Титульник

Задание

ВВЕДЕНИЕ

Современный мир находится в постоянном состоянии развития, и интернет стал неотъемлемой частью повседневной жизни, ломая и переписывая старые способы взаимодействия и обмена информацией. С развитием интернет-технологий стали популярными информационные системы, помогающие обычным пользователям находить нужную информацию, а в месте с ними и веб-приложения, обеспечивающие удобный и мгновенный доступ к различным сервисам.

Информационная система — это организованная совокупность взаимосвязанных элементов, которые собирают, обрабатывают, хранят и предоставляют информацию для достижения определенных целей.

Веб-приложение становится информационной системой, обеспечивая централизованный доступ, сбор и обработку данных, надежное хранение, использование систем управления базами данных, аналитику, коммуникацию и безопасность.

Современные веб-приложения становятся неотъемлемой частью обеспечения социальных потребностей. Возрастающая популярность веб-приложений свидетельствует о том, что люди все больше ориентируются на цифровые решения для удовлетворения своих потребностей и взаимодействия с обществом. Например, социальные сети, как ВКонтакте, где люди общаются. Или онлайн-торговые платформы, такие как Авито, где можно делать покупки. Есть и облачные хранилища, например, Google Drive, для хранения файлов. Мессенджеры, такие как WhatsApp, обеспечивают коммуникацию. И это только некоторые примеры.

Важный аспект разработки и использования веб-приложений является обеспечение конфиденциальности. С увеличением объема информации, передаваемой и хранимой в онлайн-среде, обеспечение безопасности становится необходимостью. Это особенно актуально, когда в игре личные данные пользователей и информация о потребностях организаций.

Как раз одним из ключевых аспектов разработки веб-приложения для волонтерской компании "ММОО Весь Мир Един" является обеспечение высокого уровня безопасности хранимой информации. Учитывая чувствительность данных, включая личные сведения пользователей, а также любые передаваемые важные данные пользователей, особое внимание уделяется мерам по защите конфиденциальности и предотвращению несанкционированного доступа.

**Предметом** дипломной работы является разработка веб-приложения для координации и оказания помощи для волонтерской компании "ММОО Весь Мир Един" при помощи языков программирования Python и JavaScript, языка разметки HTML и языка стилей CSS.

**Объектом** выступает молодежная межрегиональная общественная организация всестороннего развития молодого поколения, сокращенно ММОО "Весь Мир Един".

**Цель** данной дипломной работы заключается в создании информационной системы **«Весь Добрый Мир»** в виде веб-приложения, направленного на координацию волонтеров и рекламу волонтерской организации. Проект направлен на эффективное взаимодействие между теми, кто готов оказывать помощь, и теми, кто нуждается в этой помощи с полностью автономной работой с минимальным вмешательством персонала организации.

В соответствии с поставленной целью в работе определены следующие **задачи**:

* разобрать понятие информационной системы и веб-приложения;
* выполнить обзор современных методов и инструментов для разработки веб-приложения;
* проанализировать предметную область и целевую аудиторию;
* выполнить проектирование дизайна веб-приложения;
* разработать веб-приложение с применением современных программных средств разработки;
* произвести отладку и тестирование функций готового веб-приложения;
* провести автоматическую интернет-рассылку организациям в цели привлечения клиентов на сайт;
* рассчитать экономическую эффективность от внедрения веб-приложения «Весь Добрый Мир» для волонтерской организации “ММОО Весь Мир Един”;

**Структура** работы состоит из введения, двух глав, расчета экономической эффективности, заключения, списка литературы и приложения.

В работе использованы методы исследования: анализ специальной литературы, обзор и анализ существующих разработок (информационных систем для автономномного распределения работы волонтёров).

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ ПОДДЕЖКИ ВОЛОНТЕРСКОГО ДВИЖЕНИЯ ДЛЯ ММОО «ВЕСЬ МИР ЕДИН»

1.1 Понятие информационной системы и веб-приложения

Информационная система — это организованное совокупность взаимосвязанных элементов, собирающих, обрабатывающих, хранящих и предоставляющих информацию для достижения определенных целей. Информационные системы включают в себя аппаратное и программное обеспечение, базы данных, сети связи и человеческие ресурсы, которые взаимодействуют для выполнения задач по управлению информацией.

Важным аспектом информационных систем является их способность интегрировать различные источники данных и обеспечивать доступ к ним в реальном времени, что позволяет улучшать эффективность и точность бизнес-процессов.

Веб-приложение — это программное обеспечение, доступное через веб-браузер, позволяющее пользователям взаимодействовать с данными и выполнять определенные функции.

В отличие от традиционных настольных приложений, веб-приложения не требуют установки на пользовательских устройствах и доступны с любого устройства, имеющего доступ к сети. Это обеспечивает гибкость и удобство использования, что делает веб-приложения востребованными в различных сферах деятельности, таких как электронная коммерция, онлайн-образование, финансовые услуги, волонтерство и другие.

Веб-приложения состоят из клиентской и серверной частей. Клиентская часть, как правило, написана на языках программирования, таких как HTML, CSS и JavaScript, и отвечает за интерфейс и взаимодействие с пользователем. Серверная часть обрабатывает запросы пользователя, выполняет бизнес-логику, управляет базами данных и генерирует ответы, которые отправляются обратно на клиентскую сторону.

Веб-приложение может считаться информационной системой при выполнении ряда ключевых условий:

* Оно должно эффективно собирать данные, используя формы ввода и интеграцию с другими системами.
* Веб-приложение должно обеспечивать надежное хранение данных через базы данных или облачные хранилища.
* Веб-приложение должно выполнять обработку данных, включая сортировку, фильтрацию и анализ.
* Веб-приложение должно распространять информацию, что предполагает предоставление пользователям удобных инструментов для доступа к данным, таких как информационные панели и отчеты.

Таким образом, выполняя эти функции, веб-приложение может быть полноценной информационной системой.

1.2 Анализ методов и инструментов для создания веб-приложения

Инструменты дизайна, языки программирования, фреймворки и библиотеки, языки стилей и разметки, облачные платформы и хостинг, а также системы управления версиями – это основные компоненты, используемые для разработки современных информационных систем и веб-приложений. Каждый из этих элементов играет свою уникальную роль в создании эффективного, функционального и удобного для пользователя продукта.

Инструменты дизайна – это программные средства, используемые для создания графического интерфейса, макетов и визуальных элементов веб-приложений. К ним относятся Adobe Photoshop, Sketch, Figma и Adobe XD. Эти инструменты помогают дизайнерам разрабатывать привлекательные и удобные интерфейсы, что повышает пользовательский опыт.

Языки программирования – это формальные языки, предназначенные для написания программ, которые выполняют определенные задачи. Примеры популярных языков программирования включают Python, Java, JavaScript, C#, Ruby и PHP. Каждый язык имеет свои особенности и области применения.

Фреймворки и библиотеки – это наборы готового кода и инструментов, облегчающие разработку программного обеспечения. Фреймворки, такие как Django и Flask для Python, Spring для Java и Rails для Ruby, предоставляют структуры и шаблоны для ускорения разработки. Библиотеки, такие как React, Vue.js и jQuery для JavaScript, предлагают готовые компоненты и функции для реализации специфичных задач.

Языки стилей и разметки – это языки, используемые для описания внешнего вида и структуры веб-страниц. HTML (HyperText Markup Language) является основным языком разметки для создания веб-страниц, а CSS (Cascading Style Sheets) используется для их стилизации, задавая оформление и расположение элементов на странице.

Облачные платформы и хостинг – это услуги, предоставляющие инфраструктуру для размещения, управления и масштабирования веб-приложений. Примеры включают Amazon Web Services (AWS), Google Cloud Platform (GCP) и Microsoft Azure. Эти платформы предлагают разнообразные сервисы, от виртуальных серверов до баз данных и аналитических инструментов.

Системы управления версиями – это инструменты для отслеживания изменений в коде и совместной работы над проектами. Git является одной из самых популярных систем управления версиями, а платформы для хостинга репозиториев, такие как GitHub, GitLab и Bitbucket, предоставляют дополнительные возможности для управления проектами, включая инструменты для код-ревью, интеграции и деплоя.

Комбинирование всех этих инструментов и технологий позволяет создать качественное веб-приложение. Дизайн инструменты обеспечивают привлекательный и интуитивно понятный интерфейс, языки программирования и фреймворки ускоряют процесс разработки и упрощают решение сложных задач, языки стилей и разметки формируют внешний вид и структуру приложения, облачные платформы и хостинг обеспечивают надежное и масштабируемое развертывание, а системы управления версиями помогают отслеживать изменения и координировать работу команды.

Для разработки веб-приложения "Весь Добрый Мир" для волонтеров были выбраны Python, Flask, GitHub Desktop, HTML и CSS по нескольким причинам.

GitHub Desktop упрощает процесс управления версиями и совместной работы, предоставляя интуитивно понятный интерфейс для взаимодействия с репозиториями GitHub.

1.3 Языки программирования Backend

Веб-приложение состоит из клиентской (frontend) и серверной (backend) частей. Frontend включает в себя пользовательский интерфейс, отображаемый в браузере, и отвечает за взаимодействие с пользователем, визуализацию данных и отправку запросов на сервер. Вместе, frontend и backend, обеспечивает полноценное функционирование веб-приложения, обеспечивая взаимодействие с данными, бизнес-логику и обеспечивая удовлетворение потребностей пользователей.

