**Содержание**

[Введение 5](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803841)

[1 Постановка задачи](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803842) 6

[1.1 Обзор аналогичных решений 6](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803843)

[1.2 Техническое задание 9](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803844)

[1.3 Выбор средств реализации программного продукта 9](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803845)

[1.4 Вывод 10](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803846)

[2 Проектирование страниц веб-сайта 11](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803847)

[2.1 Выбор способа верстки 11](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803848)

[2.2 Выбор стилевого оформления 11](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803849)

[2.3 Выбор шрифтового оформления 12](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803850)

[2.4 Разработка логотипа 12](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803851)

[2.5 Разработка пользовательских элементов 13](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803852)

[2.6 Разработка спецэффектов 16](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803853)

[2.7 Выводы 17](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803854)

[3 Реализация структуры веб-сайта 18](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803855)

[3.1 Структура HTML-документа 18](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803856)

[3.2 Добавление таблиц стилей Scss и CSS 19](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803857)

[3.3 Использование стандартов XML (SVG) 20](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803858)

[3.4 Использование JavaScript 20](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803863)

[3.5 Выводы 21](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803859)

[4 Тестирование веб-сайта 22](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803860)

[4.1 Адаптивный дизайн веб-сайта 22](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803861)

[4.2 Кроссбраузерность веб-сайта 24](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803862)

[4.3 Руководство пользователя 25](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803863)

[4.4 Тестирование кода 27](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803863)

[4.5 Выводы 27](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803864)

[Заключение 28](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803865)

[Список использованных источников 29](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803866)

[ПРИЛОЖЕНИЕ А. Прототипы веб-страниц 30](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803867)

[ПРИЛОЖЕНИЕ Б. Макет структуры веб-сайта 36](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803868)

[ПРИЛОЖЕНИЕ В. Листинг HTML-документа 43](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803869)

[ПРИЛОЖЕНИЕ Г. Листинг Scss 65](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803870)

[ПРИЛОЖЕНИЕ Д. Листинг CSS 68](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803871)

[ПРИЛОЖЕНИЕ Е. Листинг XML-файлов 80](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803872)

[ПРИЛОЖЕНИЕ Ж. Листинг SVG 81](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803873)

[ПРИЛОЖЕНИЕ И. Листинг JS 86](file:///C:\Users\Дмитрий\Downloads\Telegram%20Desktop\пояснительная%20записка%20(2).docx#_Toc103803874)

Введение

Присутствие службы занятости в онлайн пространстве может оказать значительное воздействие на ее успех, независимо от отрасли. Некоторые организации до сих пор не осознают, что большинство клиентов ищут вакансии и информацию о трудоустройстве именно через веб-сайты служб занятости перед тем, как принять решение о трудоустройстве. Присутствие в интернете вашей службы занятости может иметь решающее значение для привлечения большего числа соискателей и увеличения эффективности поиска работы.

Веб-сайт службы занятости играет ключевую роль в обеспечении доступа к качественной и достоверной информации о вакансиях и услугах по трудоустройству. Отсутствие веб-присутствия может затруднить поиск работы и уменьшить поток потенциальных соискателей.

Кроме того, веб-сайт предоставляет возможность сбора заявок от компаний, ищущих новых сотрудников. Возможность оставить заявку на поиск персонала онлайн упрощает процесс для работодателей, делая его более быстрым и удобным. Анализ результатов поступающих заявок и статистики их обработки помогает оптимизировать работу службы занятости и улучшить ее эффективность в привлечении новых партнеров и работодателей.

**Цель** курсовой работы: разработать веб-сайт для магазина зоотоваров с использованием HTML5, а также с применением Scss/CSS3.

**Задачи** курсовой работы:

* проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта;
* разработать макет и прототип сайта;
* разработать структуру веб-сайта;
* наполнить сайт информацией по теме;
* протестировать веб-сайт;
* разработать руководство пользователя.

Целевая аудитория веб-сайта службы занятости включает как соискателей, ищущих новые карьерные возможности и вакансии, так и компании, которые ищут квалифицированных специалистов для своих вакансий. Наш сайт предоставляет удобные инструменты для эффективного поиска работы и персонала, подходящего для различных отраслей и специализаций.

1 Постановка задачи

1.1 Обзор аналогичных решений

Перед тем как приступать к разработке своей системы рекомендуется изучить уже готовые решения и посмотреть, что по мнению других разработчиков является наиболее необходимым для пользователей, чтобы приложение было полезным и удобным в использовании.

* + 1. Аналог «gsz.gow.by»

На рисунке 1.1 представлен внешний вид главного каталога сайта [1].

