#### Содержание

Введение 4

[1 Анализ](#_Toc406878074) задачи 6

[1.1 Постановка задачи](#_Элементы_синтаксиса) 6

[1.2](#_Как_выглядит_среда) Диаграмма вариантов использования 9

[2 Проектирование задачи](#_Достоинства_и_недостатки) 10

2.1 Выбор стратегии разработки и модели жизненного цикла…………….10

2.2 Инструменты разработки…………………………………………………14

2.3 Разработка UML-диаграмм 14

2.4 Разработка пользовательского интерфейса 15

2.5 Тесты на использование……………………………………………….….15

2.6 Разработка плана над проектом…………………………………………..15

3 Реализация 16

3.1 Руководство программиста 16

[4 Тестирование](#_Достоинства_и_недостатки) 17

4.1 Отчет о результатах тестирования 17

5 Руководство пользователя 18

5.1 Общие сведения…………………………………………………………...18

5.2 Выполнение программы…………………………………………………..18

[Заключение](#_Toc406878088) 22

[Список использованных источников](#_Toc406878086) 23

[Приложение А](#_Toc406878088)  24

Приложение Б 26

Приложение В 28

Приложение Г 30

Приложение Д 32

Приложение Е 34

Приложение Ж 36

**Введение**

В рамках учебной практики была поставлена задача разработки автоматизированного рабочего места для диспетчера, что представляет собой важное звено в области оперативного управления и контроля.

Цель данного проекта заключается в создании веб-приложения, предназначенного для оптимизации рабочего процесса диспетчера. Разрабатываемый ресурс предоставляет возможность эффективного взаимодействия с данными, обеспечивает оперативный мониторинг и управление задачами.

Данное веб-приложение рассчитано на широкий круг пользователей, включая диспетчеров, ответственных за планирование и распределение задач, а также исполнителей, выполняющих эти задачи.

Одним из ключевых аспектов данного проекта является создание системы, которая не только улучшает текущий процесс управления задачами, но и предоставляет полезные инструменты для анализа и оптимизации рабочего процесса. Автоматизированное рабочее место диспетчера позволяет эффективно распределять задачи, отслеживать их выполнение и обеспечивать взаимодействие между участниками.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки web-ресурса. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации.

“Реализация задачи” – это третий раздел отчета по практике, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Описано руководство программиста и будет предоставлена диаграмма компонентов.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется web-ресурсам. Будут смоделированы все возможные действия пользователя при работе с web-ресурсом, начиная от входа на сайт заканчивая закрытием вкладки.

В разделе “Применение” будет описано назначение, область применения, среда функционирования данного программного продукта.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В разделе “Список используемых источников” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведены листинг программы с необходимыми комментариями и диаграммы.

1. **Анализ задачи**
   1. **Постановка задачи**
2. **. Организационно-экономическая сущность задачи**

Название темы: на практику была поставлена задача разработать web- ресурс «Автоматизированное рабочее место диспетчера».

Цель разработки: создание системы автоматизации рабочего места диспетчера для эффективного управления задачами, их распределения и мониторинга выполнения. Продукт должен повысить производительность, снизить время обработки запросов, и обеспечить прозрачность взаимодействия между диспетчерами и исполнителями.

Назначение: оптимизация рабочего процесса диспетчера, управления задачами, их распределения, мониторинга статусов и взаимодействия с исполнителями.

Периодичность использования: будет использоваться ежедневно в течение рабочего дня. Исполнители будут использовать систему для получения задач и предоставления отчетов о выполнении.

Источники и способы получения данных:

* Ввод данных от диспетчеров: диспетчеры могут вводить новые задачи в систему, предоставляя информацию о типе задачи, адресе, сроках выполнения и особых требованиях;
* Системы мониторинга и отслеживания: интеграция с системами мониторинга и отслеживания может предоставлять данные о статусах задач и других важных метриках;
* Cбор данных от исполнителей: Система может получать отчеты и данные от исполнителей, такие как время выполнения задачи и другие сведения.

Информационная связь с другими задачами:

* Автоматическое обновление статусов: при завершении задачи исполнителем автоматическое обновление статуса в системе и генерация соответствующего отчета;
* Синхронизация с календарями: интеграция с календарными приложениями для планирования задач и отображения сроков выполнения в календаре исполнителя;
* Обратная связь и коммуникация: возможность для исполнителей оставлять комментарии и обратную связь о выполнении задачи, что может повлиять на дальнейший процесс планирования.

Обзор существующих аналогичных программных продуктов:

* «Dispatcher Pro»: это программное обеспечение предоставляет функциональность для управления задачами, отслеживания местоположения ресурсов и взаимодействия с исполнителями. Однако, оно ограничено в возможностях анализа данных и генерации отчетов;
* «TaskMaster 3000»: этот продукт предлагает широкие возможности для управления задачами, но имеет сложный пользовательский интерфейс, что затрудняет его использование;
* «DispatchR»: приложение обеспечивает простоту использования и интеграцию с внешними системами, но недостаточно эффективно решает задачи мониторинга выполнения.

1. **. Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП для диспетчера и исполнителя. Диспетчер сможет подавать задачи конкретному исполнителю, также сможет следить за выполнением задачи. Исполнитель сможет получать задачи и ставить им статус выполнения (ожидание, выполнение, выполнено и т.п.).

1. **. Описание процессов с входной, выходной и условно-постоянной информацией.**

Категория пользователей и соответствующие им процессы в системе представлены следующим образом:

Диспетчер:

* Регистрация задач:

1. ввод информации о новых задачах, включая тип, сроки выполнения и особые требования;
2. входная информация включает тип задачи, адрес, сроки и особые требования;
3. выходная информация - тип задачи, адрес, сроки и особые требования;
4. постоянная информация включает данные о доступных типах задач и статусы задач;
5. процесс включает регистрацию задачи в системе.

* Планирование и распределение:

1. распределение задач между исполнителями, учитывая их текущую загрузку;
2. входная информация включает задачи, исполнителей и текущую загрузку;
3. выходная информация - информация о распределении задач;
4. постоянная информация включает справочник исполнителей и их загрузку.

* Мониторинг и отчетность:

1. мониторинг статусов исполнителей;
2. входная информация - список задач и статусы исполнителей;
3. выходная информация - отчеты о выполненных задачах и статусы исполнителей;
4. постоянная информация включает статистику по задачам и исполнителям.

Исполнитель:

* Получение задач:

1. получение информации о задачах, включая их приоритет, адрес и сроки выполнения;
2. входная информация - задачи и их приоритеты;
3. выходная информация - информация о задаче для выполнения.

* Отчеты о выполнении:

1. предоставление отчетов о выполнении задач;
2. входная информация - информация о выполнении задач;
3. выходная информация - отчет о выполнении задачи.

Все пользователи:

* Авторизация и аутентификация:

1. обеспечение безопасного доступа к системе;
2. входная информация - логин и пароль;
3. выходная информация - авторизация в системе.

**1.1.4. Нефункциональные требования**

**Требования к применению:** интерфейс должен быть понятным, функциональным и простым в использовании. Он не должен перегружать пользователя ненужной информацией и напрягать глаза яркой цветовой гаммой.

**Требования к реализации:** для реализации данного программного продукта был использован конструктор сайтов «Tilda», а также «Google Таблицы» для работы с данными.

