 7. Автоматизация задач

- Встроенная система миграций помогает автоматически управлять изменениями в структуре базы данных.
- Административная панель позволяет управлять данными приложения без необходимости разработки пользовательского интерфейса.

#### 8. Обширное сообщество и документация

- Django имеет огромное сообщество разработчиков, которые создают плагины, делятся опытом и помогают решать проблемы.
- Качественная документация делает Django доступным для изучения даже новичками.

#### 9. Поддержка современных стандартов

- Django обеспечивает поддержку REST API через дополнительные библиотеки, такие как Django REST Framework.
- Поддерживает современные подходы к разработке: асинхронное программирование, интеграцию с фронтенд-фреймворками (React, Vue.js).

#### 10. Кроссплатформенность

- Django работает на всех основных операционных системах, таких как Windows, macOS и Linux.

#### 11. Простота масштабирования

- Поддержка различных серверных конфигураций и возможность легко интегрировать кеширование, балансировщики нагрузки и серверы для хранения медиафайлов.

#### Недостатки Django (для объективности)

##### 1. Сложность для небольших приложений:

- Для мелких проектов Django может быть избыточным, так как содержит множество функций, которые не всегда нужны.

##### 2. Зависимость от ORM:

- Встроенный ORM может быть менее гибким для сложных SQL-запросов.

### 3. Скорость работы:

- Django уступает по скорости "легковесным" фреймворкам, таким как Flask.

#### Примеры использования Django

Django активно применяется для создания сложных веб-приложений, таких как:

- Социальные сети (Instagram).
- Новостные сайты.
- Электронная коммерция (Shopify).
- Личные блоги и портфолио.

Django — это мощный инструмент для разработки веб-приложений, который сочетает богатый функционал, безопасность и производительность. Его преимущества делают его отличным выбором как для начинающих, так и для профессиональных разработчиков, обеспечивая высокую скорость и удобство разработки.

#### 2.3.3 *Bootstrap*

Bootstrap — это популярный фреймворк для разработки интерфейсов веб-приложений, созданный разработчиками из компании Twitter. Он представляет собой набор инструментов для создания адаптивных, удобных и стильных веб-интерфейсов, включающий CSS-стили, JavaScript-библиотеки и готовые компоненты интерфейса. Bootstrap помогает ускорить процесс разработки и обеспечивает единообразие дизайна на всех устройствах.

Последняя стабильная версия на момент написания — Bootstrap 5, которая отличается улучшенной производительностью, отказом от зависимости от jQuery и упрощением структуры.

Основные компоненты Bootstrap:

1. Сетка (Grid System):

- Система сеток на основе flexbox или CSS grid, позволяющая легко адаптировать страницы для различных разрешений экранов.

2. Стилизация элементов:

- Готовые стили для кнопок, таблиц, форм, текста, изображений и других HTML-элементов.

3. Компоненты:

- Набор часто используемых элементов интерфейса, таких как навигационные панели, выпадающие списки, модальные окна, карусели и многое другое.

4. JavaScript-библиотеки:

- Включают функционал для динамических компонентов: модальные окна, выпадающие меню, переключатели вкладок, аккордеоны и другие интерактивные элементы.

Преимущества Bootstrap:

1. Адаптивность и кроссплатформенность:

- Сайты, созданные с использованием Bootstrap, автоматически адаптируются к различным устройствам (смартфоны, планшеты, компьютеры) благодаря встроенной сетке и медиазапросам.

2. Быстрое прототипирование:

- Наличие готовых компонентов и стилей позволяет быстро создавать макеты и прототипы веб-страниц, что экономит время на этапе разработки.

3. Единообразный дизайн:

- Использование стандартных стилей и компонентов гарантирует, что интерфейс будет выглядеть одинаково на всех страницах сайта.

#### 4. Широкая документация и сообщество:

- Bootstrap имеет обширную документацию с примерами использования, что упрощает обучение и внедрение. Сообщество разработчиков активно делится своими наработками, шаблонами и модулями.

#### 5. Совместимость с браузерами:

- Фреймворк протестирован и оптимизирован для работы в основных браузерах, включая Google Chrome, Mozilla Firefox, Microsoft Edge и Safari.

#### 6. Поддержка кастомизации:

- Использование системы переменных (CSS Variables) позволяет легко настроить дизайн под нужды проекта. Можно изменить цветовую палитру, размеры шрифтов и другие параметры.

#### 7. Безопасность и отказ от jQuery (в Bootstrap 5):

- Новая версия Bootstrap избавилась от зависимости от jQuery, что делает приложения быстрее и безопаснее.

#### 8. Бесплатное использование:

- Bootstrap распространяется под лицензией MIT, что делает его полностью бесплатным и открытым для коммерческого и некоммерческого использования.

#### Недостатки Bootstrap:

##### 1. Ограниченность уникального дизайна:

- Часто проекты на Bootstrap выглядят однотипно из-за использования стандартных компонентов. Требуется дополнительная кастомизация для создания уникального интерфейса.

##### 2. Размер файлов:

- Подключение полного пакета Bootstrap может увеличить объём загружаемых ресурсов, что может замедлить загрузку страницы.

### 3. Переизбыток ненужного функционала:

- В проекте может быть неиспользуемая часть фреймворка, что увеличивает сложность оптимизации.

Примеры использования Bootstrap:

- Разработка корпоративных сайтов и блогов.
- Создание административных панелей и информационных панелей (dashboards).
- Быстрое создание лендингов и MVP (минимально жизнеспособных продуктов).
- Образовательные платформы и сайты электронной коммерции.

Bootstrap — это мощный инструмент для быстрой разработки адаптивных веб-приложений с минимальными затратами времени и усилий. Благодаря простоте в использовании, обширной документации и сообществу, фреймворк остаётся одним из самых популярных выборов среди веб-разработчиков.

## 2.4 Описание базы данных

Структура базы данных.

База данных построена с использованием реляционной модели.