Backend (или серверная часть) в разработке веб-приложений представляет собой ту часть приложения, которая отвечает за обработку данных, бизнес-логику и взаимодействие с базой данных. Backend обрабатывает запросы, взаимодействует с базой данных, выполняет бизнес-логику и обеспечивает функциональность приложения.

Python известен своей простотой и читабельностью, что ускоряет процесс разработки и облегчает поддержку кода. Flask, как микро-фреймворк для Python, предоставляет гибкость и минимализм, необходимые для создания легковесных и эффективных веб-приложений.

Преимущества Python:

* Читаемость кода
* Множество библиотек и фреймворков
* Кросс-платформенность
* Большое сообщество и огромная документация

Недостатки Python:

* Производительность: в сравнении с некоторыми компилируемыми языками, Python может быть менее эффективен в выполнении сложных вычислительно интенсивных задач.

Flask - легковесный веб-фреймворк на языке Python, который предоставляет платформу для создания веб-приложений. Его основные принципы - простота использования, минимализм и открытость для выбора инструментов. Flask обеспечивает эффективное взаимодействие с базой данных, удобную маршрутизацию запросов и интеграцию с другими технологиями. Для полноценной разработки это требует от разработчика только знания языка Python и знаний документации работы с фреймворком Flask

Преимущества:

* Простота использования
* Гибкость
* Расширяемость
* Огромное сообщество и документация

Недостатки:

* Меньше функциональности по умолчанию для разработки сложных приложений, требуется использование других фреймворков
* Масштабируемость: Flask менее масштабируем по сравнению с более крупными фреймворками, такими как Django.

1.4 Система управления базами данных

СУБД – это программное обеспечение, предназначенное для создания, управления и обслуживания баз данных. Они обеспечивают удобный интерфейс для хранения, организации и извлечения данных, что делает их ключевым элементом в современных информационных технологиях.

Плюсы:

* Эффективное хранение данных: СУБД позволяют организовать данные таким образом, чтобы они были легко доступны, структурированы и эффективно хранились.
* Безопасность: Многие СУБД предоставляют средства для управления доступом к данным, шифрования и аудита, обеспечивая высокий уровень безопасности.
* Многопользовательский доступ: СУБД обеспечивают одновременный доступ нескольких пользователей к данным, предотвращая конфликты и обеспечивая целостность информации.
* Язык запросов: Использование языков запросов, таких как SQL, облегчает выполнение запросов и получение нужной информации.
* Масштабируемость: СУБД могут масштабироваться от небольших локальных баз данных до крупных корпоративных систем.

Минусы:

* Сложность: Настройка и управление СУБД может потребовать навыков и времени, особенно при работе с распределенными системами.
* Зависимость от производительности: Некорректная настройка или плохой дизайн базы данных может привести к снижению производительности.
* Затраты на обслуживание: Крупные СУБД могут требовать значительных ресурсов для обслуживания, что включает в себя регулярные резервные копии, обновления и мониторинг.

Одним из преимуществ использования СУБД является обеспечение многопользовательского доступа к данным. Веб-приложения часто обслуживают множество пользователей, и СУБД предоставляют механизмы синхронизации и контроля доступа, гарантируя безопасное одновременное взаимодействие с информацией.

Безопасность данных также является важным аспектом, обеспечиваемым СУБД. Системы управления базами данных предоставляют средства управления доступом, шифрования и аудита, что существенно влияет на защиту конфиденциальности и соблюдение стандартов безопасности.

Масштабируемость – СУБД эффективно адаптируется к различным объемам данных и нагрузкам. Наконец, управление изменениями в структуре базы данных, обеспечиваемое СУБД, упрощает развитие и сопровождение веб-приложений.

* MySQL Server — это реляционная система управления базами данных (RDBMS), разработанная компанией Oracle. MySQL Server предоставляет эффективное хранение, управление и обработку данных, а также обеспечивает средства для разработки сложных информационных систем.

Плюсы:

* Масштабируемость: MySQL Server предлагает возможности масштабирования от маленьких баз данных до крупных корпоративных систем, что делает его подходящим для различных проектов.
* Безопасность: Система обеспечивает широкий спектр средств безопасности, включая управление доступом, шифрование данных и аудит, что позволяет надежно защищать данные.
* Мощные инструменты управления: MySQL Server поставляется с разнообразными инструментами управления и мониторинга, такими как MySQL Workbench, что облегчает администрирование баз данных.
* Открытый исходный код: MySQL Server является продуктом с открытым исходным кодом, что делает его бесплатным для использования и позволяет сообществу разработчиков вносить улучшения и дополнения.
* Поддержка нескольких платформ: MySQL Server поддерживает различные операционные системы, включая Windows, Linux и macOS, что обеспечивает гибкость в выборе среды разработки и развертывания.

Минусы:

* Проблемы с производительностью при высоких нагрузках: В некоторых случаях, особенно при очень высоких нагрузках, MySQL Server может требовать дополнительной оптимизации и настройки для поддержания производительности.
* Ограниченная функциональность: Несмотря на свои возможности, MySQL Server может уступать по функциональности и гибкости некоторым коммерческим системам управления базами данных, таким как MS SQL Server или Oracle Database.
* Поддержка и обновления: Хотя сообщество разработчиков активно поддерживает MySQL, предприятиям может потребоваться платная поддержка от Oracle для получения регулярных обновлений и технической помощи.

1.5 Языки разметки и таблицы стилей

HTML и CSS являются стандартными инструментами для создания и стилизации веб-страниц, обеспечивая широкую совместимость и доступность.

Плюсы:

* Семантика: HTML 5 внедрил новые семантические теги, такие как header, footer, nav, что делает код более читаемым и понятным.
* Мультимедиа: поддерживает встроенные теги для вставки аудио и видео файлов, такие как audio и video, что облегчает интеграцию мультимедийного контента.
* Локальное хранилище: Введение localStorage позволяет веб-приложениям хранить данные локально на устройстве пользователя.

Минусы:

* Сложность валидации: HTML 5 предоставляет большое количество новых элементов и атрибутов, что может сделать валидацию сложной.
* Браузерная поддержка: Некоторые устаревшие браузеры могут не полностью поддерживать все новые возможности HTML 5.

1.6 Системы управления версиями

GitHub — это платформа для хостинга репозиториев, использующая систему контроля версий Git. GitHub предоставляет разработчикам возможность управлять своими проектами, отслеживать изменения в коде, совместно работать над проектами и вести документацию.

Плюсы:

* Совместная работа: GitHub облегчает совместную работу разработчиков над проектами, предлагая инструменты для обзора кода, обсуждений и интеграции с другими сервисами.
* Контроль версий: Система Git позволяет отслеживать все изменения в коде, легко откатываться к предыдущим версиям и вести параллельную работу над различными ветками проекта.
* Интеграции и автоматизация: GitHub поддерживает интеграции с множеством других инструментов и сервисов, таких как CI/CD, что упрощает процесс автоматизации тестирования и развертывания приложений.
* Документация и вики: Платформа предоставляет средства для ведения документации и создания вики-страниц, что помогает в организации и управлении проектом.

Минусы:

* Ограничения бесплатного плана: Бесплатный план GitHub имеет определенные ограничения, такие как ограниченное количество приватных репозиториев и ограничение на использование некоторых функций.
* Коммерческая поддержка: Для получения расширенных возможностей и технической поддержки предприятиям может потребоваться подписка на платные планы.

GitHub Desktop — это десктопное приложение, предоставляющее удобный графический интерфейс для работы с репозиториями на GitHub. Оно предназначено для упрощения процессов клонирования, создания, коммитов и синхронизации репозиториев.

Плюсы:

* Удобство использования: GitHub Desktop предоставляет интуитивно понятный интерфейс, который упрощает работу с Git, даже для пользователей с ограниченным опытом в командной строке.
* Визуализация изменений: Приложение визуализирует изменения в коде, что облегчает понимание и анализ коммитов и веток.
* Интеграция с GitHub: Тесная интеграция с платформой GitHub позволяет легко синхронизировать локальные изменения с удаленными репозиториями и управлять запросами на слияние (pull requests).

Минусы:

* Ограниченная функциональность: В сравнении с командной строкой, GitHub Desktop может не поддерживать некоторые продвинутые функции Git, что может ограничивать опытных пользователей.
* Зависимость от GitHub: Приложение разработано специально для работы с репозиториями на GitHub, что может быть недостатком для пользователей, предпочитающих другие платформы для хостинга репозиториев.

Таким образом, GitHub и GitHub Desktop являются мощными инструментами для управления кодом и совместной работы над проектами, обеспечивая разработчикам гибкость, удобство и богатый набор функций для эффективной разработки программного обеспечения.

# ПЛАНИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕЖКИ ВОЛОНТЕРСКОГО ДВИЖЕНИЯ ДЛЯ ММОО «ВЕСЬ МИР ЕДИН»

2.1 Описание целевой аудитории

Молодежная межрегиональная общественная организация всестороннего развития молодого поколения "Весь Мир Един" оказывает помощь всем нуждающимся на территории Москвы и Московской области.

Миссия организации основана на принципах добровольчества и солидарности, вовлекая людей, готовых помочь тем, кто оказался в сложной ситуации. Одним из ключевых направлений работы является помощь детским домам. Регулярно предоставляется поддержка в виде финансовых средств, предметов первой необходимости, игрушек, книг и других полезных вещей, создавая для детей атмосферу заботы и радости.