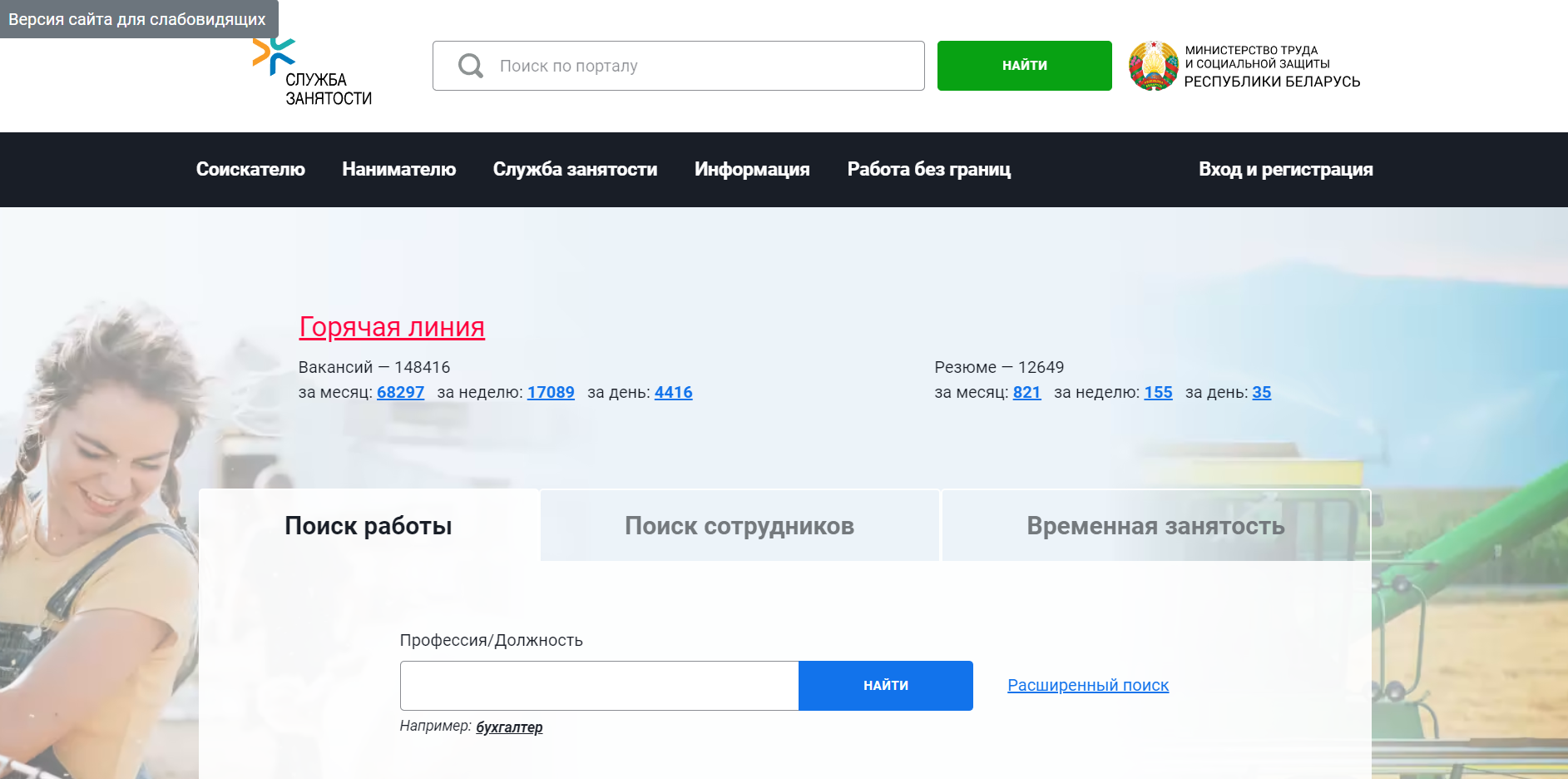


Рисунок 1.1-Окно главной страницы сайта «gsz.gow.by»

Сайт службы занятости разработан в минималистичном стиле, чтобы пользователи могли сосредоточиться на важной информации. Цветовая гамма сайта сфокусирована на желтом цвете, который привлекает внимание аудитории и стимулирует их к действию.

Данный сайт специализируется на предоставлении рабочих возможностей. Основные разделы службы включают в себя:

1. Вакансии (полный список доступных вакансий в различных отраслях и сферах деятельности).
2. Услуги (подробная информация о предоставляемых службой услугах, таких как консультации по трудоустройству, подготовка резюме и другие).
3. Обучение и развитие (информация о профессиональных тренингах, курсах повышения квалификации и других образовательных мероприятиях).
4. Партнеры и программы (сведения о сотрудничестве с компаниями и организациями, а также о программных инициативах).

Для более детального анализа мы рассмотрим один из разделов подробнее. На странице раздела представлены следующие основные категории, необходимые пользователям:

1. Вакансии (полный список текущих вакансий, возможность фильтрации по параметрам).
2. Подготовка к трудоустройству (информация о курсах подготовки, семинарах и мероприятиях, помогающих соискателям успешно пройти собеседования и найти работу).
3. Карьерные советы (статьи и советы по поиску работы, развитию карьеры и успешному трудоустройству).
4. Работодателям (информация о возможностях размещения вакансий, услугах по подбору персонала и сотрудничестве с компаниями).

Далее необходимо рассмотреть подробнее один из разделов, для того чтобы определить наиболее важные данные, которые необходимо предоставить пользователю далее. Данная страница представлена на рисунке 1.2.

На странице представлены основные пункты выбора вакансии, а именно: Профессия/Должность, Область, ЗП.

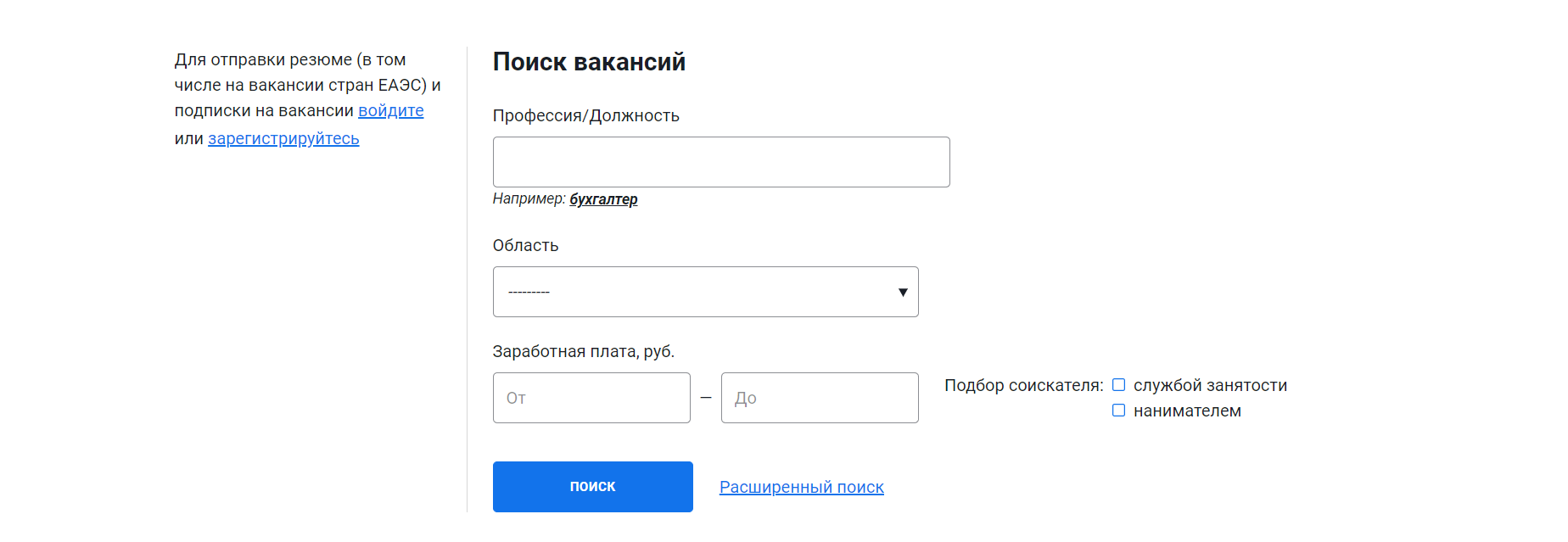


Рисунок 1.2-Окно «Поиск вакансий» веб-сайта сайта «gsz.gow.by»

Из минусов: достаточно большое простанство нижнего навигационного меню.