Основные таблицы:

1. User - пользователь:
  - id (ключ)
  - username

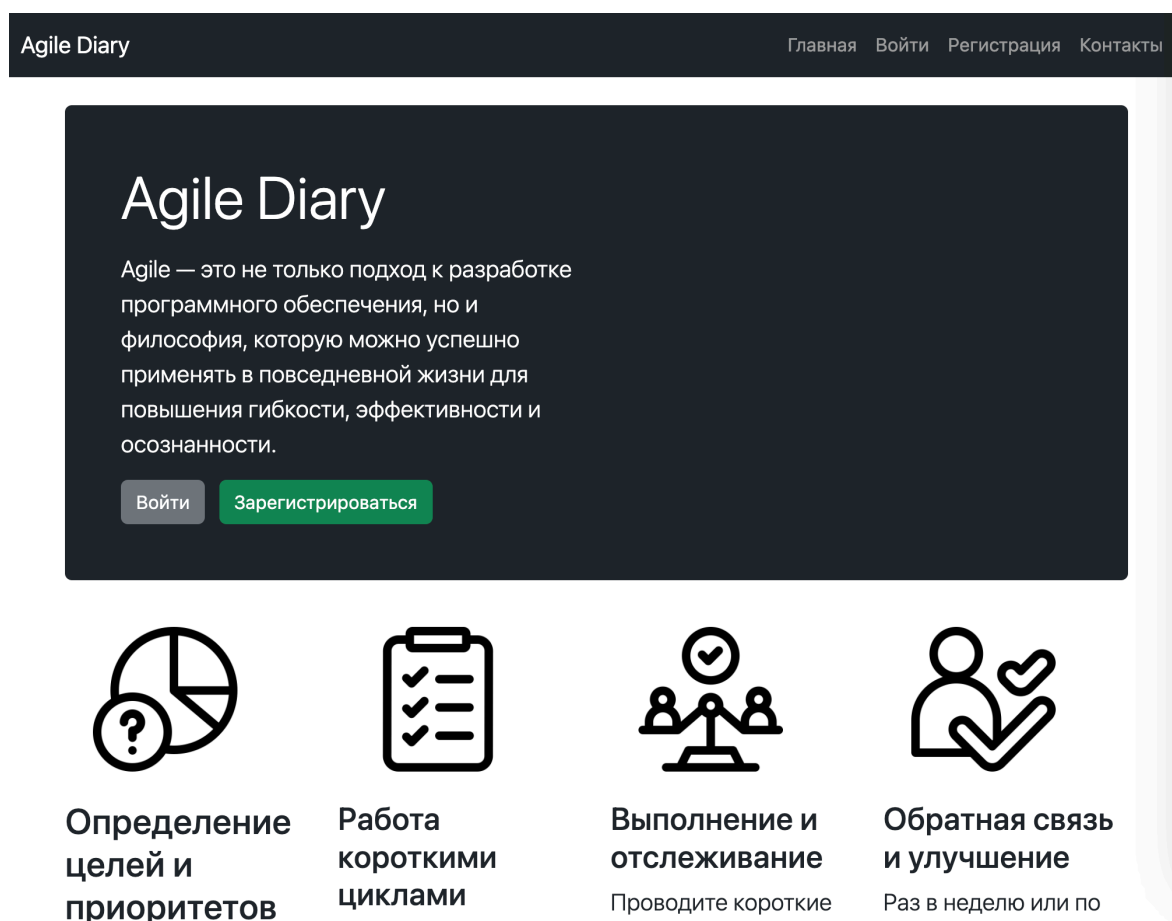
- email
- password (хэширован)
- 2 . Goal - цель:
  - id (ключ)
  - user
  - title
  - description
  - created\_at (дата создания)
  - is\_completed (статус выполнения)
  - is\_archived (статус архивирования)
- 3 . DiaryEntry - ежедневные записи:
  - id (ключ)
  - user
  - date
  - gratitude (список благодарностей)
  - achievements (список достижений)
  - thoughts (текстовые заметки)
- 3 . Task - подзадачи у цели:
  - id (ключ)
  - title
  - project (связь с Goal)
  - status (completed/uncompleted)
- 4 . Feedback - обратная связь:
  - id (ключ)
  - name
  - email
  - Text

Преимущества выбранной структуры

- Простота добавления новых записей и целей.
- Чёткие связи между пользователями и их данными.
- Возможность масштабирования, например, добавления новых типов записей.

## 2.5 Этапы реализации

### 2.5.1 Проектирование интерфейса



Использованы подходы Agile-разработки, что позволило собрать требования от целевой аудитории и создать удобный интерфейс.



• **Главная страница:** информация о приложении, кнопки регистрации и входа, если пользователь не авторизован (рис.2.1) и кнопка перехода на страницу дневника, если пользователь авторизован (рис. 2.2)