Центральную роль в деятельности организации играют волонтеры-студенты или бывшие, выпустившиеся студенты. Они активно участвуют в организации и проведении мероприятий, уделяя внимание и заботу детям из детских домов. Усилия волонтеров создают позитивную атмосферу и вдохновляют других на участие и помощь. Организация предоставляет студентам возможность пройти производственную практику, что помогает им приобрести ценный опыт в IT-отрасли и развить социальные навыки. Практика способствует развитию лидерских качеств и открывает новые горизонты для студентов, предлагая перспективы для их личностного и профессионального роста.

Информационная система "Весь Добрый Мир" разработана для поддержки и оптимизации работы волонтерской организации ММОО "Весь Мир Един".

Информационная система для управления деятельностью Местной Молодежной Общественной Организации «Весь Мир Един» предназначена для поддержки и оптимизации работы волонтерской организации, направленной на объединение молодых людей и реализацию различных социальных, культурных и экологических проектов. Целевая аудитория данной системы является многослойной и включает в себя различные группы пользователей, каждая из которых имеет свои уникальные потребности и цели.

Обычные пользователи

Обычные пользователи — это люди, которые могут просматривать информацию на сайте и следить за деятельностью организации без необходимости регистрации. Однако у них есть возможность зарегистрироваться и присоединиться к одной из организаций в будущем. Для этой группы пользователей система предоставляет доступ к основным новостям и информации о проектах, что позволяет им быть в курсе событий и, при желании, стать активными участниками волонтерской деятельности.

Зарегистрированные пользователи

Зарегистрированные пользователи имеют больше возможностей по сравнению с обычными пользователями. Они могут:

* Размещать запросы о помощи: Создавать новые запросы на помощь для различных проектов.
* Откликаться на чужие запросы: Принимать участие в проектах, созданных другими пользователями или организациями.
* Создавать организации: Основывать новые организации, которые будут размещать свои собственные запросы о помощи.

Система предоставляет зарегистрированным пользователям удобные инструменты для управления своими проектами и взаимодействия с другими участниками.

Организации

Организации, зарегистрированные в системе, могут действовать от лица группы людей и выполнять следующие действия:

* Размещать запросы о помощи: Создавать и публиковать запросы на помощь для своих проектов.
* Откликаться на чужие запросы: Участвовать в проектах, созданных другими организациями или пользователями.

Система предоставляет организациям инструменты для координации работы, управления ресурсами и взаимодействия с волонтерами.

Администрация

Администрация системы отвечает за обеспечение ее корректной работы и поддержание порядка. Основные функции администрации включают:

* Просмотр отклоненных системой запросов о помощи: Проверка и анализ запросов, отклоненных автоматически системой.
* Обработка жалоб: Рассмотрение жалоб пользователей и принятие необходимых мер для их решения.

Система предоставляет администрации необходимые инструменты для мониторинга, управления пользователями и обеспечения безопасности данных.

Целевая аудитория информационной системы "Весь Добрый Мир" включает в себя несколько ключевых групп пользователей. Каждая из этих групп имеет свои уникальные потребности и цели, которые учитываются при разработке и использовании системы. Обычные пользователи могут следить за деятельностью организации, зарегистрированные пользователи и организации получают инструменты для активного участия в проектах, а администрация обеспечивает бесперебойную работу системы и безопасность данных.

Такой подход позволяет создать эффективную и многофункциональную систему, способную удовлетворить потребности всех участников процесса и способствовать успешной реализации миссии организации.

2.2 Принципы информационной системы

Информационная система для ММОО «Весь Мир Един» должна учитывать следующие ключевые принципы: интуитивную навигацию, прозрачность и доступность информации, вовлеченность пользователей, масштабируемость и адаптивность, а также безопасность и конфиденциальность.

Интуитивная навигация позволяет пользователям легко находить нужную информацию благодаря логической иерархии и понятным элементам управления. Основные компоненты, обеспечивающие интуитивную навигацию:

* Меню: четко структурированное меню с основными разделами сайта.
* Навигация: помогает пользователям отслеживать свое местоположение на сайте.
* Иконки и кнопки: понятные иконки и кнопки, облегчающие взаимодействие с системой.
* Прозрачность и доступность информации

Масштабируемость и адаптивность системы обеспечиваются корректной работой на различных устройствах и готовностью к расширению функционала. Основные методы:

* Адаптивный дизайн: использование медиа запросов CSS для корректного отображения на любых устройствах.
* Гибкая архитектура: возможность добавления новых функций без значительных изменений системы.

Безопасность и конфиденциальность гарантируются защитой данных пользователей и безопасным хранением информации благодаря SSL шифрованию данных для защиты от несанкционированного доступа.

2.3 Проектирование структуры и дизайна

* 1. Проектирование структуры

Проектирование структуры и дизайна информационной системы для ММОО «Весь Мир Един» представляет собой важный этап, направленный на создание интуитивного и функционального веб-приложения, которое будет эффективно поддерживать деятельность организации и удовлетворять потребности ее целевой аудитории.

Анализ требований

Первым шагом в проектировании структуры и дизайна является сбор и систематизация всех требований к будущей информационной системе. Это включает в себя:

* Понимание целей и задач организации.
* Анализ потребностей различных групп пользователей.
* Определение функциональных требований к системе.

На основе этих данных формируется концептуальная модель системы, которая служит основой для дальнейшей разработки.

Информационная архитектура

Информационная архитектура включает в себя организацию и структуру контента, а также определение навигационных элементов, обеспечивающих удобный доступ к информации. Основные компоненты информационной архитектуры:

* Главная страница (index.html): Содержит общую информацию об организации, ее миссии и основных направлениях деятельности.
* Страница создания запроса о помощи (create-event.html): Форма для волонтеров и организаций, позволяющая создавать новые запросы о помощи.
* API страница для подтверждения почты (email\_confirmation.html): Страница для подтверждения адреса электронной почты пользователя.
* Страница с конкретным запросом помощи (event.html): Отображает подробную информацию о конкретном запросе на помощь.
* Форма входа (login.html): Страница для аутентификации пользователей.
* Меню ошибки (oops-error.html): Страница, отображаемая при возникновении ошибок.
* Страница конфиденциальности (privacy.html): Содержит политику конфиденциальности организации.
* Настройка профиля (profile-edit.html): Страница для редактирования профиля пользователя.
* Профиль конкретного пользователя (profile.html): Отображает информацию о пользователе.
* Форма регистрации (registration.html): Страница для регистрации новых пользователей.
* Меню всех запросов о помощи (requests.html): Отображает все текущие запросы о помощи с ежедневным обновлением данных.
* Форма восстановления пароля (reset-password.html): Страница для восстановления пароля.
* Меню с результатами поиска (results.html): Отображает результаты поиска запросов о помощи.
* Правила (rules.html): Содержит правила использования системы.
* Условия использования (tou.html): Содержит условия использования сайта.

Основные компоненты информационной архитектуры:

* Главная страница: Обеспечивает общее представление о сайте и содержит новости, основные направления деятельности организации и контактную информацию.
* Разделы для волонтеров: Включают информацию о текущих и будущих проектах, возможности регистрации на мероприятия и отслеживания своего вклада.
* Разделы для координаторов проектов: Обеспечивают инструменты для планирования и управления проектами, распределения задач и мониторинга прогресса.
* Раздел для администраторов: Включает аналитические отчеты и статистику для оценки эффективности деятельности организации.
* Разделы для партнеров и спонсоров: Предоставляют информацию о результатах проектов и возможностях взаимодействия.
* Разделы для общественности и СМИ: Содержат новости, отчеты и информацию о предстоящих мероприятиях.

Проектирование дизайна

Проектирование дизайна включает в себя создание визуально привлекательного и интуитивно понятного интерфейса, который обеспечивает удобство использования для всех групп пользователей. Основные принципы дизайна:

* Юзабилити: Удобство использования и навигации по сайту.
* Модульность: Разделение контента на логические блоки.
* Адаптивность: Обеспечение корректного отображения на различных устройствах.
* Доступность: Учет потребностей пользователей с ограниченными возможностями.

Проектирование структуры и дизайна информационной системы "Весь Добрый Мир" направлено на создание многофункционального и удобного веб-приложения, способного удовлетворить потребности всех участников процесса. Такой подход позволяет обеспечить успешную реализацию миссии организации и эффективное взаимодействие между всеми группами пользователей.

Визуальный дизайн

Визуальный дизайн системы должен быть привлекательным и отражать миссию и ценности организации. Основные принципы визуального дизайна включают:

* Эстетика: Дизайн должен быть визуально привлекательным и соответствовать бренду.
* Консистентность: Все элементы дизайна должны быть согласованы между собой.
* Простота: Минимализм и ясность в представлении информации.
* Эффективность: Дизайн должен поддерживать основные функции системы и способствовать достижению целей пользователей.

Прототипирование

Прототипирование является важным этапом в процессе проектирования, позволяющим тестировать и улучшать дизайн до его финальной реализации. Основные этапы прототипирования включают:

* Создание черновых эскизов: Ручное рисование идей и концепций.
* Разработка цифровых прототипов: Использование специализированных программ для создания интерактивных макетов.
* Тестирование прототипов: Проведение пользовательских тестов для выявления и устранения проблем.