1.1.2 Аналог «rostrud.gov.ru»

На рисунке 1.3 представлен внешний вид главного каталога сайта

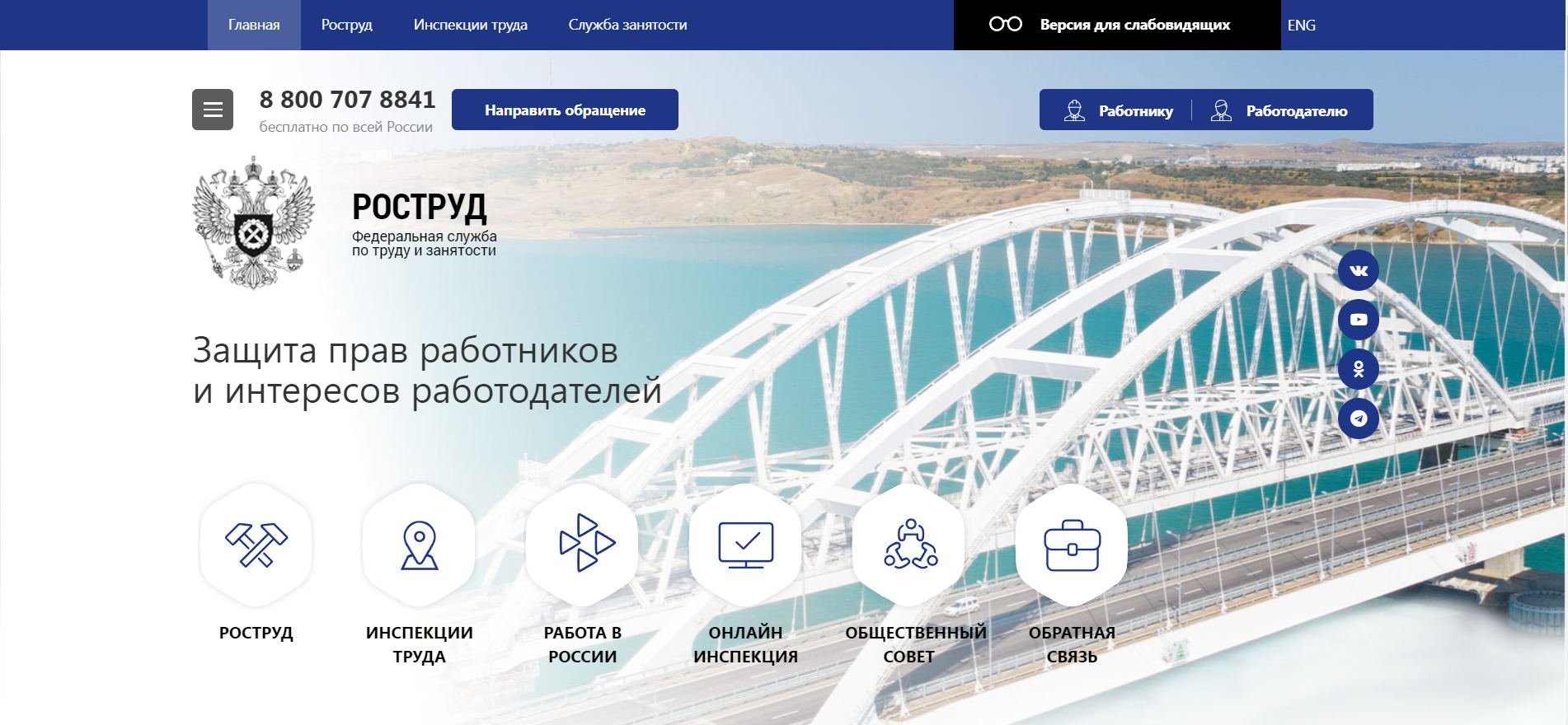


Рисунок 1.3-Окно главной страницы веб-сайта «Аналог «rostrud.gov.ru»

Дизайн этого сайта мне нравится больше. Элементы читаются лучше, чёткий текст, все нужные контакты распологаются на-виду. Однако цветовая гамма заднего фона немного сливается с элементам самого веб-сайта. В качестве основного цвета выбран синий, символизирующий море вакансий.

Далее рассмотрим информационную страницу «РосТруд», она представлена на рисунке 1.4.