Рис 2.1 - Главная страница с кнопками входа и регистрации

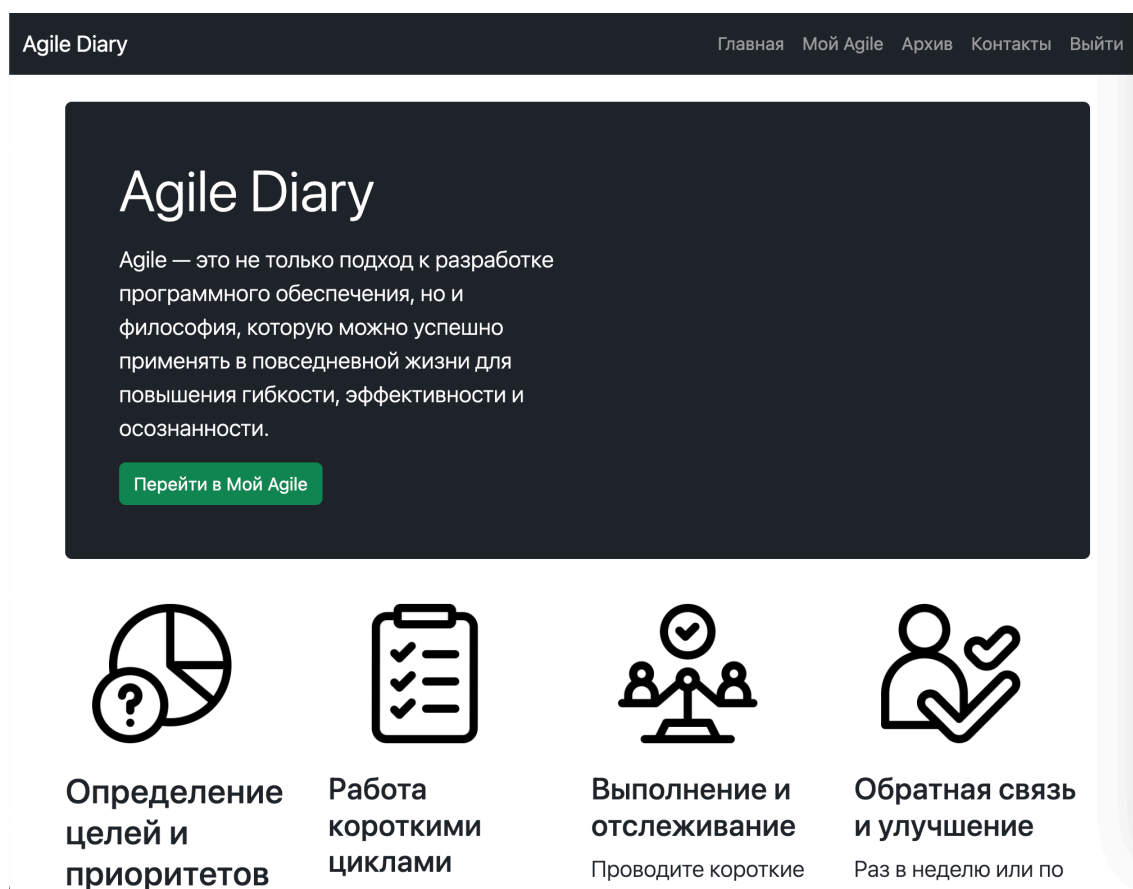


Рис 2.2 - Главная страница с кнопкой перехода на страницу дневника

• **Страница дневника:** карточки с записями, кнопка для добавления новой записи и кнопка добавления новой цели. У целей кнопка добавления задач, которые можно отмечать как выполненные или удалять. Кнопка «выполнить», если цель выполнена (рис. 2.3) и кнопка

«вернуть в работу», если цель требует доработки (рис 2.4). Кнопка «В архив», если цель более не актуальна. Кнопка редактирования, для внесения корректировок в цель. Новый пользователь увидит пустую страницу с возможностью добавления новых целей или записей (рис 2.5)

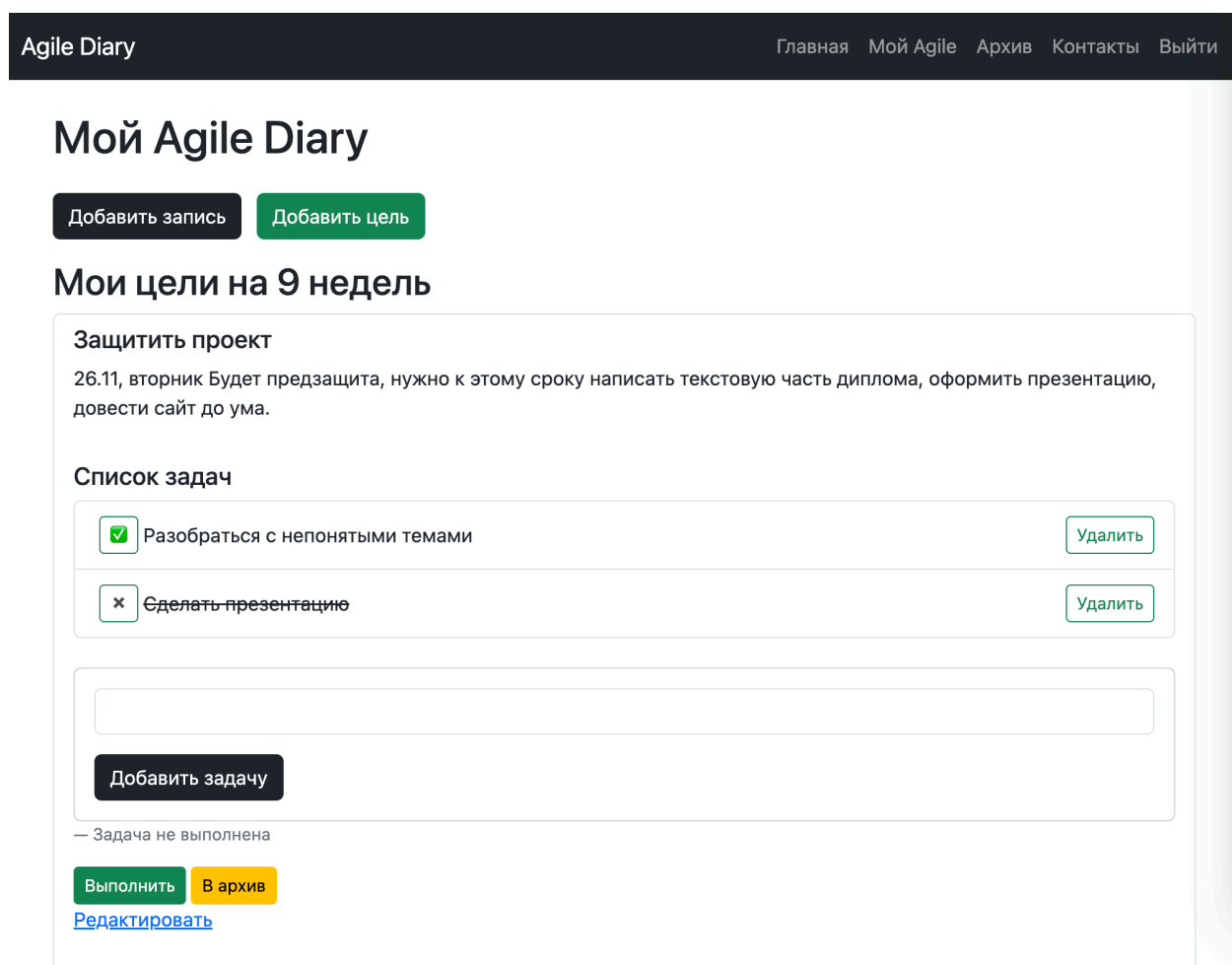


Рис 2.3 - Страница дневника с активной кнопкой «выполнить»