Разработка пользовательского интерфейса

Разработка пользовательского интерфейса (UI) включает создание визуальных элементов и взаимодействий, которые обеспечивают удобство и эффективность использования системы. Основные технологии, используемые для разработки UI:

* HTML и CSS: Основные технологии для создания структуры и стиля веб-страниц.
* JavaScript и библиотеки: Использование JavaScript и библиотек, таких как React или Angular, для создания интерактивных элементов.
* Инструменты дизайна: Adobe XD, Figma, Sketch для создания и тестирования макетов.

Виды макетов

В процессе проектирования используются различные виды макетов для визуализации и тестирования дизайна:

* Фиксированный (Static)

Жестко заданные размеры элементов, которые не меняются с изменением размера экрана. Простой в реализации, но плохо адаптируется к различным устройствам.

* Резиновый (Liquid)

Использует относительные единицы измерения, такие как проценты, для адаптации к различным размерам экранов. Гибкий, но может создавать проблемы с дизайном при экстремальных размерах окон.

* Адаптивный (Adaptive)

Предусматривает несколько фиксированных макетов для разных размеров экранов. Подгружается соответствующий макет в зависимости от устройства. Требует больше ресурсов для разработки и поддержки.

* Отзывчивый (Responsive)

Сочетает гибкие сетки и медиа-запросы для оптимального отображения на всех устройствах. Использует единый код, что упрощает разработку и поддержку.

* Мобильный первый (Mobile First)

Разработка дизайна и функционала сначала для мобильных устройств, затем для десктопов и планшетов. Улучшает производительность и пользовательский опыт на мобильных устройствах. Этот системный подход к проектированию позволяет создавать комплексные и удобные для пользователей решения, отражающие миссию и ценности организации, обеспечивая эффективное взаимодействие и достижение поставленных целей.

Создание макетов

Был создан шаблон проекта сайта, включающий макеты: Главной страницы, Страницы регистрации и входа пользователя, Страницы поиска запросов и Профиля пользователя. Этот подход позволяет заранее визуализировать структуру и функционал сайта, что существенно облегчает процесс разработки и улучшает конечный пользовательский опыт. Рассмотрим каждый макет подробнее.



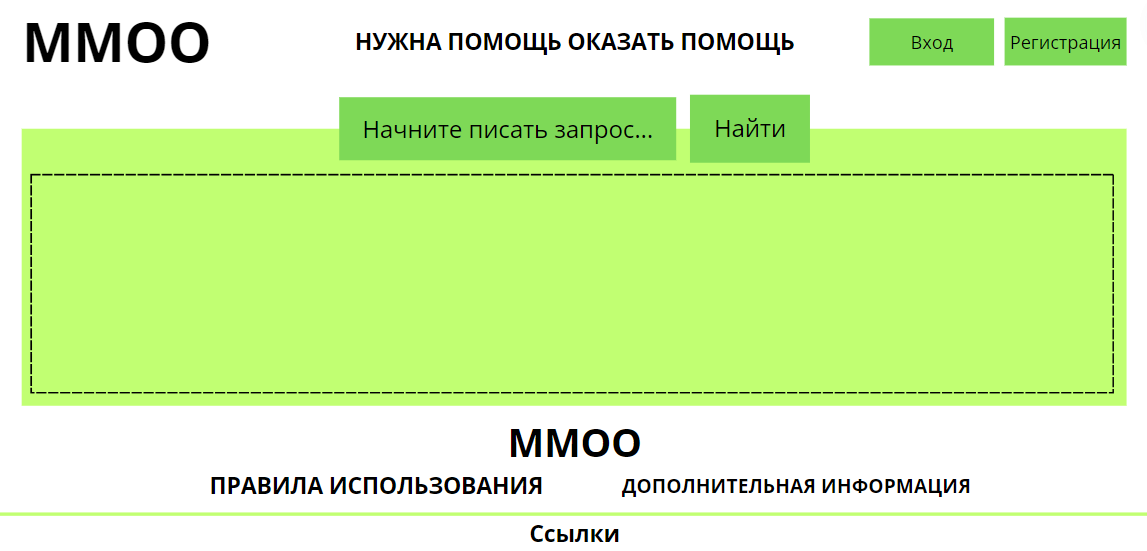
1. Главная страница

Главная страница служит первым контактным пунктом с пользователем. Она включает в себя навигационные элементы, основные разделы сайта, а также информационные блоки, привлекающие внимание посетителей и направляющие их к основным функциям ресурса.



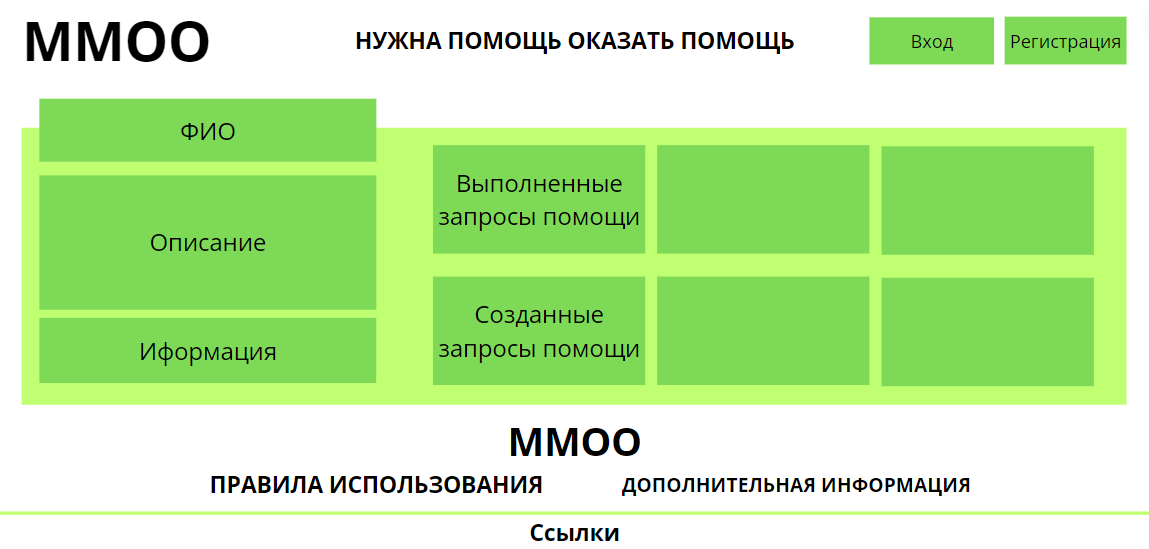
1. Страница регистрации или входа пользователя

Страница регистрации и входа предназначена для обеспечения простого и интуитивно понятного процесса создания учетной записи и авторизации пользователей. Этот макет включает формы ввода данных, сообщения об ошибках и элементы интерфейса для восстановления пароля.



1. Страница поиска запросов

Страница поиска запросов обеспечивает пользователям возможность эффективно искать и находить нужную информацию на сайте. В макет включены поисковая строка, фильтры, сортировка результатов и удобное отображение найденной информации.



1. Профиль пользователя

Профиль пользователя содержит личные данные, настройки и историю активности пользователя. Макет включает различные секции для редактирования информации, просмотра активности и управления учетной записью.

Создание наработанных макетов на ранних этапах проектирования имеет несколько ключевых преимуществ:

* Ясность и структура: Макеты позволяют визуализировать структуру и расположение элементов на каждой странице, что способствует более точному и структурированному подходу к разработке.
* Пользовательский опыт: Предварительное создание макетов помогает выявить потенциальные проблемы с юзабилити и навигацией, обеспечивая более комфортный пользовательский опыт.
* Коммуникация и согласованность: Макеты служат отличным инструментом для обсуждения и согласования деталей проекта между дизайнерами, разработчиками и заинтересованными сторонами.
* Экономия времени и ресурсов: Четкие макеты позволяют сократить время на доработки и изменения на поздних этапах разработки, что экономит ресурсы проекта.

2.4 Разработка информационной системы

для ММОО «Весь Мир Един»

Современные веб-приложения играют ключевую роль в различных сферах деятельности, предоставляя удобные и эффективные инструменты для взаимодействия пользователей с информацией и сервисами. При их разработке можно использовать множество вариантов технологий и подходов, позволяющих достичь различных целей и удовлетворить специфические требования проектов.

Для разработки информационной системы "Весь Добрый Мир" в виде веб-приложения были выбраны следующие основные технологии: HTML, CSS, JavaScript, Python, Flask и Jinja2. Эти технологии обеспечивают оптимальный вариант для создания функционального, удобного и масштабируемого веб-приложения, способного удовлетворить потребности всех участников процесса.

HTML отвечает за создание структуры веб-страниц, обеспечивая их семантическое и логичное построение. Это основной язык разметки, используемый для организации контента на веб-странице.

Основные используемые функции и принципы при разработке:

* Семантическая разметка: Использование тегов для обозначения различных типов контента (заголовки, абзацы, списки и т.д.).
* Ссылки и изображения: Встраивание гиперссылок и изображений для навигации и визуального представления информации.
* Формы: Создание интерактивных форм для ввода данных пользователями.

CSS отвечает за стилизацию и визуальное оформление веб-страниц, делая их привлекательными и удобными для восприятия. CSS позволяет отделить содержание документа от его представления, что способствует улучшению юзабилити и управляемости.

Основные используемые функции и принципы при разработке:

* Стилизация элементов: Определение внешнего вида элементов HTML, таких как цвета, шрифты, отступы и границы.
* Адаптивный дизайн: Использование медиазапросов для адаптации веб-страниц под различные устройства и экраны.
* Анимации и трансформации: Добавление визуальных эффектов для улучшения пользовательского опыта.