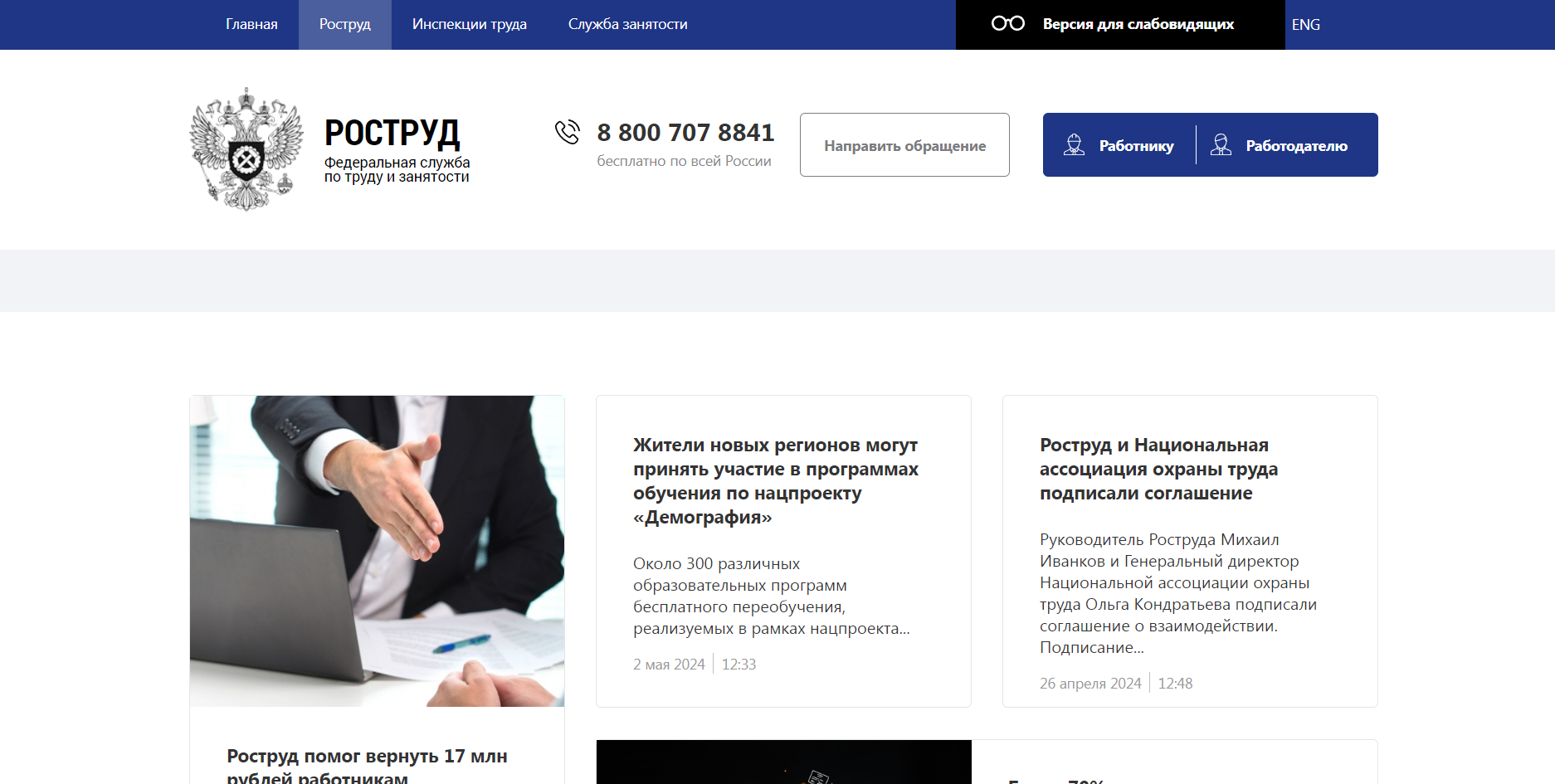


Рисунок 1.4-Окно «РосТруд» сайта «rostrud.gov.ru»

**1.2 Техническое задание**

Задача заключается в создании полноценного многостраничного веб-сайта на тему "Служба занятости". Сайт должен содержать информацию о службе, включая контактные данные, описание услуг и каталог вакансий. Основной акцент будет сделан на каталоге вакансий.

Основные цели сайта:

1. Предоставление навигационного меню для удобного доступа к различным разделам сайта.
2. Предоставление блоков-ссылок для быстрого перехода к нужной информации.
3. Предоставление подробной информации о вакансиях и услугах службы.

На главной странице навигационное меню будет размещаться вверху. Оно будет содержать ссылки на основные разделы сайта, включая информацию о службе, услугах, актуальные вакансии и контактные данные.

Веб-страница будет адаптирована для устройств с разными размерами экранов: для мобильных устройств (ширина до 450px), планшетов (ширина до 1024px) и персональных компьютеров (ширина от 1024px). Адаптивный дизайн обеспечит удобство использования сайта на различных устройствах, сохраняя при этом функциональность и наглядность представления информации.

1.3 Выбор средств реализации программного продукта

Курсовая работа выполняется в редакторе кода Visual Studio Code.

При создании веб-страниц были использованы языки: HTML, CSS, JS.

HTML (HyperText Markup Language) — стандартный язык разметки гипертекстовых страниц в Интернете. Страницы успешно интерпретируются браузерами, которые отображают их на экранах различных электронных устройств в удобном для человека виде.

CSS — это фактически язык стилей, который определяет отображение HTML-документов. CSS работает со шрифтами, с цветами символов и фона, с полями, со строками, с высотой и с шириной элементов отображения, с фоновыми изображениями, с позиционированием элементов и со многим другим.

Если HTML необходим для структурирования содержания страницы, то CSS необходим для того, чтобы форматировать это структурированное содержание.

JavaScript это язык, который позволяет применять сложные вещи на web странице — каждый раз, когда на ней происходит что-то большее, чем просто её статичное отображение —без JavaScript не обошлось. Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять скрипты на странице. JavaScript можно использовать не только в браузере, это полноценный язык, программы на котором можно запускать и на сервере.

2 Проектирование страниц веб-сайта

2.1 Выбор способа верстки

Требования проекта включают в себя адаптивность и кроссбраузерность. По этой причине была выбрана grid и flex-box вёрстки.

Grid-модель позволяет создавать структуры, необходимые для обеспечения отзывчивости сайтов на различных устройствах. Это означает, что сайт будет одинаково хорошо смотреться на компьютере, телефоне и планшете.

Flex – это новая технология, которая уже имеет достаточно широкую поддержку браузеров. Flexbox предоставляет инструменты для быстрого создания сложных, гибких макетов, и функции, которые были сложны в традиционных методах CSS.

2.2 Выбор стилевого оформления

Выбор стилевого оформления является важным этапом разработки проекта по нескольким причинам:

Создание уникального имиджа. Стилевое оформление проекта может помочь создать уникальный и запоминающийся имидж. Оно может помочь подчеркнуть уникальность проекта, выделить его на фоне конкурентов и привлечь больше внимания потенциальных пользователей.

Улучшение пользовательского опыта. Стилевое оформление проекта может влиять на пользовательский опыт. Четко продуманное и привлекательное оформление может сделать проект более удобным и привлекательным для пользователей, что может повысить их удовлетворенность использованием проекта.

Улучшение узнаваемости. Хорошо продуманное и привлекательное стилевое оформление может помочь увеличить узнаваемость проекта. Люди запоминают лучше визуальную информацию, поэтому стилевое оформление может помочь проекту стать более узнаваемым и запоминающимся.

Минимализм как стиль оформления интернет-магазина имеет свои преимущества и может быть эффективным выбором по следующим причинам:

Фокус на продукте. Минималистичный дизайн позволяет убрать все лишнее, что может отвлечь пользователя от продукта, и сосредоточить его внимание на самом товаре. Это может увеличить вероятность того, что пользователь совершит покупку.

Улучшенная навигация. При минималистичном дизайне интернет-магазина упрощается навигация по сайту и повышается его скорость загрузки. Это может улучшить опыт покупки пользователя и увеличить конверсию.

Стиль и элегантность. Минималистичный дизайн может создать ощущение стиля и элегантности, что может быть важным для привлечения пользователей и установления доверия к бренду.

В целом, стилевое оформление является важным аспектом разработки проекта, который может оказать значительное влияние на его успех и эффективность.

2.3 Выбор шрифтового оформления

Было выбрано одношрифтовое оформление текста на веб-странице, это имеет ряд плюсов:

Единообразие: Использование одного шрифта создает единый стиль и общий визуальный облик сайта, что делает его более красивым и профессиональным.

Простота восприятия: Посетители сайта не будут отвлекаться на различия в шрифтах и сосредоточатся на контенте. Это способствует более легкому восприятию информации.

Быстрая загрузка: Использование одного шрифта снижает количество запросов к серверу для загрузки шрифтов, что может ускорить загрузку страниц сайта.

Удобство в управлении: Одношрифтовое оформление упрощает управление и поддержку сайта, поскольку нет необходимости следить за множеством используемых шрифтов и их совместимостью с различными браузерами и устройствами.