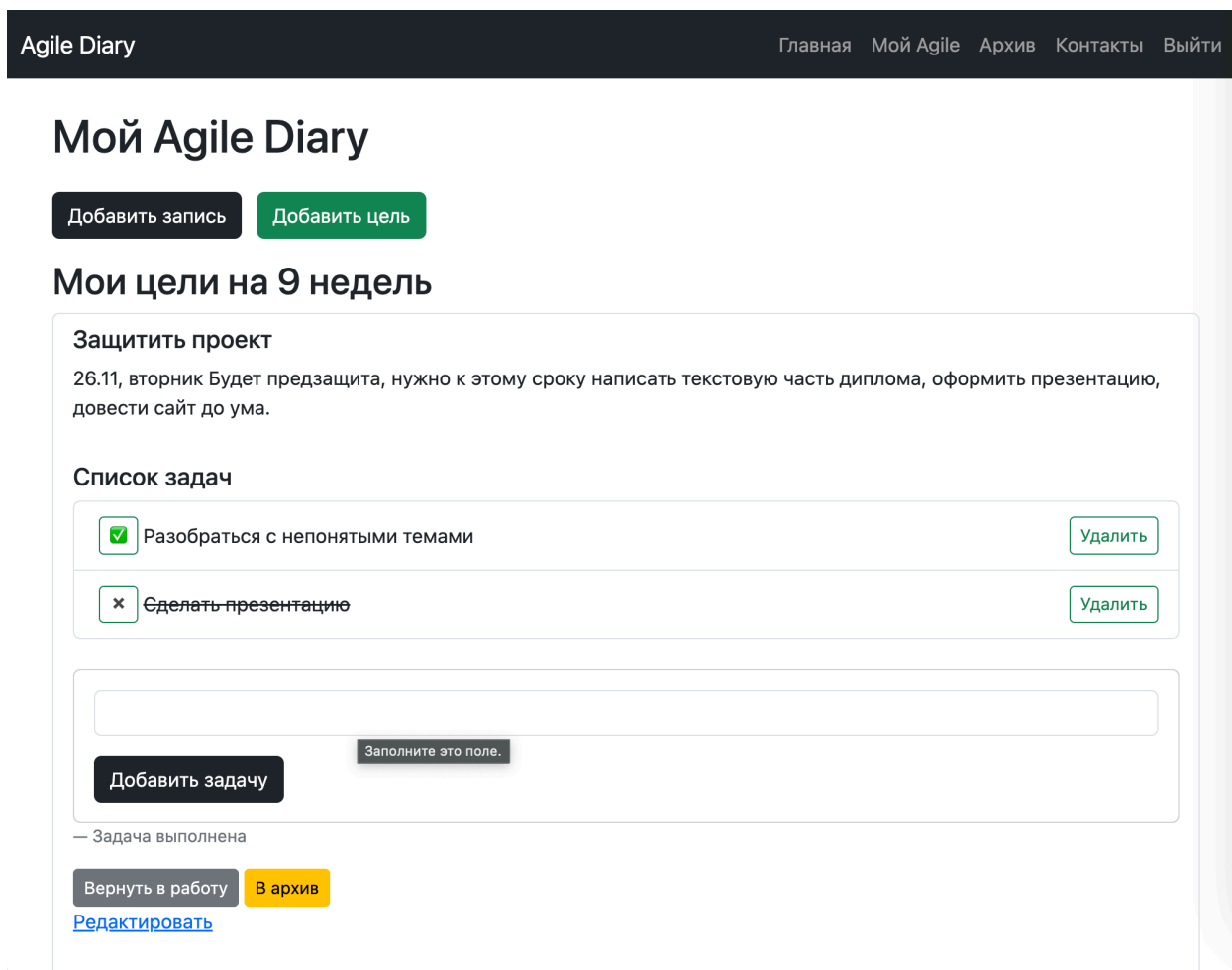


Рис 2.4 - Страница дневника с активной кнопкой «Вернуть в работу»

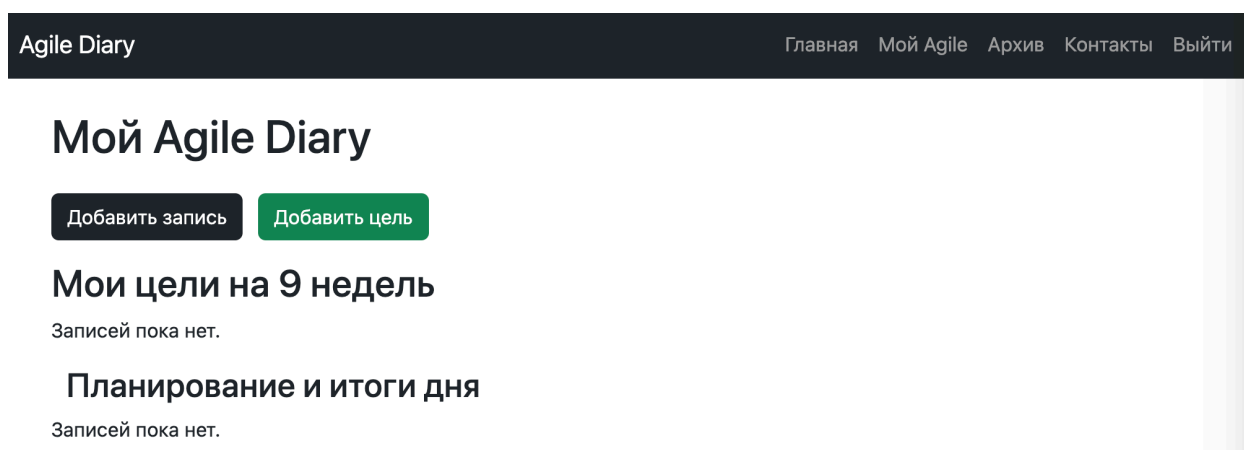


Рис 2.5 - Страница нового пользователя

- **Страница архива:** список архивированных целей (рис 2.6). Из архива можно вернуть цель в работу и она появится на странице дневника.