JavaScript добавляет динамичность и интерактивность элементам интерфейса, позволяя реализовать современные функциональные возможности веб-приложений. JavaScript выполняется на стороне клиента, что позволяет взаимодействовать с пользователем без необходимости обновления страницы.

Основные используемые функции и принципы при разработке:

* Динамическое обновление контента: Изменение и обновление содержимого веб-страниц без перезагрузки.
* События и обработчики: Реакция на действия пользователя, такие как клики, ввод данных и перемещение мыши.
* API и AJAX: Взаимодействие с сервером для получения и отправки данных асинхронно.

Python обеспечивает серверную часть приложения, благодаря своей простоте и мощным возможностям. Этот язык программирования используется для написания серверного кода, обработки данных и выполнения бизнес-логики.

Основные используемые функции и принципы при разработке:

* Простота и читаемость: Легкость написания и понимания кода, что ускоряет процесс разработки и поддержки.
* Мощные библиотеки: Наличие большого количества библиотек и фреймворков для решения различных задач.
* Универсальность: Возможность использования Python для веб-разработки, анализа данных, автоматизации и многого другого.

Flask – легкий и гибкий веб-фреймворк на Python, используемый для создания серверной логики и управления запросами. Flask позволяет быстро разрабатывать веб-приложения и API, обеспечивая необходимую функциональность и расширяемость.

Основные используемые функции и принципы при разработке:

* Минимализм: Легковесный фреймворк без лишних зависимостей, обеспечивающий гибкость и простоту.
* Расширяемость: Возможность добавления необходимых модулей и расширений для увеличения функционала.
* Простота настройки: Легкость настройки и конфигурации приложения, что ускоряет процесс разработки.

Jinja2 – это шаблонизатор для Python, который интегрируется с Flask, позволяя создавать динамически генерируемые HTML-страницы. Jinja2 используется для отделения логики от представления, что улучшает структуру и читаемость кода.

Основные используемые функции и принципы при разработке:

* Шаблоны и макеты: Создание общих шаблонов и макетов для повторного использования на различных страницах.
* Переменные и выражения: Вставка данных и выполнение логических операций внутри шаблонов.
* Циклы и условия: Управление выводом данных с помощью циклов и условий для создания динамического контента.

Выбор этих технологий обоснован их популярностью, надежностью и широкими возможностями для разработки веб-приложений. Они обеспечивают быструю и эффективную реализацию функционала, необходимого для поддержки и оптимизации работы волонтерской организации ММОО «Весь Мир Един».

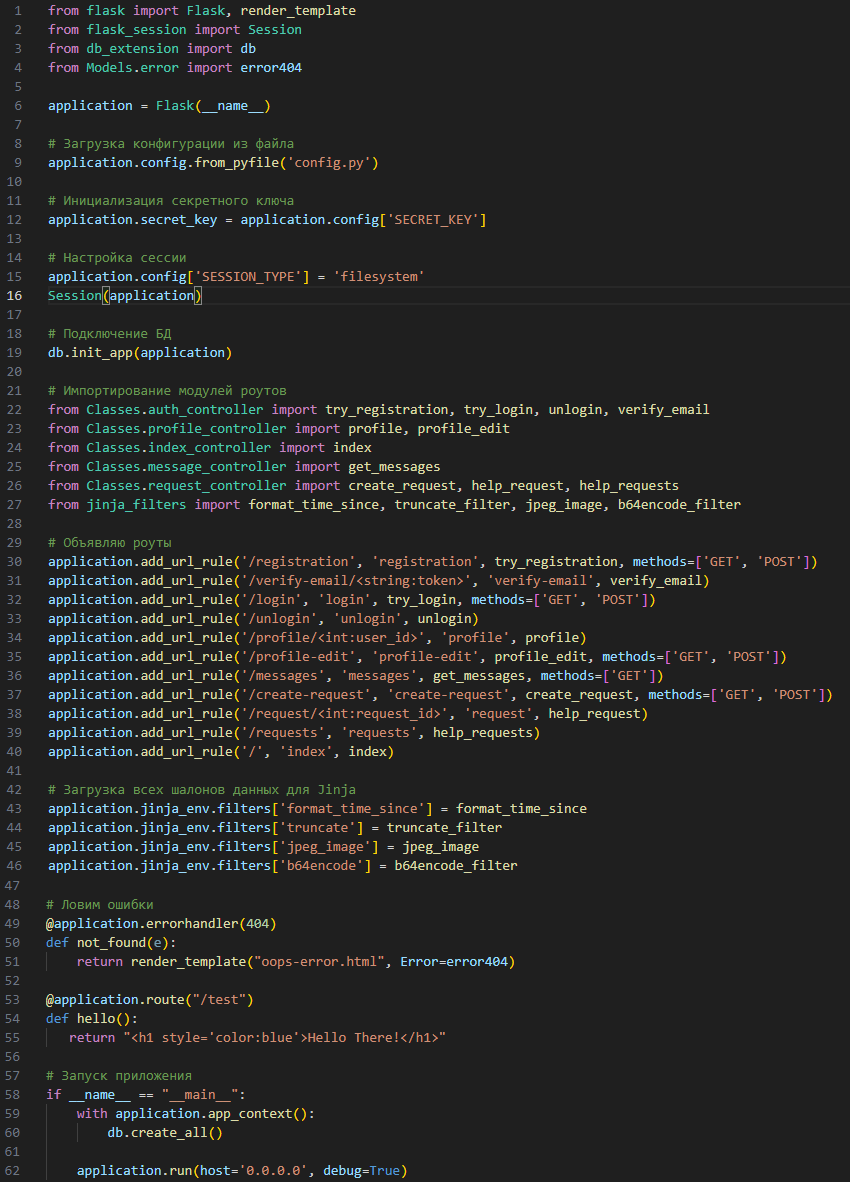
Интеграция HTML, CSS, JavaScript, Python, Flask и Jinja2 обеспечивает создание целостного и функционального веб-приложения. HTML и CSS формируют структуру и стиль страниц, JavaScript добавляет интерактивность, а Python и Flask реализуют серверную логику и управление данными. Jinja2 позволяет динамически генерировать HTML, интегрируя серверную и клиентскую части. Такое сочетание технологий обеспечивает оптимальную производительность, гибкость и удобство использования системы, удовлетворяя потребности всех участников процесса, если проект бы разрабатывался не в одиночку, и способствуя успешной, а главное быстрой реализации проекта для ММОО «Весь Мир Един».

Реализация

При разработке веб-приложений в Flask используется архитектурная модель Model-View-Controller (MVC). MVC разделяет приложение на три взаимосвязанных компонента: модель, представление и контроллер. Это обеспечивает лучшую организацию кода, упрощает разработку и поддержку приложений.

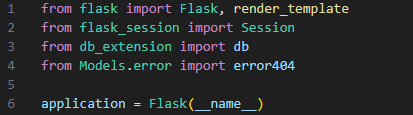
* Model (Модель): Отвечает за управление данными и бизнес-логикой приложения. Включает взаимодействие с базой данных, обработку данных и правила бизнес-логики.
* View (Представление): Отвечает за отображение данных пользователю. Включает шаблоны, которые генерируют HTML, отображающий данные, полученные из модели.
* Controller (Контроллер): Отвечает за обработку запросов пользователя и взаимодействие с моделью и представлением. Контроллеры принимают запросы, вызывают необходимые методы модели и передают данные в представление для отображения.

Файл main.py играет ключевую роль в реализации и запуске веб-приложения "Весь Добрый Мир". В этом файле происходит настройка и инициализация основных компонентов системы, таких как маршрутизация, сессии, база данных и обработка ошибок.



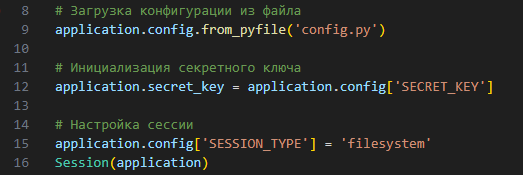
1. Файл main.py

Инициализация приложения



1. Инициализация приложения

Файл начинается с импортирования необходимых библиотек и модулей, а также создания экземпляра Flask приложения.



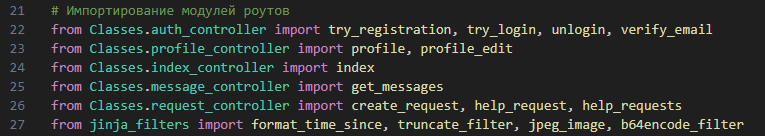
1. Загрузка конфигурации

Конфигурация приложения загружается из файла config.py, включая секретный ключ для сессий. Сессии настраиваются для хранения на файловой системе.



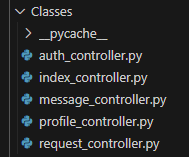
1. Подключение базы данных

База данных инициализируется и подключается к приложению, что позволяет взаимодействовать с данными через SQLAlchemy.