Сохранение стиля: Один шрифт может быть выбран в соответствии с общим стилем и брендированием компании, что помогает подчеркнуть ее уникальность и ценности.

2.4 Разработка логотипа

Логотип — это фирменный знак, который люди ассоциируют с конкретным брендом. Логотип — это не абстрактный набор символов или просто красивое изображение.

Логотип сайта является важным элементом брендинга и визуальной идентичности. Вот несколько причин, почему логотип сайта является важным:

Узнаваемость бренда. Логотип является визуальным символом бренда и помогает создать узнаваемость среди потенциальных клиентов. Когда пользователи видят логотип, они могут легко связать его с брендом и лучше запомнить его.

Отражение имиджа компании. Логотип может отражать имидж компании и ее ценности. Например, если компания заботится об окружающей среде, ее логотип может содержать зеленый цвет и символ природы.

Отличие от конкурентов. Логотип может помочь бренду отличиться от конкурентов и выделиться на фоне других компаний.

Привлечение внимания. Логотип может быть ярким и привлекательным, что поможет привлечь внимание потенциальных клиентов.

Усиление брендовой лояльности. Логотип может помочь усилить брендовую лояльность клиентов. Когда клиенты видят логотип, связанный с их любимым брендом, они могут чувствовать более сильную привязанность к бренду.

Логотип выполнен под концепцию сайта. Чтобы вызвать у пользователя ассоциацию с общностью, он представлен в виде группы людей. Логотип, представленный на рисунке 2.2 был создан с помощью сайта svg.io.

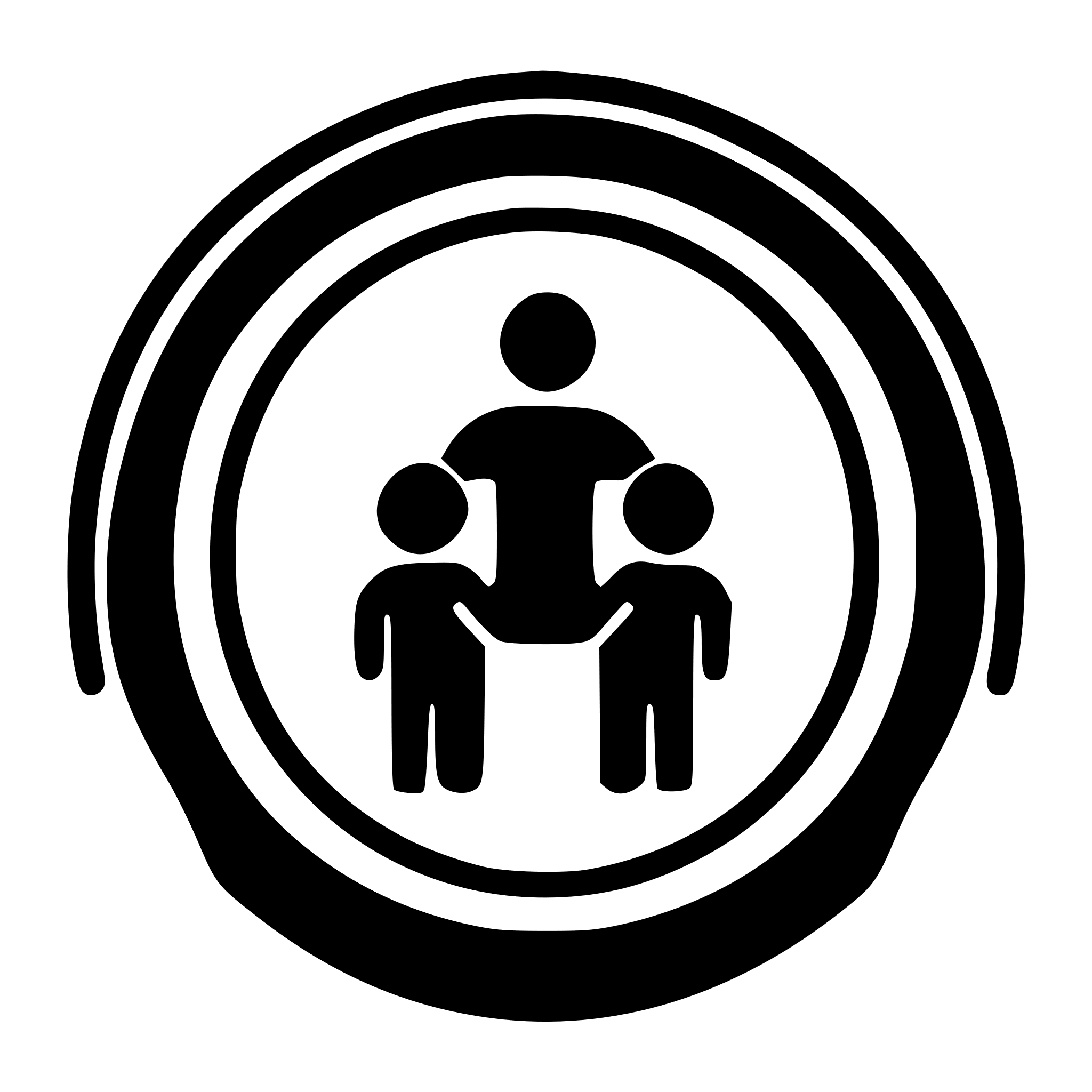


Рисунок 2.2-Логотип сайта «Employment Service»

2.5 Разработка пользовательских элементов

В проекте будут представлены элементы пользовательского интерфейса, с которыми пользователь сможет взаимодействовать: блоки с информацией о товаре, навигационное меню, футер с основной информацией.

Навигация – ключевой критерий удобства сайта, своеобразная «карта местности». Представляет собой набор специальных приемов, методов и элементов, дающих возможность посетителям перемещаться между различными страницами ресурса. Навигационное меню представлено на рисунке 2.3.

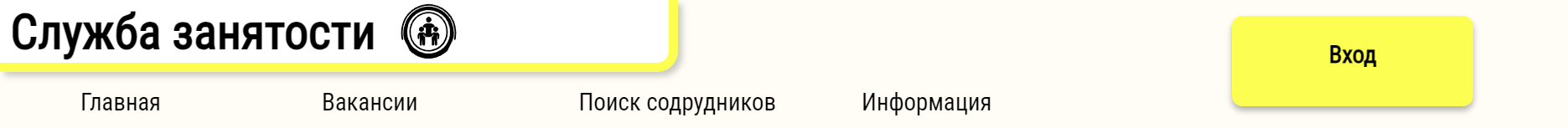


Рисунок 2.3-Навигационное меню

Блок-ссылки в навигационном меню сайта службы занятости должны быть простыми, наглядными и удобными для пользователя. Они должны содержать ссылки на основные разделы и сервисы, предлагаемые службой занятости, такие как поиск вакансий, размещение резюме, информация о компании, контактная информация и возможно ссылки на полезные ресурсы и инструменты для поиска работы. Также блок-ссылки могут включать в себя дополнительные разделы, такие как новости, советы по трудоустройству, обучающие программы и т. д. Они должны быть хорошо структурированы и легко доступны для пользователя, чтобы обеспечить удобство в навигации по сайту.