**Требования к надежности:** веб-сайт должен иметь возможность самовосстановления после сбоя.

**Требования к интерфейсу:** Каждая страница сайта должна иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы восприятие информации пользователем былопонятным. Объекты (изображения), расположенные на экране, не должны быть слишком маленькие или замещать другие функции. Пользователь не должен испытывать какой-либо дискомфорт в плане восприятия информации, отображенной на экране. Все страницы адаптивны под разные устройства и разрешения экрана для более удобного использования данного web-ресурса.

**1.2. Диаграмма вариантов использования**

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования.

Актером или действующим лицом является любая сущность, взаимодействующая с системой извне. Это может быть как живое существо, так и любая друга система, которая может служить источником воздействия на моделируемую систему так, как определяет сам разработчик.

Вариант использования является стандартным языком UML и применяется для спецификаций общих особенностей системы и любой другой сущности. Отдельные варианты использования обозначаются на диаграмме эллипсом, в котором содержится его краткое название.

Отношение ассоциации является главным понятием языка UML и используется при построении всех графических моделей. Оно служит для обозначения роли актера и отдельном варианте использования. На диаграмме отношение ассоциации обозначается сплошной линией между актером и вариантом использования.

Для отображения взаимосвязи экземпляра отдельного варианта использования с более общим вариантом используется отношение расширения, обозначаемое направленной пунктирной линией со стрелкой от исходного варианта. Данная линия помечается ключевым словом «extend».

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данная линия помечается ключевым словом «include».

Диаграмма вариантов использования, разработанная для данного проекта, представленав Приложении А.

**2. Проектирование задачи**

**2.1. Выбор стратегии разработки и модели жизненного цикла**

Для разработки веб-ресурса следует выбрать стратегию разработки и модель жизненного цикла. Выбор осуществлялся посредством составления таблиц:

Таблица 2 – Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V- образную, 5 за RAD, 5 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: на основе результатов заполнения таблица 3 подходящей является RAD модель и инкрементная модель.

Таблица 3 – Выбор модели жизненного цикла на основе команды разработчиков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории команды разработчиков проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

Вычисления: 5 за каскадную, 5 за V-образную, 4 за RAD, 5 за инкрементную, 2 за быстрого прототипирования и 1 за эволюционную.

Итог: на основе результатов заполнения табл. 4 подходящими являются каскадная, V-образная и нкрементная модели.

Таблица 4 **–** Выбор модели жизненного цикла на основе коллектива пользователей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволю  ционная |
| Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 0 за каскадную, 0 за V-образную, 2 за RAD, 1 за инкрементную, 4 за быстрого прототипирования и 2 за эволюционную.

Итог: на основе результатов заполнения табл. 5 подходящей является модель быстрого проектирования.

Таблица 5 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
|  | | | | | | |
| Продолжение таблицы 5 | | | | | | |
| Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 4 за каскадную, 5 за V-образную, 5 за RAD, 7 за инкрементную, 5 за быстрого прототипирования и 7 за эволюционную.

Итог: на основе результатов заполнения табл. 6 подходящей является инкрементная модель.

Общий итог: в итоге заполнения таблиц – наиболее подходящей является инкрементная модель.

**2.2. Инструменты разработки**

Для разработки данного проекта будет выбрана среда разработки MS Visual Studio Code, которая является наиболее актуальной средой для создания приложений данного типа.

Разработка будет производится на таких языках программирования, как:

* HTML (HyperText Markup Language) – для создания разметки сайта;
* CSS (Cascading Style Sheets) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки;
* JS (JavaSсript) – «оживление сайта», задание реакции элементов на действия пользователя; прототипно-ориентированный сценарный язык программирования. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.
* C# (C-Sharp) – это объектно-ориентированный язык программирования, разработанный компанией Microsoft. C# широко используется для создания различных типов приложений, включая веб-приложения, приложения для Windows, службы и многое другое. C# также может быть использован в совокупности с ASP.NET для создания веб-приложений и взаимодействия с базами данных.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

* WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;
* Microsoft Office Word – для написания документации к программному продукту;

Разработка проекта будет происходить на ноутбуке со следующими параметрами:

* процессор AMD Ryzen 5 5600H, 4200 MHz, 6 ядер, 12 потоков;
* объем оперативной памяти 16.00 GB;
* объем места на жестких дисках 475 GB;
* видеокарта NVIDIA RTX 3050 Laptop 4GB;
* ОС Windows 11 Home.

**2.3. Разработка UML-диаграмм**

В первом разделе «Постановка задачи» были описаны функциональные требования к проекту. На их основе построена диаграмма последовательности.

Диаграмма последовательности – это диаграмма, предназначенная для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

Графические диаграммы последовательности имеют два представления. Одно – слева направо – в виде вертикальных линий, которые соответствуют линии жизни отдельного участка взаимодействия. Второе – вертикальная временная ось, направленная сверху вниз.

Диаграмма последовательности представлена в Приложении В.

**2.4. Разработка пользовательского интерфейса**

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемого web-ресурса. Программный продукт имеет визуальный дизайн с удобными и приятными цветами. В ходе разработки был спроектирован дизайн главной страницы web-ресурса «Olympic». Разработанная структура сайта расположена в Приложении Г.

Для организации эффективной работы пользователя нужно создать целостный программный продукт данной предметной области, в котором все компоненты будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Web-ресурс должен позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами. Все исходные данные будут разделены на несколько групп.

Прототип – это наглядная модель пользовательского интерфейса. В сущности, это «черновик», созданный на основе представления разработчика о потребностях пользователя. Итоговые страницы сайта могут отличаться от прототипа. С структурой сайта можно ознакомиться в Приложении Г.

Диаграмма деятельности представлена в Приложении В.

UX/UI макеты представлены в Приложении Ж.

**2.5. Тесты на использование**

В ходе разработки веб-ресурса были составлены тесты, которые необходимо выполнить в дальнейшем. Тесты составлены таким образом, чтобы предусмотреть максимальное количество возможных действий. Тесты описаны в таблице в Приложении Ж.

**2.6. Разработка плана работы над проектом**

Для разработки плана над проектом был использован Microsoft Excel. Microsoft Excel помог в создании плана разработки проекта, который позволил контролировать стадию выполнения. Это должно было повысить эффективность разработки. План разработки, составленный в Microsoft Excel, представлен в Приложении Д.

**3. Реализация**

**3.1. Руководство программиста**

Программный продукт разработан с использованием таких языков: Html / Css и JavaScript. Данные языки позволяют создавать полноценные web-сайты, разберёмся с каждым компонентом по подробнее:

**HTML** является языком разметки, который определяет структуру веб-страницы. Он используется для создания элементов, таких как заголовки, параграфы, изображения и ссылки.

**CSS** представляет собой язык стилей, который управляет внешним видом элементов, созданных с использованием HTML. С его помощью задаются цвета, шрифты, расположение элементов, анимации и другие аспекты дизайна.

**JavaScript** - язык программирования, придающий динамичность и интерактивность веб-страницам. Он используется для создания клиентских скриптов, обработки событий, изменения содержимого страницы в реальном времени и взаимодействия с пользователем.

Вместе эти три технологии формируют основу веб-разработки, предоставляя средства для создания структуры (HTML), стилей (CSS) и программной логики (JavaScript) на веб-страницах. Этот тройной подход обеспечивает возможность создания современных веб-приложений, способных к разнообразным и мощным функциональностям.