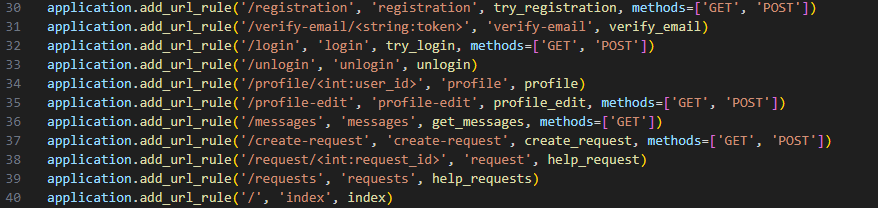
1. Импортирование модулей маршрутизации

Импортируются контроллеры для обработки различных маршрутов, включая аутентификацию, профиль, главную страницу, сообщения и запросы о помощи. Также импортируются фильтры для Jinja2, которые используются для форматирования данных в шаблонах. Контроллеры, импортируемые из различных модулей, обрабатывают регистрацию, вход в систему, профили пользователей, сообщения и запросы о помощи.



1. Список контроллеров

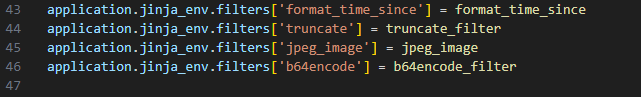
Контроллеры определяют маршруты (роуты) и связывают их с соответствующими функциями, которые обрабатывают запросы пользователей.



1. Объявлениемаршрутов с помощью контроллеров

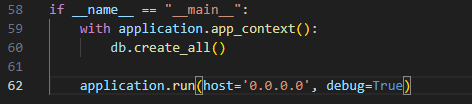
Маршруты (роуты) определяются для обработки различных запросов пользователей. Каждый маршрут связывается с соответствующим контроллером, который обрабатывает запрос и возвращает ответ.

Каждый маршрут связан с функцией, которая обрабатывает запросы и возвращает соответствующий ответ.



1. Настройка Jinja2 фильтров

Фильтры Jinja2 настраиваются для форматирования данных в шаблонах, что улучшает отображение информации на веб-страницах.



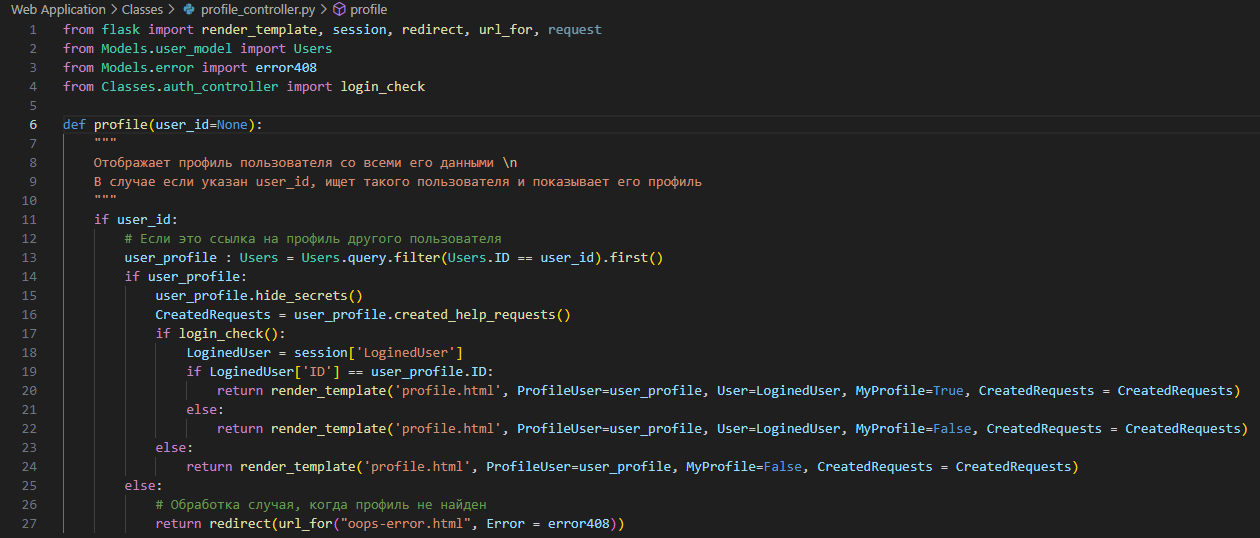
1. Запуск приложения

Приложение запускается с указанными настройками, включая создание всех таблиц в базе данных при первом запуске. Приложение работает в режиме отладки, что упрощает процесс разработки и тестирования.

В готовом виде приложения «Весь Добрый Мир» режим откладки обязательно выключается.

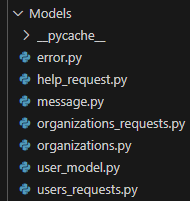
Представления (Views) отвечают за отображение данных пользователю. В Flask представления создаются с помощью шаблонов Jinja2 и функций контроллеров, которые обрабатывают запросы и возвращают готовые HTML-страницы.

В качестве примера рассмотрим контроллер profile\_controller.py, который отвечает за отображение профиля пользователя.



1. Контроллер profile\_controller.py

Контроллер импортирует необходимые модули и самое главное - модели (Models) Flask, которые нужны для работы с шаблонами (render\_template), сессиями (session), переадресациями (redirect), URL (url\_for) и запросами (request). Также импортируются модели пользователей и ошибок, а также функция проверки входа.



1. Список всех моделей

* error.py - Хранение информации об ошибках, которые могут возникать в приложении.
* help\_request.py - Управление данными о запросах на помощь, созданных пользователями.
* message.py - Хранение данных о сообщениях, отправленных пользователями друг другу.
* organizations\_requests.py - Управление данными о запросах, связанных с организациями.
* organizations.py - Хранение информации об организациях, участвующих в платформе.
* user\_model.py - Управление данными о пользователях, таких как личные данные и контактная информация.
* users\_requests.py - Управление данными о запросах, созданных пользователями или адресованных им.

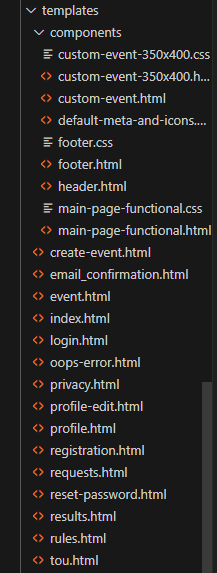
Функция profile отвечает за отображение профиля пользователя.

1. Проверка user\_id: Если передан user\_id, функция пытается найти соответствующего пользователя в базе данных.
2. Поиск пользователя: Users.query.filter(Users.ID == user\_id).first() ищет пользователя с указанным user\_id.
3. Обработка найденного профиля:
   * Скрытие секретных данных: user\_profile.hide\_secrets()
   * Получение созданных запросов: CreatedRequests = user\_profile.created\_help\_requests()
4. Проверка входа: Если пользователь вошел в систему (login\_check()), отображаются данные о текущем пользователе и профиле:
   * Если текущий пользователь смотрит свой профиль: Передаются данные для отображения своего профиля (MyProfile=True).
   * Если текущий пользователь смотрит чужой профиль: Передаются данные для отображения чужого профиля (MyProfile=False).
5. Обработка ошибки: Если профиль не найден, происходит редирект (переадресация) на страницу ошибки.

HTML страницы играют ключевую роль в предоставлении пользователям информации и взаимодействия с веб-приложением. В Flask для генерации HTML страниц используются шаблоны Jinja2, которые позволяют динамически вставлять данные и управлять логикой отображения.

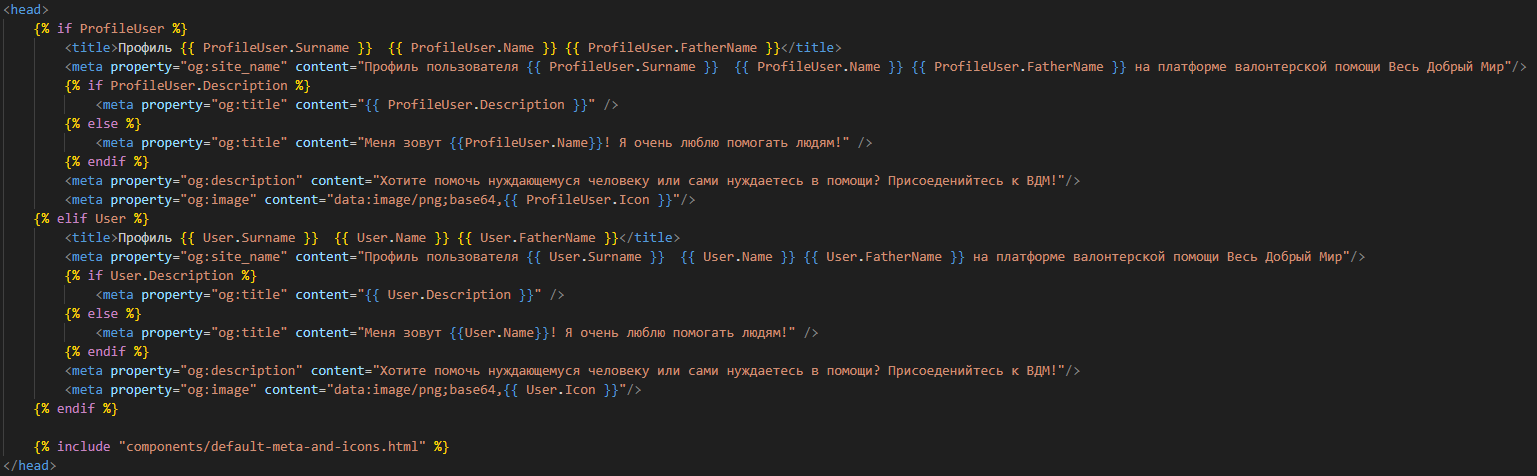
Список страниц

В информационной системе "Весь Добрый Мир" используются следующие HTML страницы:



1. Список представлений (Views)

Рассмотрим HTML код страницы профиля пользователя profile.html:



1. Заголовок и мета-теги

* Проверка существования ProfileUser:

Если передан ProfileUser, заголовок и мета-теги страницы заполняются данными профиля этого пользователя.