Информация в блоках должна легко читаться, чтобы не отвлекать внимание пользователя.

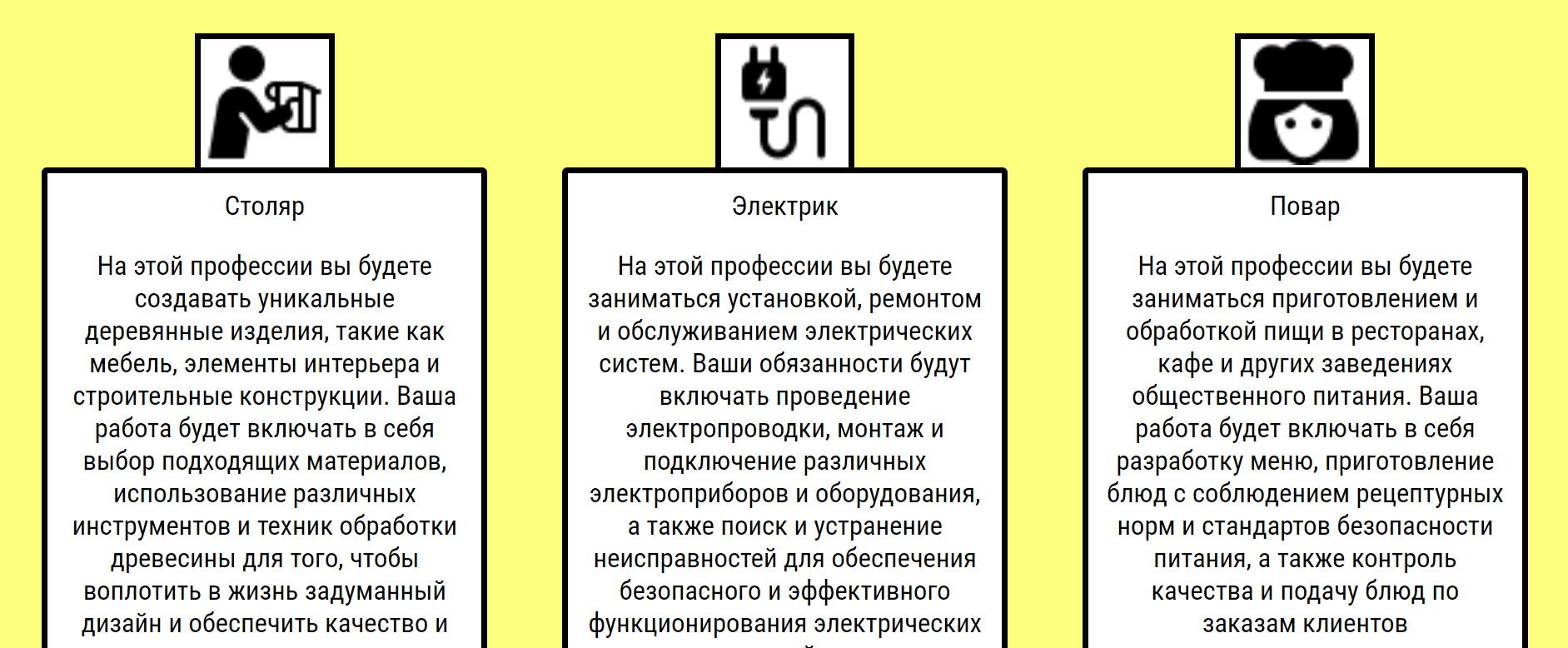


Рисунок 2.4-Информационные блоки сайта «Employment Service»

Лишние элементы могут дизориентировать пользователя, который не имеет достаточных навыков использования подобными сайтами. Для этого были сделаны чёткие блоки, показывающие куда ведёт ссылка.



Рисунок 2.5-навигационное меню сайта «Employment Service»

Текстовая информация должна показывать цель веб-страницы в полной мере. У пользователя не должно появляться ощущение запутанности в информации.

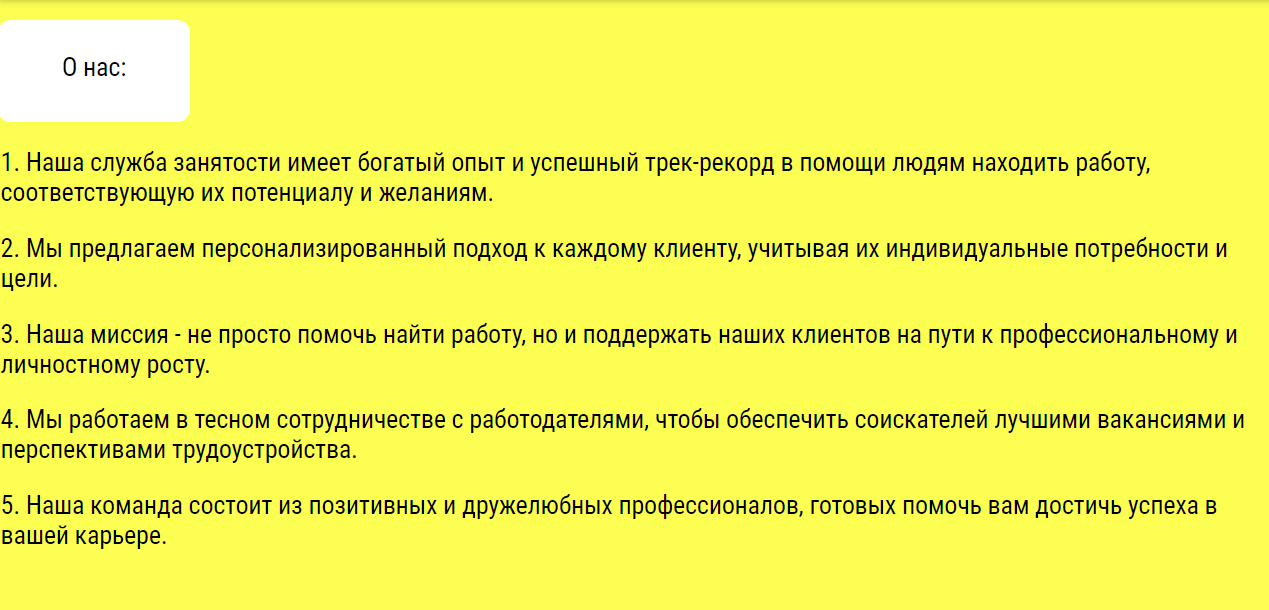


Рисунок 2.6-Информационная страница сайта «Employment Service»

Система авторизации на сайте также не должна вызывать затруднения у пользователя. По возможности она не должна содержать большое количество элементов для ввода данных.