В разработке веб-приложений широко используются компоненты. Навигационные компоненты, такие как меню и хлебные крошки, обеспечивают удобную навигацию. Формы и вводные компоненты, включая поля ввода и календари, собирают данные от пользователей. Компоненты содержимого, такие как карточки и модальные окна, предоставляют информацию и взаимодействие. Элементы социального взаимодействия, такие как кнопки "Поделиться" и комментарии, обогащают контент. Компоненты страницы профиля, графические элементы, компоненты взаимодействия с картами, аудио и видео плееры - все они являются основными строительными блоками для разработки веб-приложений.

Этот компонентный подход упрощает разработку, делает код более читаемым и обеспечивает повторное использование. HTML задает структуру, CSS - стиль, а JavaScript - динамику, создавая тем самым современные и привлекательные веб-приложения.

**4. Тестирование**

**4.1. Отчет о результатах тестирования**

Тестирование программного обеспечения было осуществлено в соответствии с тестами на использование, описаными в разделе «Тесты на использование».

Таблица 8 – Тестирование функций

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Физический результат | Результат  тестирования |
| +Проверка поля ввода «Имя» | Корректный ввод | Ввод значений прошёл успешно | Выполнено |
| Проверка поля ввода «Пароль» | Корректный ввод и введённые значения должны быть невидны | Ввод значений прошёл успешно и значений не видно | Выполнено |
| Проверка поля ввода «Фамилия» | Корректный ввод | Ввод значений прошёл успешно | Выполнено |
| Проверка кнопки «Добавить задачу» | Переход на вкладку «Добавить задачу» | Переход прошел успешно | Выполнено |
| Проверка кнопки «Выйти» | Переход на вкладку «Авторизации прошла успешна» | Переход прошел успешно | Выполнено |
| Проверка кнопки «Просмотр задач» | Переход на вкладку «Просмотр задач» | Переход прошел успешно | Выполнено |

В результате проведения тестирования программного продукта выяснилось, что все раннее оговоренные функции были разработаны, а также протестированы. Тесты показали, что все функции работают правильно.

**5. Руководство пользователя**

**5.1. Общие сведения**

Цель данного проекта заключается в создании веб-приложения, предназначенного для оптимизации рабочего процесса диспетчера. Разрабатываемый ресурс предоставляет возможность эффективного взаимодействия с данными, обеспечивает оперативный мониторинг и управление задачами.

**5.2. Выполнение программы**

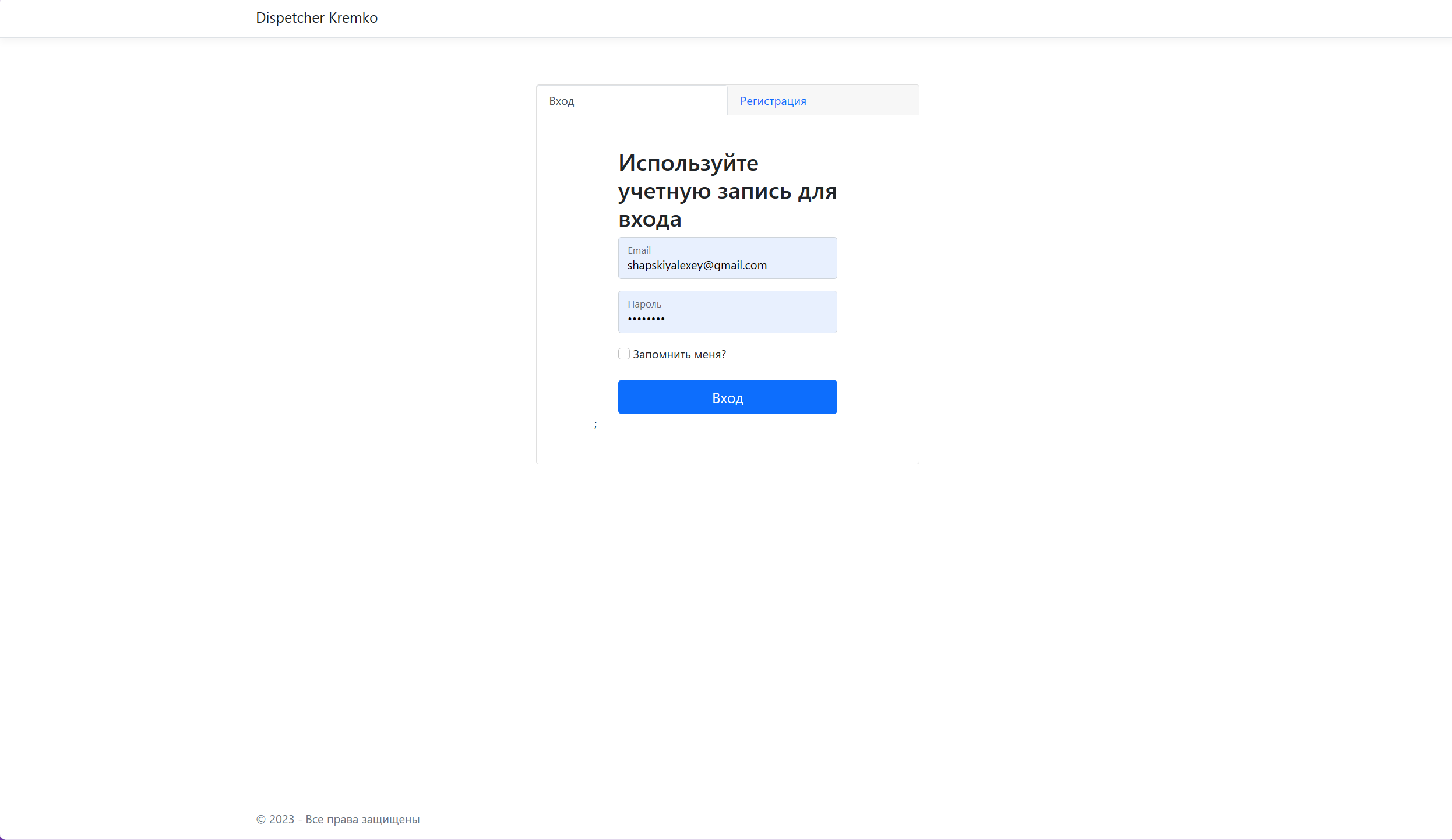
При запуске сайта появляется страница авторизации (рисунок 1). В которой пользователь может зайти в аккаунт.

Рисунок 1 – Раздел «Авторизации»

Так же гость может перейти на вкладку «Регистрации» и зарегистрироваться на сайте в роли исполнителя (рисунок 2).