* Заголовок страницы:

<title>Профиль {{ ProfileUser.Surname }} {{ ProfileUser.Name }} {{ ProfileUser.FatherName }}</title>: Устанавливает заголовок страницы на основе фамилии, имени и отчества пользователя.

* Мета-теги Open Graph:

<meta property="og:site\_name" content="Профиль пользователя {{ ProfileUser.Surname }} {{ ProfileUser.Name }} {{ ProfileUser.FatherName }} на платформе волонтерской помощи Весь Добрый Мир"/>: Устанавливает название сайта и описание профиля пользователя.

<meta property="og:title" content="{{ ProfileUser.Description }}" />: Устанавливает заголовок на основе описания профиля пользователя, если оно существует. В противном случае используется дефолтный заголовок: "Меня зовут {{ProfileUser.Name}}! Я очень люблю помогать людям!".

<meta property="og:description" content="Хотите помочь нуждающемуся человеку или сами нуждаетесь в помощи? Присоединяйтесь к ВДМ!"/>: Устанавливает описание страницы.

<meta property="og:image" content="data:image/png;base64,{{ ProfileUser.Icon }}"/>: Устанавливает изображение профиля.

* Проверка существования User:

Если ProfileUser не существует, используется User, и аналогично устанавливаются заголовок и мета-теги.

* Включение стандартных мета-тегов и иконок:

{% include "components/default-meta-and-icons.html" %}: Включает общий файл с мета-тегами и иконками для страницы.

По итогу в данном проекте происходит использование современных технологий и следование модели MVC, что обеспечивает структурированность и гибкость в создании веб-приложения.

Этот комплексный подход позволяет создать эффективное, масштабируемое и удобное в использовании веб-приложение, которое будет поддерживать деятельность организации, способствовать привлечению новых волонтеров и партнеров, а также укреплять имидж организации в глазах общественности.

2.5 Этапы Процедур Отладки и Тестирования Информационной Системы

Этапы тестирования и отладки в проекте играют ключевую роль в обеспечении качества и надежности информационной системы для ММОО «Весь Мир Един». Процесс начинается с модульного тестирования, в ходе которого каждая отдельная функция и компонент проверяются на корректность работы. На этом этапе разработчики проводят детальные проверки кода, чтобы убедиться в отсутствии ошибок и багов в отдельных модулях. Модульное тестирование позволяет выявить и устранить проблемы на ранней стадии разработки, что значительно снижает риск возникновения серьезных ошибок в будущем.

После успешного завершения модульного тестирования начинается интеграционное тестирование, где проверяется взаимодействие между различными компонентами системы. Это важный этап, так как именно здесь выясняется, насколько правильно интегрированы модули и как они функционируют в совокупности. Особое внимание уделяется проверке корректности передачи данных между компонентами и стабильности их работы при взаимодействии.

Следующим этапом является системное тестирование, в рамках которого проводится комплексная проверка всей системы на соответствие требованиям и спецификациям. Здесь тестируются все функциональные возможности системы, проверяется работа пользовательского интерфейса и производится нагрузочное тестирование для оценки производительности системы под разными условиями. Этот этап позволяет убедиться в том, что система работает правильно и эффективно в реальных условиях эксплуатации.

Тестирование пользовательского опыта (UX-тестирование) является важным элементом, направленным на оценку удобства использования интерфейса, интуитивности навигации и общего впечатления пользователя от взаимодействия с системой. В процессе UX-тестирования собирается обратная связь от пользователей, на основании которой вносятся необходимые изменения и улучшения, что позволяет создать более удобный и привлекательный продукт.

Завершающим этапом является тестирование безопасности, включающее проверку системы на уязвимости, анализ потенциальных угроз и реализацию мер по защите данных пользователей. В ходе этого этапа тестируются различные сценарии атаки, проверяется надежность аутентификации и авторизации, а также шифрование данных. Этот этап критически важен для обеспечения безопасности и конфиденциальности пользовательской информации.

Все проведенные этапы тестирования и отладки тесно связаны между собой и направлены на создание качественного и надежного продукта. Комплексный подход к тестированию позволяет своевременно выявить и устранить проблемы, обеспечивая высокое качество конечного продукта и удовлетворение требований пользователей.

Пример Тестирования

Pass

# ГЛАВА 3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Экономическая часть ВКР содержит экономическое обоснование проектных предложений. Экономическое обоснование должно содержать расчеты, которые показывают, что благодаря внедрению разработки снизится себестоимость того или иного производственного процесса. Смысл вашей ВКР в улучшении каких-либо процессов на предприятии, экономический эффект от которого может выражаться в следующих количественных изменениях:

- повышение производительности рабочего места/объекта инфраструктуры\всей инфраструктуры, в результате чего сокращается необходимое количество человеко-часов или машино-часов для выполнения того же объема операций. Соответственно, также за единицу времени может быть выполнено больше операций, что, например, может принести больше выручки предприятию;

- сокращение доли ручного труда и количества персонала, необходимого для выполнения рутинных операций, экономия на фонде оплаты труда;

Определение трудоемкости работ по обследованию объекта, для которого разрабатывается проект.

Таблица 1.

|  |  |  |
| --- | --- | --- |
| № | Наименование операции | Трудоемкость, час. |
| 1  2  3 | Анализ текущих проектов поддержки волонтерского движения. Поиск и установка ПО для разработки веб-приложений  Анализ хостинг сервисов на рынке | 30  10  15 |

Итого суммарная трудоемкость работ по обследованию объекта - 55 час. (ТЕо)

(В Таблице 1 отражаются подробные операции по обследованию объекта)

Определение трудоемкости работ по разработке проекта.

Таблица 2.

|  |  |  |
| --- | --- | --- |
| № | Наименование операции | Трудоемкость, час. |
| 1 | Настройка окружения для разработки | 5 |
| 2 | Моделирование и разработка базы данных | 60 |
| 3 | Разработка backend-части | 120 |
| 4 | Разработка frontend-части | 120 |
| 5 | Интеграция и тестирование модулей | 15 |
| 6 | Функциональное тестирование | 16 |
| 7 | Бета-тестирование | 20 |
| 8 | Исправление ошибок | 30 |
| 9 | Развертывание системы | 8 |

Итого суммарная трудоемкость работ по разработке проекта – 394 час. (ТЕп)

(В Таблице 2 отражаются подробные операции по разработке проекта)

Смета расходов по разработке проекта.

1. Расходы на используемое оборудование.

Таблица 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование | Цена, руб. | Срок службы, лет | Амортизация в год, руб. | Стоимость затрат, руб. |
| 1 | Монитор | 15 000 | 5 | 3 000 | (55+394)\*3000/1973 = 680.83 |
| 2 | Мышь | 1 500 | 3 | 500 | (55+394)\*500/1973 = 113.47 |
| 3 | Клавиатура | 2 000 | 4 | 500 | (55+394)\*500/1973 = 113.47 |
| 4 | Персональный компьютер | 60 000 | 8 | 7 500 | (55+394)\*7500/1973 = 1702.07 |
| 5 | МФУ (принтер) | 10 000 | 5 | 2 000 | (55+394)\*2000/1973 = 453.89 |

Итого стоимость затрат на используемое оборудование - 3063.73 руб. (Соб)

(В Таблице 3 отражается оборудование, используемое при разработке проекта, например, ПК, принтер, сканер, фототехника и т.п.)

1. Расходы на используемые программные средства.

Таблица 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование | Цена, руб. | Срок эксплуатации (до выхода новых версий), лет | Расходы в год, руб. | Стоимость затрат, руб. |
| **1** | **2** | **3** | **4** | **5** | **6** |
| 1 | MS Windows | 12000 | 3 | 4 000 | 4000 х (60+ 452) / 1973 =  1038 руб. |

Итого стоимость затрат на используемые программные средства - 1038 руб. (Спо)

(В Таблице 4 отражаются лицензионные программные средства, используемые при разработке проекта, например, ОС, СУБД, язык программирования, Photoshop и т.п.)

1. Сводная таблица расходов на создание проекта.

Таблица 5.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Расходы в месяц, руб. | Затраты за период разработки, руб |
| 1 | Трудозатраты по обследованию объекта (с учетом отчислений во внебюджетные фонды в размере 30%) |  | ТЕо х 365 (норма 1 часа, руб.) х 1.3 = 26 037.5 |
| 2 | Трудозатраты по разработке проекта (с учетом отчислений во внебюджетные фонды в размере 30%) |  | ТЕп х 365 (норма 1 часа, руб.) х 1.3 = 186 647 |
| 3 | Используемое оборудование |  | Соб Таблица 3 = 3 063.73 |
| 4 | Используемые программные средства |  | Спо Таблица 4 = 1 038 |
| 5 | Материалы, сырье, комплектующие | 150 | 150 х (ТЕо + ТЕп) / 168 = 449.72 |
| 6 | Затраты на документацию, внедрение и обучение сотрудников Заказчика |  | 25 000 |
| 6 | Затраты на рекламу | 10 000 | 15 000 |
| 7 | Транспортные расходы | 700 | 700 х (ТЕо + ТЕп) / 168 = 2 098.28 |
| 8 | Коммунальные услуги | 7 000 | Среднемесячная стоимость услуг х (ТЕо + ТЕп) / 168 = 18 614.58 |
| 9 | Электроэнергия | 1 500 | Среднемесячная стоимость энергии х (ТЕо + ТЕп) / 168 = 3 981.25 |
| 10 | Телефон |  | Среднемесячная стоимость услуг связи х (ТЕо + ТЕп) / 168 = 1 327.08 |
| 11 | Интернет |  | Среднемесячная стоимость услуг х (ТЕо + ТЕп) / 168 = 2 654.17 |

Итого расходы на создание проекта - 285 911.31 руб. (С)

Определение доходов от реализации проекта. Маркетинговое исследование.