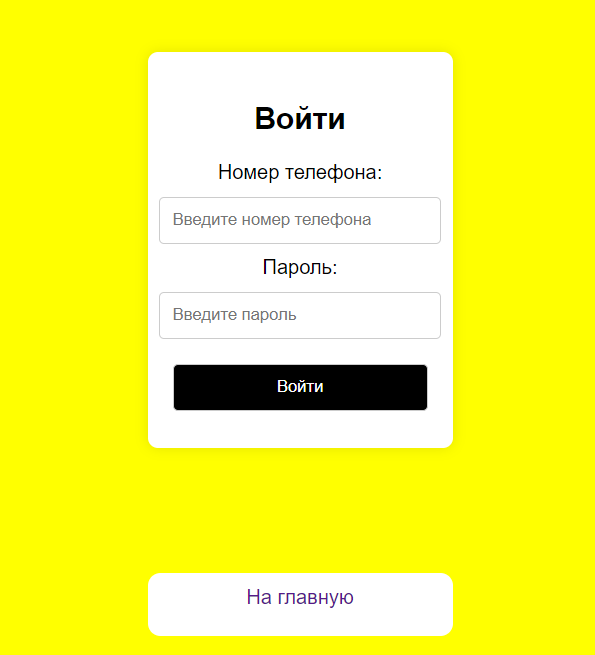


Рисунок 2.8-меню авторизации сайта «Employment Service»

2.7 Выводы

В данном разделе был создан прототип веб-сайта, определены задачи по созданию содержания на веб-сайте. Выбрано единое цветовое и шрифтовое оформления сайта для удобства просмотра пользователем. Разработан дизайн и расположение на сайте пользовательских элементов, спецэффектов и анимации. Логотип выполнен под концепцию сайта.

3 Реализация структуры веб-сайта

3.1 Структура HTML-документа

Структура веб-сайта представляет собой организацию связанных между собой документов, которая позволяет пользователям легко перемещаться по страницам. При формировании структуры необходимо учитывать потребности посетителей, их запросы и интересы.

Хорошая структура HTML-документа имеет большое значение для эффективной работы сайта. Вот несколько причин, почему это так важно:

Улучшение доступности: хорошая структура HTML-документа может сделать сайт более доступным для пользователей с ограниченными возможностями. Например, правильное использование заголовков и других элементов разметки может помочь пользователям, использующим программы чтения с экрана, легче понимать контент.

Хорошая структура HTML-документа может помочь поисковым системам лучше понимать контент страницы. Это может привести к улучшению рейтинга сайта и его поисковой выдаче.

Улучшение скорости загрузки: хорошая структура HTML-документа может ускорить загрузку страницы. Например, правильное использование семантических тегов может помочь браузеру быстрее отрисовать страницу.

Улучшение совместимости: хорошая структура HTML-документа может помочь сделать сайт более совместимым с различными браузерами и устройствами. Это может улучшить пользовательский опыт и увеличить количество посетителей на сайте.

Тег <head> предназначен для хранения служебных элементов, которые не отображаются на странице, за исключением заголовка <title>.

Основное содержимое сайта размещается внутри тега <body>, который включает в себя семантические теги, такие как header, section и footer, которые помогают программам понимать тип информации, содержащейся на странице.

Структура HTML главной страницы представлена в листинге 3.1.

< ! DOCTYPE html>

<html>

<head>…</head>

<body>

<div class=”header\_div”>…</div>

<section class=”section-1”>…</section>

<section class=”section-2”>…</section>

<section class=”section-3”>…</section>

<section class=”section-4”>…</section>

<section class=”section-5”>…</section>

<div class=”footer\_div”>…</div>

</body>

</html>

Листинг 3.1 – Структура HTML главной страницы

С помощью блоков div и соответствующих им классам и стилям были созданы шапка сайта и «подвал» для всех страниц сайта. Теги section визуализируют элементы и информацию, находящиеся между «крышей» и «подвалом».

3.2 Добавление таблиц стилей Scss и CSS

В ходе разработки проекта были применены различные подходы к подключению таблиц стилей, включая внешнее, внутреннее и строковое подключение.

<head>

<title>главная</head>

<link rel=”stylesheet” href=”общее.css”>

<link rel=”stylesheet” href=”главная.css”>

</head>

Листинг 3.2 – Пример внешнего подключения

<style>

.SomeClass1{

color: rgb(255, 0, 0);

font-size: 12px;

margin-top: 5px;

}

.SomeClass2{

color: green;

font-size: 16px;

margin-top: 5px;

}

</style>

Листинг 3.3 – Пример внутреннего подключения

Для определения стилей для повторяющихся элементов и основной разметки страниц было использовано внешнее подключение, при котором стили размещаются в отдельном файле и могут быть использованы на любой странице сайта. Внутреннее подключение было применено для внесения изменений в контент на второстепенных страницах при минимальном объеме кода.

<a style=”text-decoration: none;”><<Вернуться на главную</a>

Листинг 3.4 – Пример внутреннего подключения

Для небольших изменений в стиле текста на страницах сайта было использовано строковое подключение. Все таблицы стилей были организованы с использованием селекторов по классу, что позволило улучшить структуру кода и обеспечить более эффективную поддержку сайта в будущем. Такой подход позволяет быстро и легко вносить изменения в стили сайта, облегчает сопровождение проекта и ускоряет его загрузку.

3.4 Использование JavaScript

JavaScript (JS) является одним из самых распространенных языков программирования для веб-разработки. Этот язык позволяет добавлять динамичность и интерактивность на веб-страницы, что делает пользовательский опыт более увлекательным и позволяет создавать более сложные веб-приложения.

function toggleMenu() {

    var menu = document.getElementById("menu");

    var overlay = document.getElementById("overlay");

    if (menu.style.display === "block") {

        menu.style.display = "none";

        overlay.style.display = "none";

    } else {

        menu.style.display = "block";

        overlay.style.display = "block";

    }

}

function closeMenu() {

    var menu = document.getElementById("menu");

    var overlay = document.getElementById("overlay");

    menu.style.display = "none";

    overlay.style.display = "none";}

Листинг 3.6 – Пример использования JS для создания «бургер» меню.