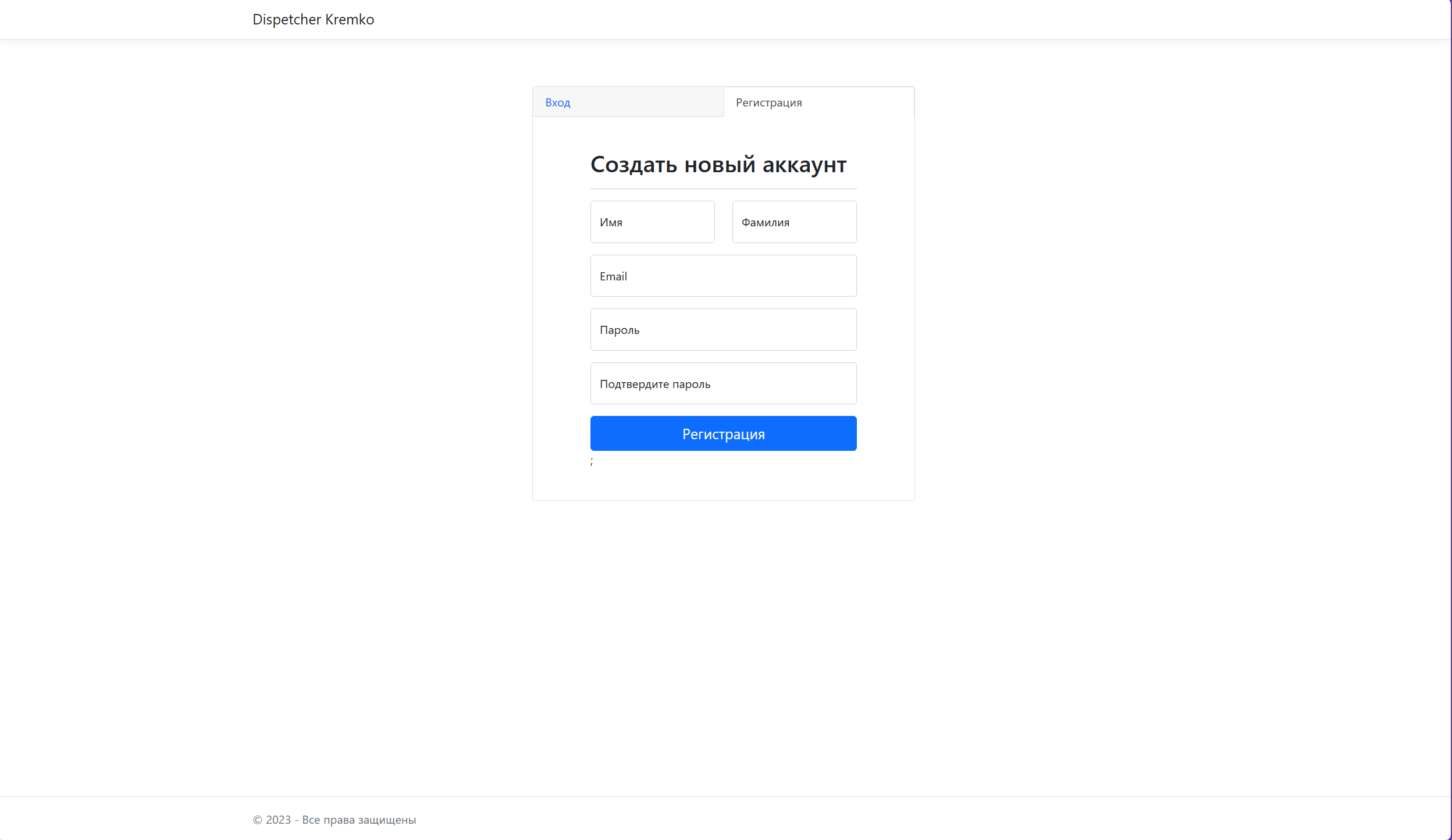
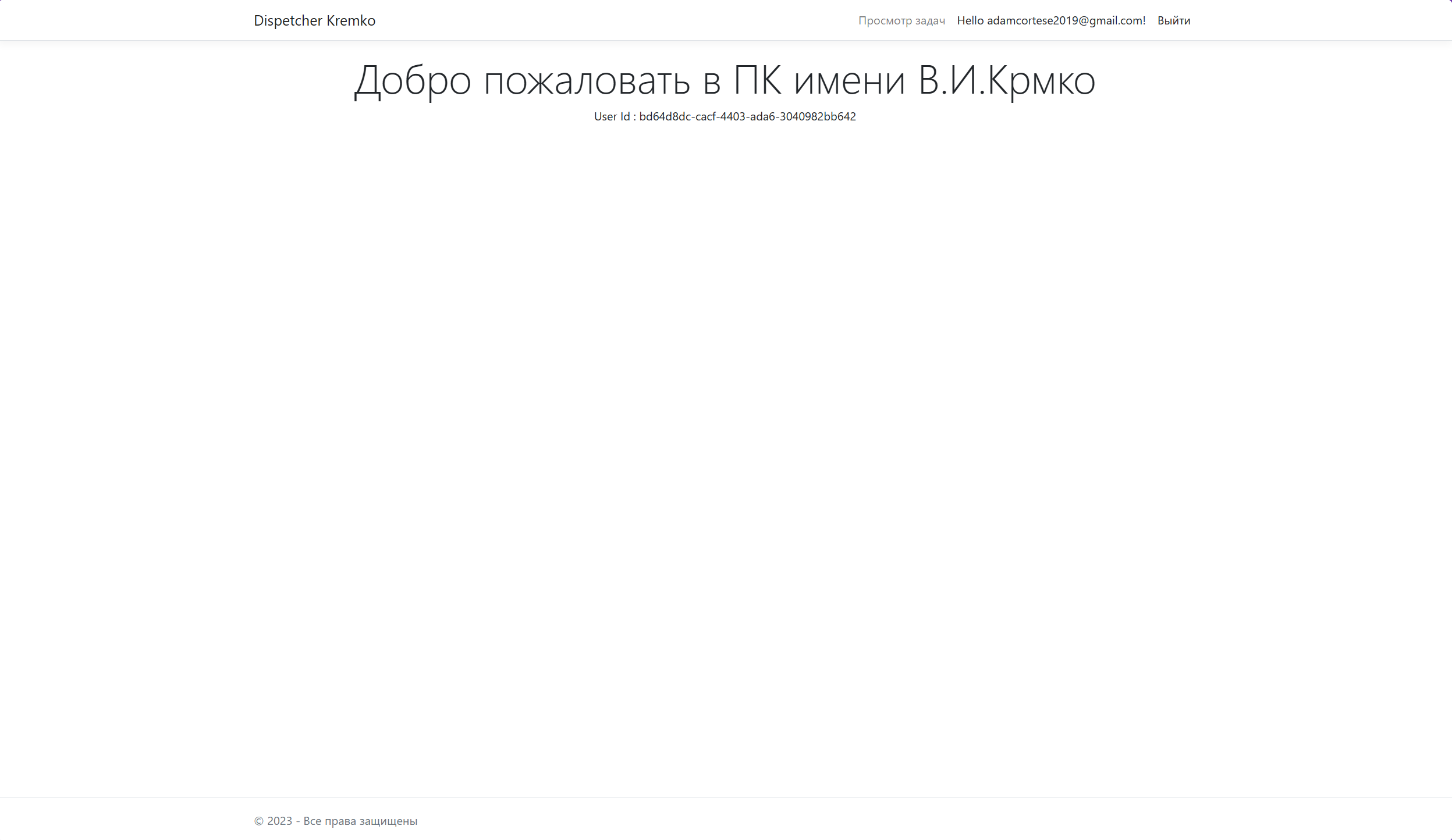


Рисунок 2 – Раздел «Регистрации»

На главной форме в роли исполнителя появляется дополнительная вкладка «Просмотр задач» (рисунок 3).