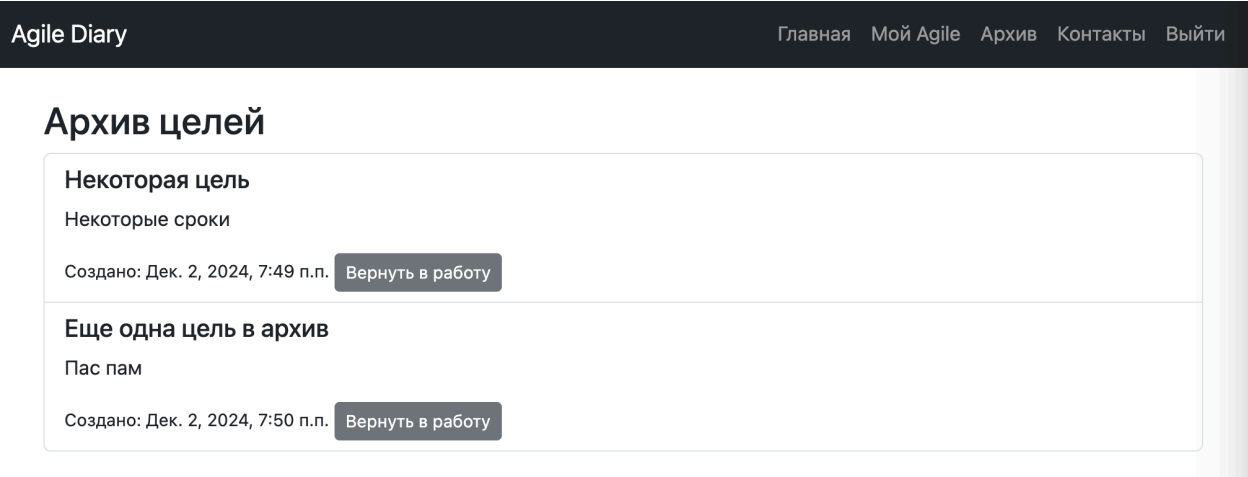


Рис 2.6 - Страница архивированных целей

- **Страница контактов и обратной связи:** контактная информация и форма для обратной связи с кнопкой «отправить» (рис. 2.7).

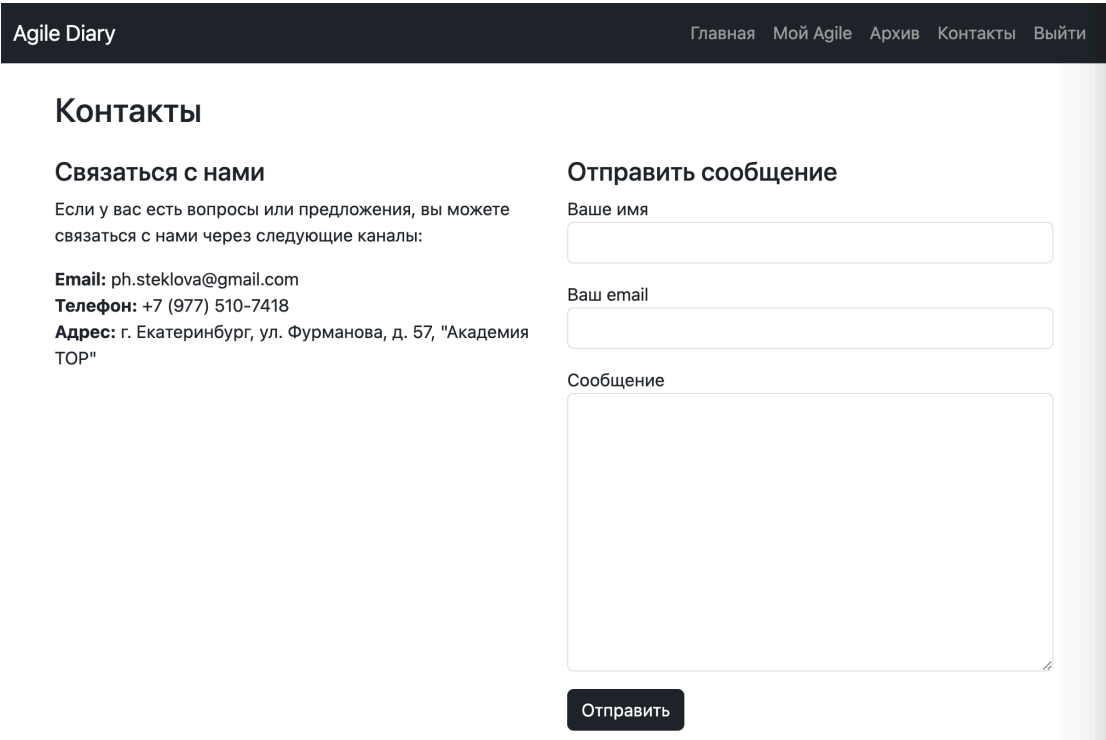
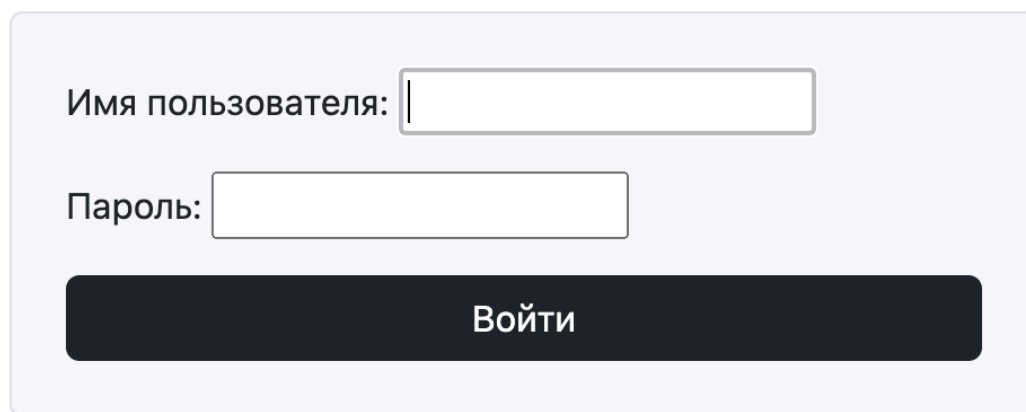


Рис 2.7 - Страница контактов и обратной связи.

### 2.5.2 Разработка функционала

- Реализованы формы для входа (рис 2.8) и регистрации (рис 2.9).