1. Изучение потребителя.

(Анализ, кто является потребителем.)

1. Цена реализации проекта на рынке.

Таблица 6.

|  |  |  |
| --- | --- | --- |
| № | Наименование цены | Цена, руб. |
| 1 | Минимальная цена на рынке | 500 000 |
| 2 | Максимальная цена на рынке | 1 500 000 |
| 3 | **Цена реализации (Ц)** | 595 000 |

(Цена реализации устанавливается исходя из анализа минимальной, максимальной цены аналогичного проекта на рынке и цены, которую готов заплатить заказчик проекта)

Оценка экономической эффективности разработки проекта.

1. Выручка = 595 000 руб. (Ц)
2. Выручка без НДС = 495 833 руб. (ЦЧ = Ц/1.2)
3. Расходы = 285 911 руб. (С)
4. Прибыль = 209 922 руб. (П = ЦЧ – С)
5. Налог на прибыль = 31 488 руб. (НП = П х 0.15)
6. Чистая прибыль = 178 434 руб. (ПЧ = П - НП)
7. Рентабельность реализованной продукции = 35,98%

(ROM = ПЧ / ЦЧ (выручке) х 100%)

Расчет экономической эффективности использования проекта.

1. **Сопоставление трудоемкости выполнения операций.**

Расчёт экономической эффективности внедрения проекта производится на основании сопоставления трудоёмкости выполнения операций в ручном режиме и с использованием разработанной автоматизированной информационной системы (АИС).

Таблица 7

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование показателя | Обозначение | Единица измерения | Исходные данные |
| Среднее количество операций, направленных на поддержку волонтерского движения. | Na | ед. в день | 20 |
| Время, затрачиваемое на такую операцию (до внедрения автоматизации) | Ta1 | час | 0,32 |
| Время, затрачиваемое на такую операцию на предприятии (после внедрения автоматизации) | Ta2 | час | 0,08 |

Для расчёта годовых трудозатрат до внедрения автоматизации находим годовую трудоёмкость в часах.

T1 = 248 \* (Na \* Ta1) = 1 587

где 248 – количество рабочих дней в году.

Для расчёта годовых трудозатрат после внедрения автоматизации находим годовую трудоёмкость в часах.

T2 = 248 \* (Na \* Ta2) = 397

1. **Расчет годовых эксплуатационных затрат.**

Часовая заработная плата одного работника, использующего АИС рассчитывается исходя из месячной зарплаты, которая составляет 80000 руб.

ЗАИС = 80000 \* 12 / 248 / 8 = 483,87 руб.

Отчисления на социальные нужды.

Осн = ЗАИС \* 30% = 483,87 \* 0,3 = 145,16 руб.

Годовые эксплуатационные затраты при ручном режиме работы составят:

А = Т1 \* (ЗАИС + Осн) = 998 270 руб/год

Годовые эксплуатационные затраты при автоматизированном режиме работы составят

П = Т2 \* (ЗАИС + Осн) = 249 724 руб/год

Годовая экономия эксплуатационных затрат в натуральном выражении равна:

Эн – Т1-Т2 (часов). Сэкономленное рабочее время сотрудников можно использовать для их перепрофилирования или сократить затраты на персонал путем увольнения высвободившейся рабочей силы.

Годовая экономия эксплуатационных затрат в денежном выражении равна:

Эгэ = А – П (в рублях) = 748 546 руб.

1. **Оценка экономической эффективности использования проекта.**

Окупаемость капитальных вложений определим по формуле:

Ток = Ц / Эгэ = 0,79 года или 288,35 дня

Результаты расчетов сведем в таблицу:

Таблица 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №№  п/п | Наименование показателя | Единица  измерения | Значение показателя | |
| до внедрения | проект |
| 1. | Трудоёмкость обработки информации | час. | 1 587 (Т1) | 397 (Т2) |
| 2. | Капитальные вложения | руб. | \_\_\_\_\_\_ | 595 000 (Ц) |
| 3. | Годовые эксплуатационные расходы | руб./год | 998 270 (А) | 249 724 (П) |
| 4. | Годовая экономия эксплуатационных расходов | руб./год | 748 546 (Эгэ) | |
| 5. | Срок окупаемости | лет | 0,79 (Ток) | |

Выводы*:*

Внедрение проекта "Весь Добрый Мир" предоставляет предприятию ММОО "Весь Мир Един" возможность значительного снижения затрат на привлечение и управление волонтерами, а также повышение эффективности оказания помощи нуждающимся. Этот проект способствует укреплению социальной ответственности предприятия и улучшению его репутации в сообществе.

Проведённые расчёты демонстрируют, что данный проект обладает высокой экономической эффективностью и обеспечивает быструю окупаемость инвестиций. Экономические модели подтверждают, что вложения в этот проект принесут значительные финансовые выгоды в кратчайшие сроки, делая его перспективным и устойчивым с точки зрения рентабельности.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта была разработана информационная система поддержки волонтерского движения для ММОО «Весь Мир Един».

Проведенные исследования разработки и сама разработка подтвердили актуальность создания специализированного программного обеспечения для оптимизации работ и продвижения некоммерческих организаций.

В первой главе был произведен анализ методов и инструментов разработки, что помогло сделать выбор в языках программирования. Детальная оценка требований и потребностей целевой аудитории помогла добиться оптимальных выборов в проектировании дизайна и функций информационной системы.

Во второй главе проведена разработка базы данных, серверной и клиентской частей веб-приложения. Были использованы современные технологии: HTML, CSS, JavaScript, Python, Flask и Jinja2, что обеспечило высокую производительность, надежность и оптимизацию дальнейшей поддержки информационной системы.

В третей главе экономическая оценка проекта показала его высокую рентабельность и быструю окупаемость, подтверждая целесообразность и эффективность его реализации.

Разработка и в последствии внедрение информационной системы снизила трудозатраты на операции поддержки волонтерского движения и помогла автоматизировать систему отправления запросов о помощи и систему отклика на них волонтеров.

Таким образом, дипломный проект успешно демонстрирует применение современных методов и технологий в разработке информационных систем в виде веб-приложений. Полученная информационная система будет дальше продолжать работать и улучшаться, что поможет организации ММОО «Весь Мир Един» в дальнейшей волонтерской работе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Погорелов, С. В. Программирование на Python 3: подробное руководство / С. В. Погорелов. - СПб.: БХВ-Петербург, 2019. - 480 с.
2. Марков, А. В. Flask. Практическое программирование на Python / А. В. Марков. - М.: ДМК Пресс, 2021. - 352 с.
3. Гринберг, Я. Л. HTML и CSS. Путь к совершенству / Я. Л. Гринберг. - М.: ДМК Пресс, 2020. - 384 с.
4. Гринберг, Я. Л. JavaScript. Путь к совершенству / Я. Л. Гринберг. - М.: ДМК Пресс, 2019. - 448 с.
5. Фланаган, Д. JavaScript: Подробное руководство / Д. Фланаган. - М.: Вильямс, 2017. - 704 с.
6. Ramalho, L. Fluent Python: Clear, Concise, and Effective Programming / L. Ramalho. - O'Reilly Media, 2015. - 792 c.
7. Grinberg, M. Flask Web Development: Developing Web Applications with Python / M. Grinberg. - O'Reilly Media, 2018. - 336 c.
8. Shaw, Z. Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code / Z. Shaw. - Addison-Wesley Professional, 2017. - 320 c.
9. Duckett, J. HTML and CSS: Design and Build Websites / J. Duckett. - Wiley, 2011. - 490 c.
10. Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development / J. Duckett. - Wiley, 2014. - 640 c.
11. W3Schools. HTML Tutorial. [Электронный ресурс]. https://www.w3schools.com/html/
12. Mozilla Developer Network (MDN). CSS Reference. [Электронный ресурс]. https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
13. JavaScript.info. The Modern JavaScript Tutorial. [Электронный ресурс]. https://javascript.info/
14. Jinja2 Documentation. [Электронный ресурс]. https://jinja.palletsprojects.com/
15. Flask Documentation. [Электронный ресурс]. https://flask.palletsprojects.com/
16. Python.org. The Python Language Reference. [Электронный ресурс]. https://docs.python.org/3/reference/
17. O'Reilly Safari Books Online. [Электронный ресурс]. https://www.safaribooksonline.com/

|  |  |
| --- | --- |
| ПРИЛОЖЕНИЯ | **ПРИЛОЖЕНИЕ 1** |

Главная страница

Страница Входа

Страница Регистрации

Страница Профиля пользователя

Страница формы создания Запроса о помощи

Страница созданного Запроса о помощи

Страница списка и поиска Запросов о помощи