 Рисунок 3 – Раздел «Главная форма (исполнитель)»

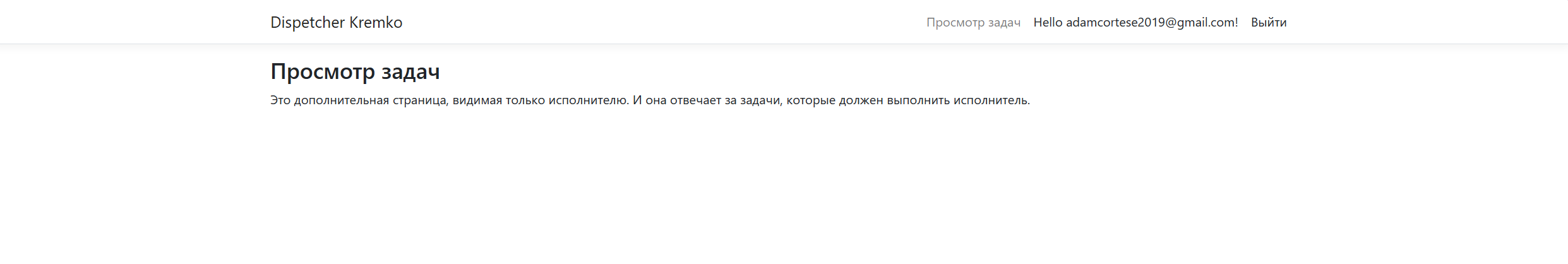
На странице «Просмотр задач» можно увидеть текст (рисунок 4).

Рисунок 4 – Раздел «Просмотр задач (исполнитель)»

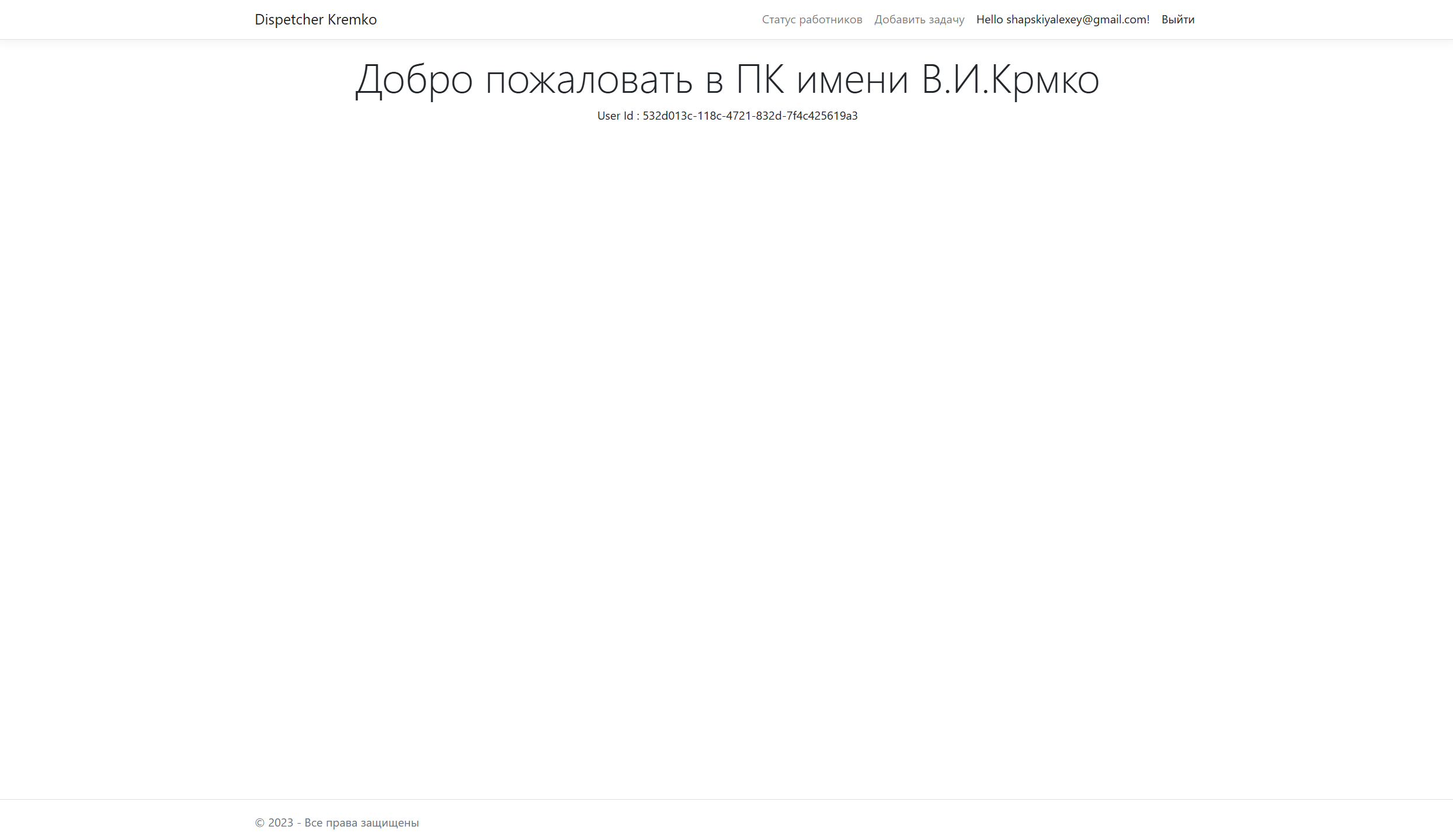
Также после авторизации в роли диспетчера, становяться доступны 2 вкладки: статус работников и добавить задачу (рисунок 5)

Рисунок 5 – Раздел «Главная форма (диспетчер)»