## Вход



The login form is contained within a light purple rounded rectangle. It features two input fields: the first is labeled 'Имя пользователя:' and the second is labeled 'Пароль:'. Below these fields is a dark blue button with the text 'Войти' in white.

Имя пользователя:

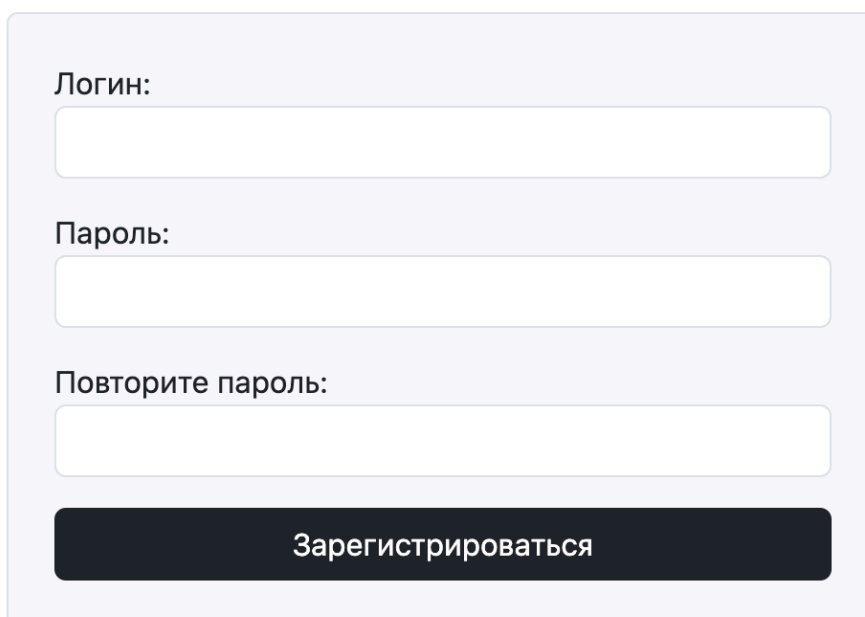
Пароль:

**Войти**

Рис

2.8 - Форма для входа

## Регистрация



The registration form is contained within a light purple rounded rectangle. It features three input fields: the first is labeled 'Логин:', the second is labeled 'Пароль:', and the third is labeled 'Повторите пароль:'. Below these fields is a dark blue button with the text 'Зарегистрироваться' in white.

Логин:

Пароль:

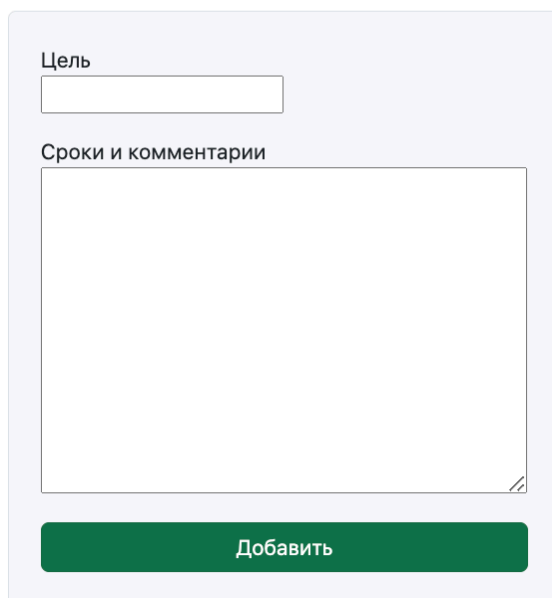
Повторите пароль:

**Зарегистрироваться**

Рис 2.9 - Форма для регистрации

- Встроен механизм управления целями: форма добавления (рис. 2.10), форма редактирования (рис. 2.11), архивирование.

## Добавить цель

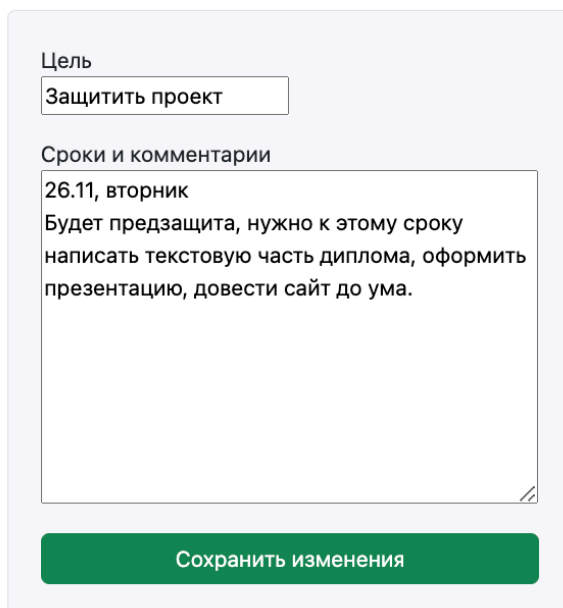


The form for adding a goal is titled "Добавить цель". It contains two main input areas: a text field labeled "Цель" (Goal) and a larger text area labeled "Сроки и комментарии" (Deadlines and comments). At the bottom of the form is a green button labeled "Добавить" (Add).

Рис 2.10 - Форма

добавления цели

## Редактировать цель



The form for editing a goal is titled "Редактировать цель". It contains two main input areas: a text field labeled "Цель" (Goal) with the text "Защитить проект" (Defend the project) and a larger text area labeled "Сроки и комментарии" (Deadlines and comments) with the text "26.11, вторник. Будет предзащита, нужно к этому сроку написать текстовую часть диплома, оформить презентацию, довести сайт до ума." (26.11, Tuesday. There will be a pre-defense, need to write the text part of the diploma by this deadline, prepare a presentation, bring the site to the finish line). At the bottom of the form is a green button labeled "Сохранить изменения" (Save changes).