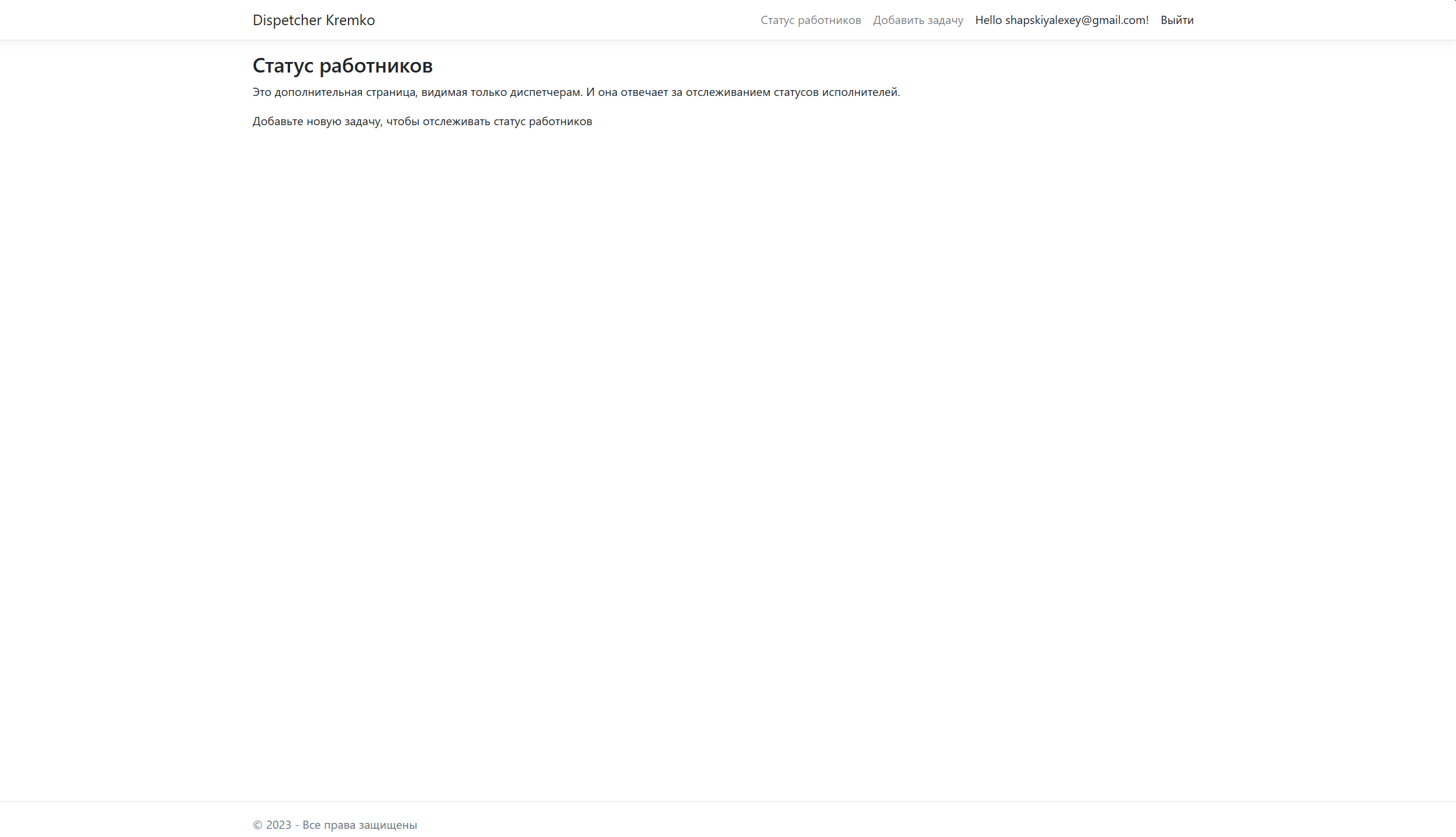
На странице «Статус работников» можно увидеть текст (рисунок 6).

Рисунок 6 - Раздел «Статус работников (диспетчер)»

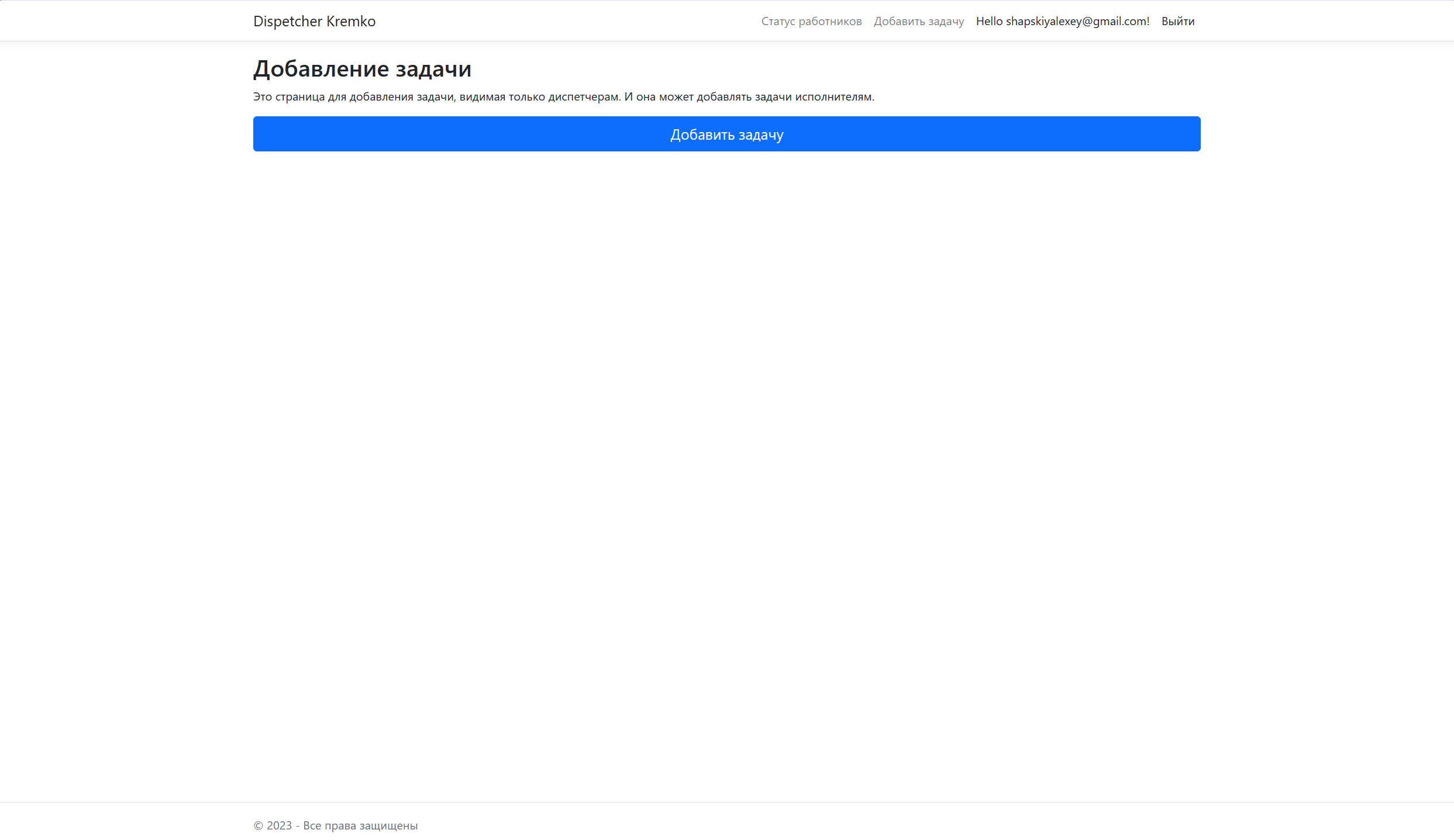
На странице «Добавить задачу» можно увидеть текст (рисунок 7).

Рисунок 7 - Раздел «Добавить задачу (диспетчер)»

**Заключение**

Целью данного проекта "DispetcerKremko" было создание автоматизированного рабочего места для диспетчера с целью оптимизации управления задачами и обеспечения эффективного взаимодействия между диспетчерами и исполнителями.

В процессе разработки были использованы современные технологии веб-программирования, такие как ASP.NET Core и Entity Framework. Кроме того, был создан интуитивно понятный интерфейс, обеспечивающий удобство использования приложения.

Проект успешно прошел этап тестирования, в ходе которого выявленные недоработки были оперативно устранены, гарантируя стабильную работу системы.

Результатом работы стало создание функционального и надежного веб-приложения, способного улучшить процессы оперативного управления и повысить эффективность взаимодействия в команде диспетчеров и исполнителей. В ходе выполнения проекта также были закреплены и расширены знания в области веб-разработки и использования инструментов, таких как ASP.NET Core.

**Список использованных источников**

Документация ASP.NET Core MVC: [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/enus/aspnet/core/mvc/overview?view=aspnetcore-5.0 – Дата доступа: 01.12.2023.

Руководство по использованию MySQL с ASP.NET Core: [Электронный ресурс] – Режим доступа: https://dev.mysql.com/doc/connector-net/en/ – Дата доступа: 27.11.2023.

**Приложение А**

Диаграмма вариантов использования

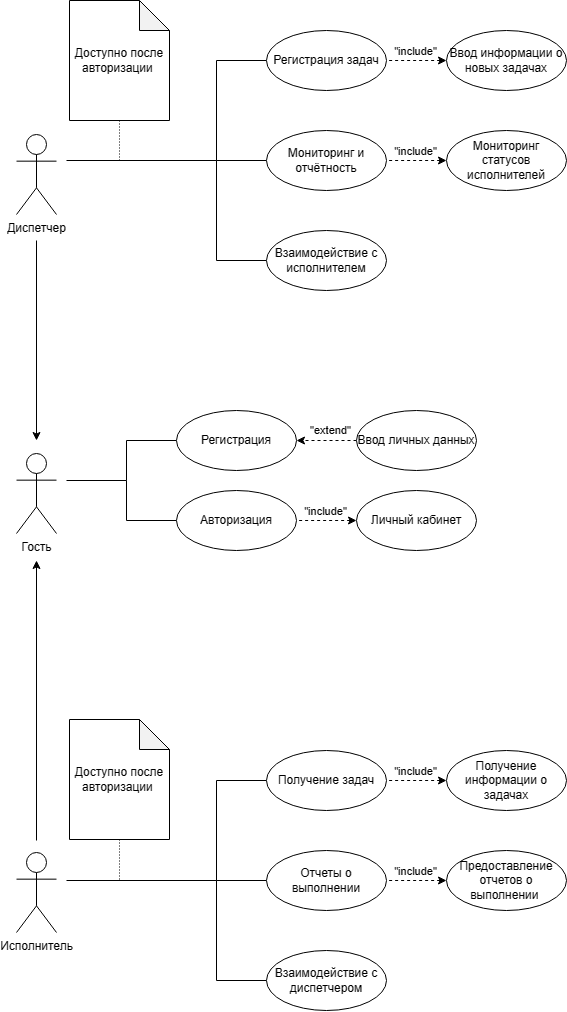


Рисунок А.1 – Диаграмма вариантов использования

**Приложение Б**

Диаграмма последовательности

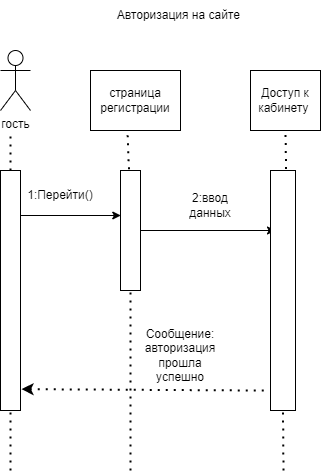


Рисунок Б.1 – Диаграмма последовательности

**Приложение В**

Диаграмма классов

****

Рисунок В.1 – Диаграмма классов

**Приложение Г**

Структура сайта

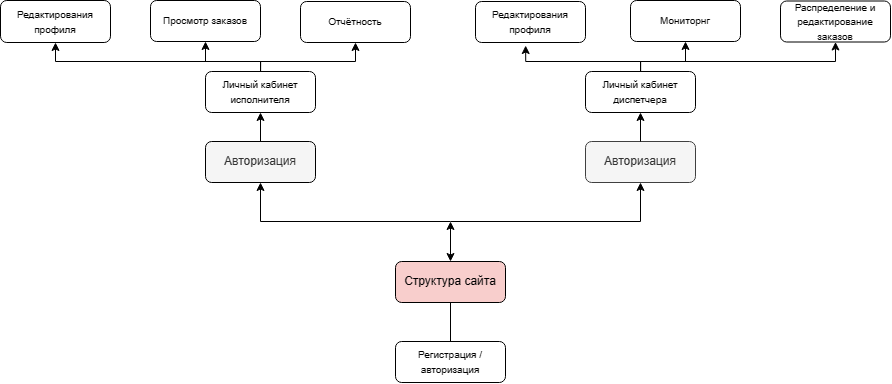


Рисунок Г.1 – Структура сайта

**Приложение Д**

Диаграмма Ганта

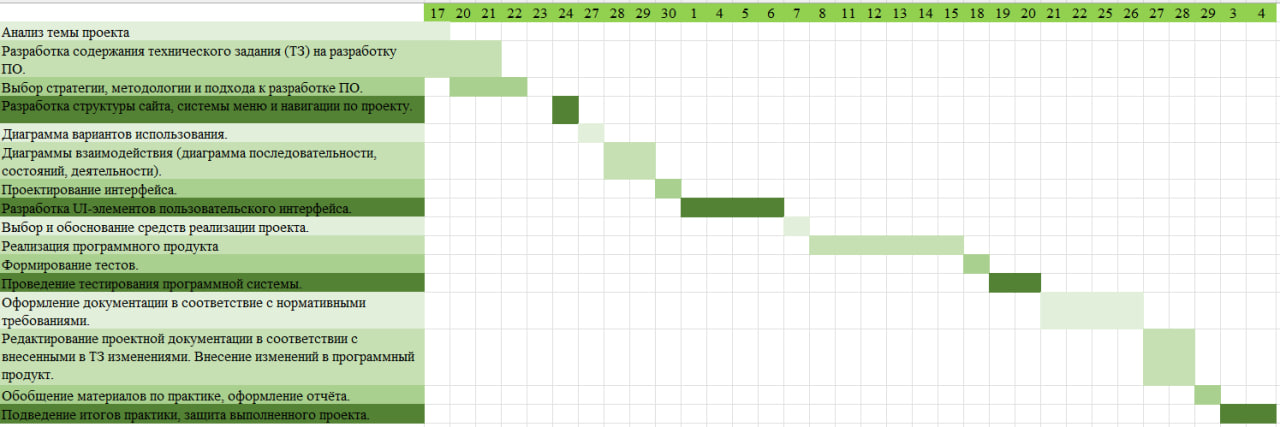


Рисунок Д.1 – Диаграмма Ганта

**Приложение Е**

Тесты на использование

Таблица Е.1 – Тесты на использование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название теста | Действие | Исходная информация | Ожидаемый результат | Фактический результат | Статус |
| Просмотр страниц сайта | 1 Нажать на каждый пункт меню |  | Открытие выбранного пункта меню сайта | Открытие выбранной страницы сайта | Выполнено успешно |
| Отображение шрифтов на сайте | 1 Проверить отображение шрифты на каждой странице сайта |  | Шрифты присуствуют на страницах | Шрифты присуствуют на страницах | Выполнено успешно |
| Возвращение на главную страницу | Нажать на логотип в шапке |  | Возвращение на главную страницу | Возвращениена главную страницу | Выполнено успешно |