Рис 2.11 - Форма редактирования цели

- Внутри цели добавлена форма добавления задач (рис. 2.12)

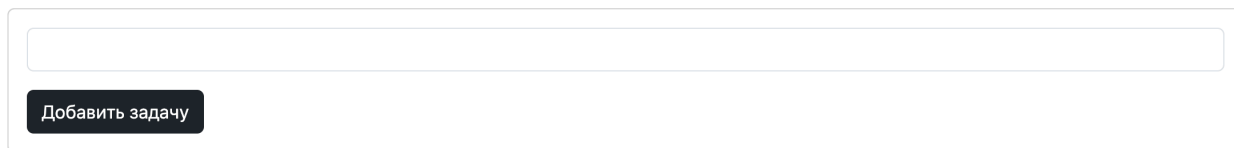
The image shows a user interface element for adding tasks. It consists of a light gray rectangular container. Inside the container, at the top, is a horizontal input field with rounded ends. Below the input field, on the left side, is a dark gray button with the white text "Добавить задачу" (Add task).

Рис 2.12 - Форма добавления задач

- Добавлен функционал дневника для записи благодарностей, достижений и мыслей (рис. 2.13) и их редактирования (рис 2.14).
- Добавлена форма для обратной связи (рис 2.7).

### 2.5.3 Тестирование

Приложение тестировалось на:

- Работоспособность всех маршрутов.
- Корректное сохранение данных в базе.
- Отображение интерфейса на мобильных устройствах.



## Добавить запись

Благодарности дня

Напишите 3 пункта благодарности

Главные достижения дня

Напишите 3 достижения

Цели на завтрашний день

Что сработало хорошо? Что можно улучшить?  
Какие планы на следующий день?

Сохранить

## Редактировать запись

Благодарности дня

1) Первая запись благодарности  
2) Вторая запись благодарности  
3) Третья запись благодарности

Главные достижения дня

1) Первая запись достижения  
2) Вторая запись достижения  
3) Третья запись достижения

Цели на завтрашний день

Какой-то текст цели Цель на день и анализ  
прошедшего дня

Сохранить изменения

Р

Рис 2.13 - Форма добавления  
записи

Рис 2.14 - Форма редактирования  
записи



## 2.6 Экономические расчёты

### Затраты на разработку

#### 1.Проектирование:

- Время: 10 часов
- Ставка: 500 руб./час
- Итого: 10,000 руб.

#### 2.Разработка:

- Время: 80 часов
- Ставка: 500 руб./час
- Итого: 40,000 руб.

#### 3.Тестирование:

- Время: 3 часа
- Ставка: 500 руб./час
- Итого: 1,500 руб.

Общий бюджет: 51,500 руб.

### Затраты на хостинг и домен

- VPS-хостинг: 5,000 руб./год
- Домен: 1,000 руб./год

### Потенциальная монетизация

- Реклама: прогнозируемая выручка до 50,000 руб./год.
- Премиум-аккаунты: до 50,000 руб./год.

## Глава 3. Обобщение результатов

### 3.1 Предложенное решение

Итоговое описание приложения

Веб-приложение "Agile-дневник" — это современная система, разработанная для управления личными целями, анализа достижений и ведения личных заметок. Оно интегрирует ключевые функции, необходимые пользователю для самоорганизации:

- Управление целями: добавление, редактирование, архивирование и отметка выполнения задач.
- Ведение дневника: запись благодарностей, достижений и мыслей.
- Управление пользователями: регистрация, авторизация и управление профилем.

Архитектура приложения

Приложение состоит из следующих ключевых компонентов:

#### 1. Клиентская часть:

Использован фреймворк Bootstrap для создания адаптивного пользовательского интерфейса. Интерфейсы разделены на логические блоки: регистрация, управление целями, дневник и архив.

#### 2. Серверная часть:

Разработана с использованием фреймворка Django. Сервер обрабатывает запросы пользователей, управляет базой данных и предоставляет данные клиентской части через маршруты.

#### 3. База данных:

Реализована с помощью SQLite на этапе разработки. В боевой эксплуатации может быть заменена на PostgreSQL для повышения масштабируемости.

Функционал

Регистрация и авторизация:

- Пользователь может создать учётную запись с указанием имени пользователя и email.
- Реализована защита паролей с использованием хэширования.

Работа с целями:

- Возможность добавления целей с указанием сроков выполнения и описания.
- Управление статусами задач: активные, выполненные, архивные.

Ведение дневника:

- Возможность записи благодарностей, достижений и мыслей за день.
- Просмотр и редактирование записей.

Архив:

- Раздел для просмотра выполненных целей и старых записей.

### *3.2 Последствия внедрения*

Продуктивность пользователей

Использование приложения способствует:

- Улучшению навыков управления временем благодаря визуализации задач и их статусов.
- Повышению осознанности за счёт ежедневной рефлексии.

Личная эффективность

Пользователи отмечают:

- Лёгкость в постановке и достижении целей.
- Увеличение мотивации через анализ прогресса и благодарностей.

Долгосрочные выгоды

- Формирование полезной привычки планирования и анализа.

- Возможность использования данных дневника для анализа личного роста.

### *3.3 Рекомендации по использованию*

Интеграция с другими инструментами

**1. Интеграция с календарями:**

Синхронизация с Google Calendar или Outlook для автоматического отображения сроков задач.

**2. API для внешних приложений:**

Разработка RESTful API для возможности подключения внешних инструментов анализа.

Разработка мобильного приложения

Для удобства пользователей можно разработать мобильное приложение на основе существующего веб-приложения. Это упростит доступ к функционалу и увеличит количество пользователей.

Дальнейшее развитие функционала

- Добавление напоминаний о задачах.
- Реализация функции анализа данных, предоставляющей пользователям отчёты о продуктивности.
- Расширение раздела архива, включая функции поиска и фильтрации.

## *Заключение*

Дипломная работа была посвящена разработке веб-приложения "Agile-дневник", предназначенного для управления личными целями, ежедневной рефлексии и анализа продуктивности. В процессе выполнения проекта были выполнены все поставленные задачи, начиная с проектирования системы и заканчивая её практической реализацией. Приложение предоставляет пользователю удобный инструмент, который способствует улучшению навыков самоорганизации, осознанности и личной эффективности.

### Основные этапы работы

Работа над проектом включала несколько ключевых этапов:

#### **1. Разработка архитектуры приложения:**

Был проведён анализ существующих решений, что позволило выделить основные потребности пользователей и спроектировать оптимальную структуру системы.

Архитектура приложения была построена на основе современного веб-фреймворка Django, который обеспечивает высокую производительность и гибкость разработки.

#### **2. Создание базы данных:**

Проектирование базы данных включало определение ключевых сущностей: пользователей, целей и записей дневника.

Структура базы данных была реализована с использованием SQLite на этапе разработки.

#### **3. Реализация функционала:**

- Разработаны формы для регистрации и авторизации пользователей.
- Реализованы интерфейсы для добавления, редактирования и архивирования целей.



- Созданы страницы для ведения дневника, позволяющие пользователю фиксировать благодарности, достижения и мысли.
- Реализована страница архива, где пользователь может просматривать выполненные цели и старые записи.

#### 4. Создание пользовательского интерфейса:

Для повышения удобства использования применялась библиотека Bootstrap, которая обеспечила адаптивный и эстетичный дизайн.

#### 5. Тестирование и оптимизация:

Приложение было протестировано для выявления ошибок и обеспечения стабильной работы при увеличении нагрузки.

### Полученные результаты

В результате выполнения работы было разработано полнофункциональное веб-приложение, которое включает следующие возможности:

- Регистрация и авторизация пользователей с защитой данных.
- Управление целями, включая добавление, редактирование, архивирование и установку статуса "выполнено".
- Ведение записей дневника, направленных на рефлекссию и анализ личных достижений.
- Просмотр и управление архивом завершённых целей и записей.

Приложение отличается интуитивно понятным интерфейсом, надёжной серверной частью и высокой производительностью.

### Практическая значимость

Разработанное приложение обладает высокой практической ценностью, так как предоставляет пользователям эффективный инструмент для самоорганизации. Оно может быть полезным для различных категорий пользователей, включая студентов, специалистов, фрилансеров и других людей, заинтересованных в управлении своим временем и целями.

"Agile-дневник" объединяет принципы Agile-методологии с рефлексивным подходом, что позволяет пользователям не только планировать задачи, но и анализировать свой прогресс, отмечать достижения и благодарности.

Преимущества приложения:

- Простота и гибкость использования.
- Возможность адаптации под различные потребности.
- Высокая доступность за счёт использования веб-технологий.

Рекомендации

Разработка "Agile-дневника" является первым этапом более масштабного проекта. В дальнейшем приложение может быть расширено и улучшено за счёт следующих дополнений:

**1. Добавление новых функций:**

- Синхронизация с внешними календарями, такими как Google Calendar или Outlook.
- Напоминания о задачах с помощью уведомлений.
- Возможность совместного использования целей и дневников с другими пользователями.

**2. Мобильное приложение:**

Разработка мобильного приложения на основе веб-версии значительно увеличит аудиторию пользователей.

**3. Интеграция с другими системами:**

Создание API для интеграции с популярными инструментами самоорганизации, такими как Trello или Notion.

**4. Расширение аналитического функционала:**

Введение графиков и отчётов, которые помогут пользователям оценивать их продуктивность за разные периоды времени.

Веб-приложение "Agile-дневник" является успешным примером использования современных технологий для решения задачи личной

самоорганизации. Оно совмещает удобство использования с функциональностью и гибкостью, что делает его универсальным инструментом для широкой аудитории пользователей.

Внедрение приложения в практику способствует развитию навыков планирования и анализа, а также повышению осознанности пользователей.

Дальнейшее развитие и масштабирование "Agile-дневника" откроет дополнительные возможности для пользователей и обеспечит приложение новым функционалом, отвечающим современным запросам.

## Список литературы

1. Официальная документация Django:
  - Django Software Foundation. *Django Documentation*. URL: <https://docs.djangoproject.com/en/stable/>.
2. Официальная документация Bootstrap:
  - Bootstrap Team. *Bootstrap Documentation*. URL: <https://getbootstrap.com/docs/>.
3. Книги и статьи о продуктивности и самоорганизации:
  - Дэвид Аллен. "Как привести дела в порядок. Искусство продуктивности без стресса." Манн, Иванов и Фербер, 2017.
  - Стивен Кови. "7 навыков высокоэффективных людей." Альпина Паблишер, 2016.
  - Джеймс Клир. "Атомные привычки. Как приобрести хорошие привычки и избавиться от плохих." Манн, Иванов и Фербер, 2018.
  - Леван Абашидзе. "Продуктивность. Как делать больше за меньшее время." Эксмо, 2020.
4. Статьи о продуктивности и саморазвитии:
  - Medium. "Почему ежедневное ведение дневника — лучший инструмент для продуктивности." URL: <https://medium.com/@journalwriter/why-daily-journaling-is-the-best-productivity-tool-df2f1b3b79b8>.
  - Михаил Хазин. "Как повысить свою личную эффективность." Издательство «Алгоритм», 2018.
  - Анна Лобанова. "Как стать продуктивнее, или что мы знаем о том, как управлять своим временем." URL: <https://www.kp.ru/putevoditel/kultura/produktivnost/>.
  - Оксана Жданова. "Техника продуктивности и самоорганизации." Издательство «Эксмо», 2019.
5. Литература по Agile-методологиям:

- Джефф Сазерленд. "Scrum: искусство делать в два раза больше за меньшее время." Манн, Иванов и Фербер, 2015.
  - Майк Коун. "Применение пользовательских историй в Agile-разработке." Вильямс, 2011.
6. Статьи о веб-разработке и инструментах:
- Real Python. "Django Tutorials for Beginners and Professionals." URL: <https://realpython.com/tutorials/django/>.
  - Смэшинг Магазин. "*Как начать работать с Bootstrap.*" URL: <https://www.smashingmagazine.com/2020/02/bootstrap-beginners-guide/>.
  - Mozilla Developer Network. "*Введение в HTML.*" URL: <https://developer.mozilla.org/ru/docs/Web/HTML>.
7. Документация SQLite:
- SQLite Consortium. *SQLite Documentation*. URL: <https://sqlite.org/docs.html>.
8. Исследования по личной эффективности:
- Американская психологическая ассоциация. "*Влияние ведения дневника на психическое здоровье.*" URL: <https://www.apa.org/monitor/2021/01/journaling>.

*Лист замечаний*

<i>Ф.И.О.</i>	<i>Суть замечаний, оценка и подпись</i>