**Приложение Ж**

UX/UI - дизайн

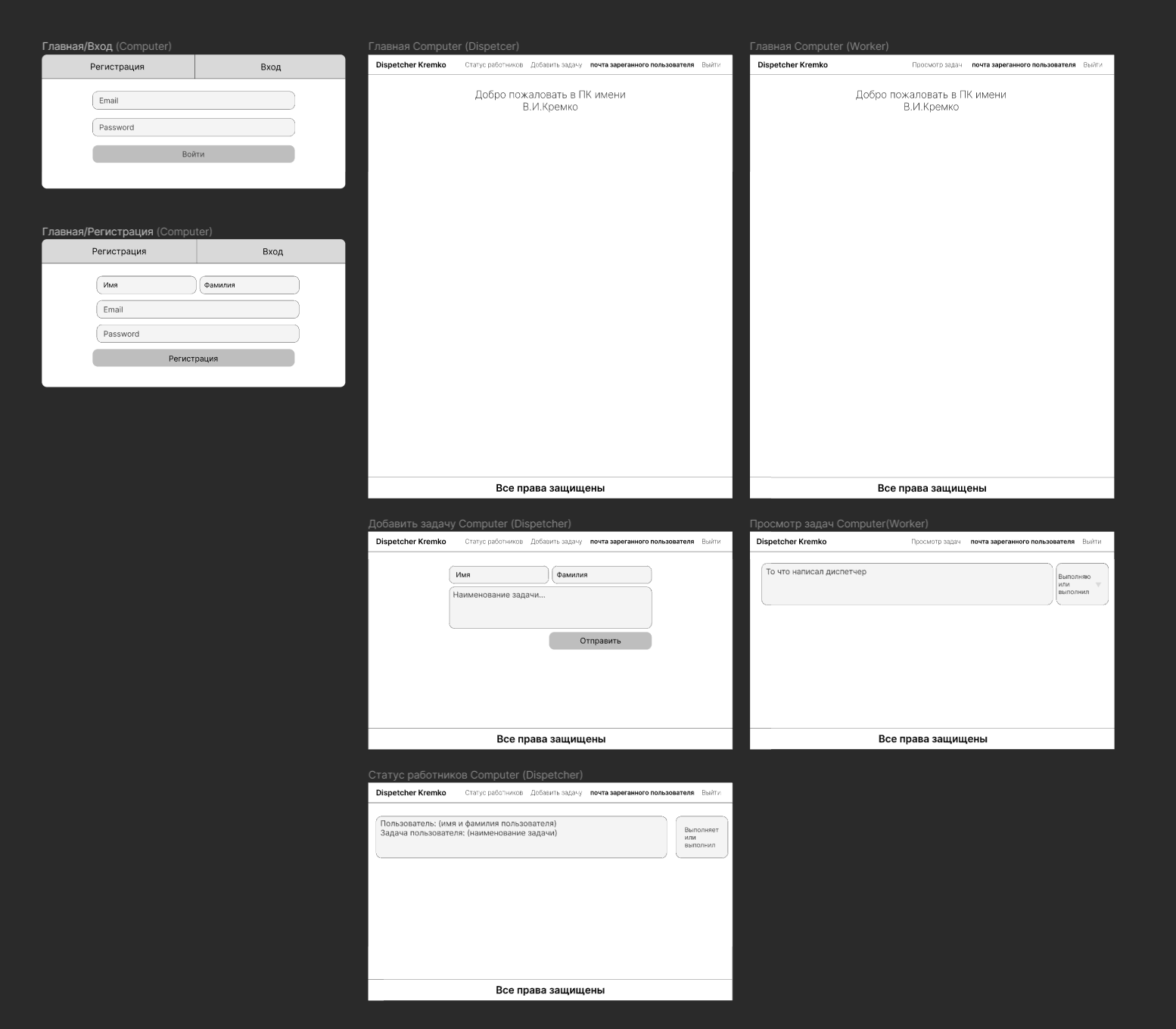
****

Рисунок Ж.1 – UI / UX макеты страниц сайта компьюторной версии

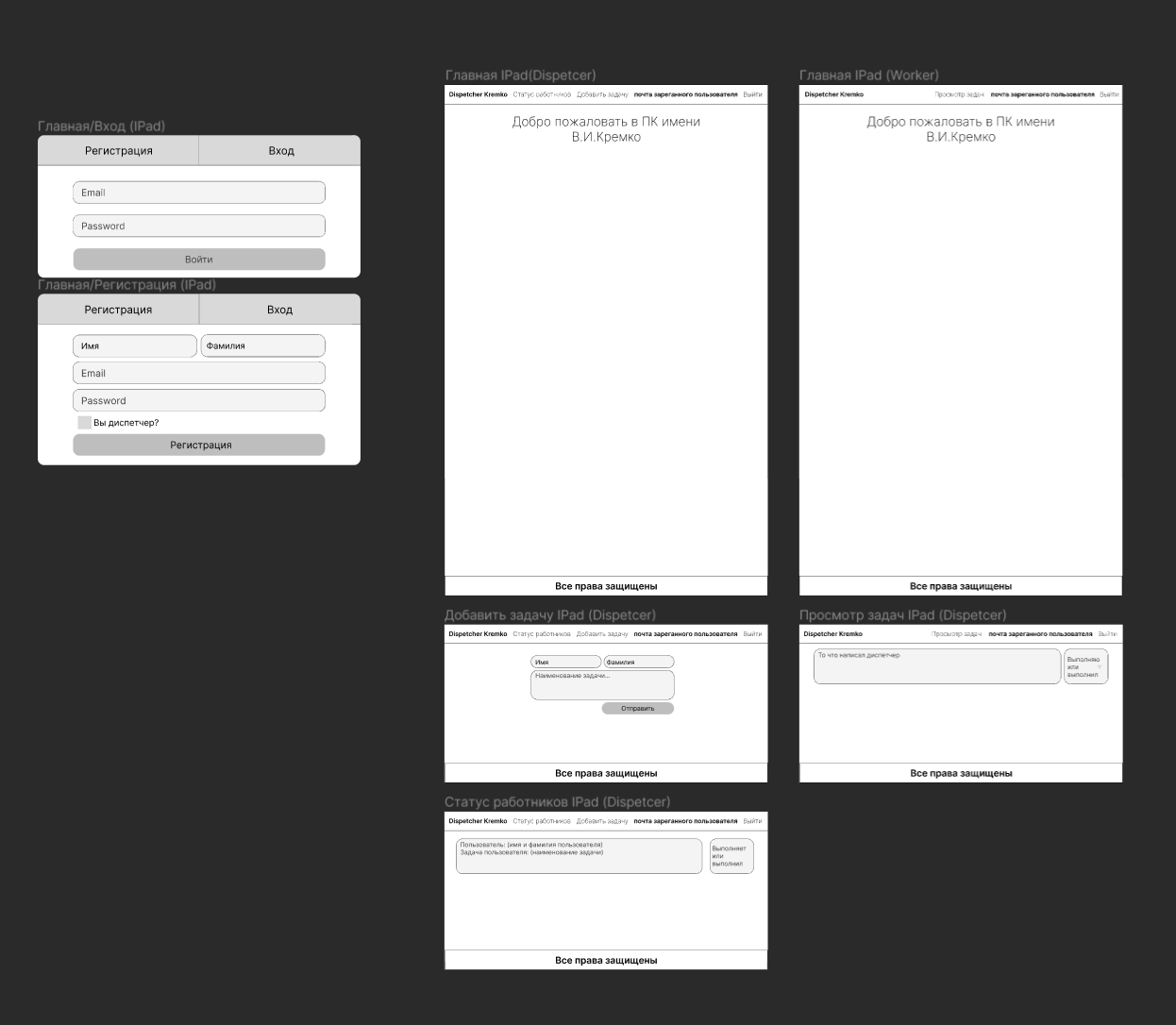


Рисунок Ж.2 – UI / UX макеты страниц сайта планшетной версии



Рисунок Ж.3 – UI / UX макеты страниц сайта мобильной версии