JavaScript может использоваться для создания различных интерактивных элементов на странице, таких как выпадающие меню, слайдеры, анимации, формы обратной связи и многое другое. С помощью JS можно изменять содержимое, стиль и расположение элементов на странице, а также обрабатывать действия пользователя, такие как клики и ввод текста.

3.5 Выводы

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей CSS. Было продемонстрировано использование стандартов (SVG), а также языка программирования JS. Было определено стилевое оформление веб-сайта, созданы анимации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

4 Тестирование веб-сайта

4.1 Адаптивный дизайн веб-сайта

Для достижения адаптивности в проекте применялись методы верстки grid и flex, а также использовались медиа-запросы. С помощью grid-верстки была разработана основная структура контента на странице. Использование свойства minmax() позволило динамически изменять размер блоков в зависимости от ширины экрана.

Ресурс должен одинаково хорошо демонстрироваться на экране компьютера (рис. 4.1) и смартфоне(рис. 4.2).

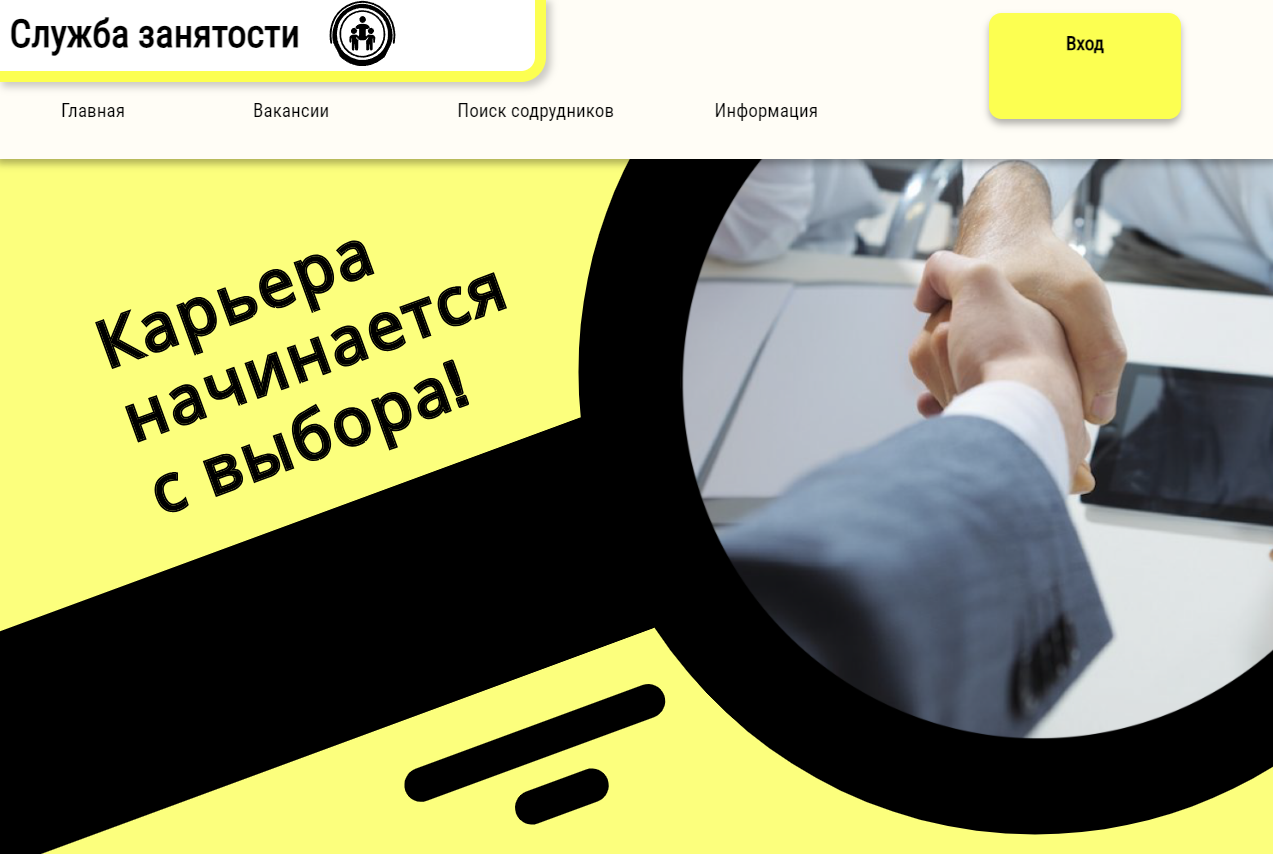


Рисунок 4.1- Вид страницы для компьютера

Медиа- запросы использовались для перемещения элементов основной части страницы при размерах мобильной версии.

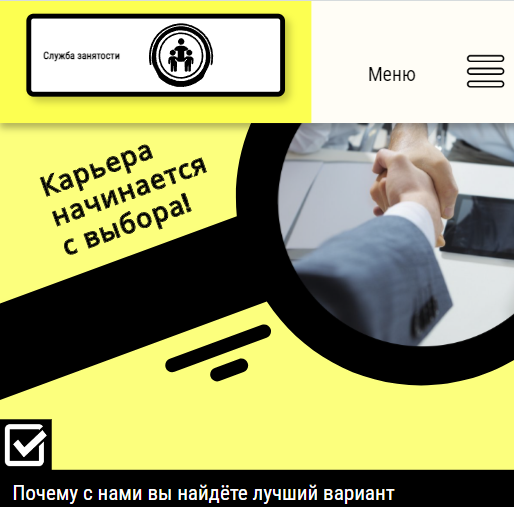


Рисунок 4.2- Вид страницы для мобильных устройств

Адаптивность является очень важным аспектом веб-разработки, так как позволяет сайту корректно отображаться на различных устройствах с разными размерами экранов. С увеличением числа устройств с различными размерами экранов, таких как смартфоны, планшеты, ноутбуки и настольные компьютеры, становится критически важным, чтобы сайт был доступен и удобочитаем на всех устройствах.

Адаптивность позволяет автоматически изменять расположение и размер элементов страницы в зависимости от размера экрана устройства, на котором происходит просмотр сайта. Таким образом, пользователь получает оптимальное визуальное восприятие сайта на любом устройстве и не испытывает неудобств при просмотре страницы.

Кроме того, адаптивность сайта имеет важное значение для улучшения SEO-оптимизации, так как поисковые системы высоко оценивают мобильную доступность сайтов и ранжируют их в соответствии с этим фактором.

Для этого создается адаптивный дизайн и используется адаптивная верстка. В итоге сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

4.2 Кроссбраузерность веб-сайта

Кроссбраузерность – важный критерий корректной работы веб-ресурса. Это инструмент поддержания посетительской лояльности, повышения конверсии и поднятия рейтинга в поисковых системах.

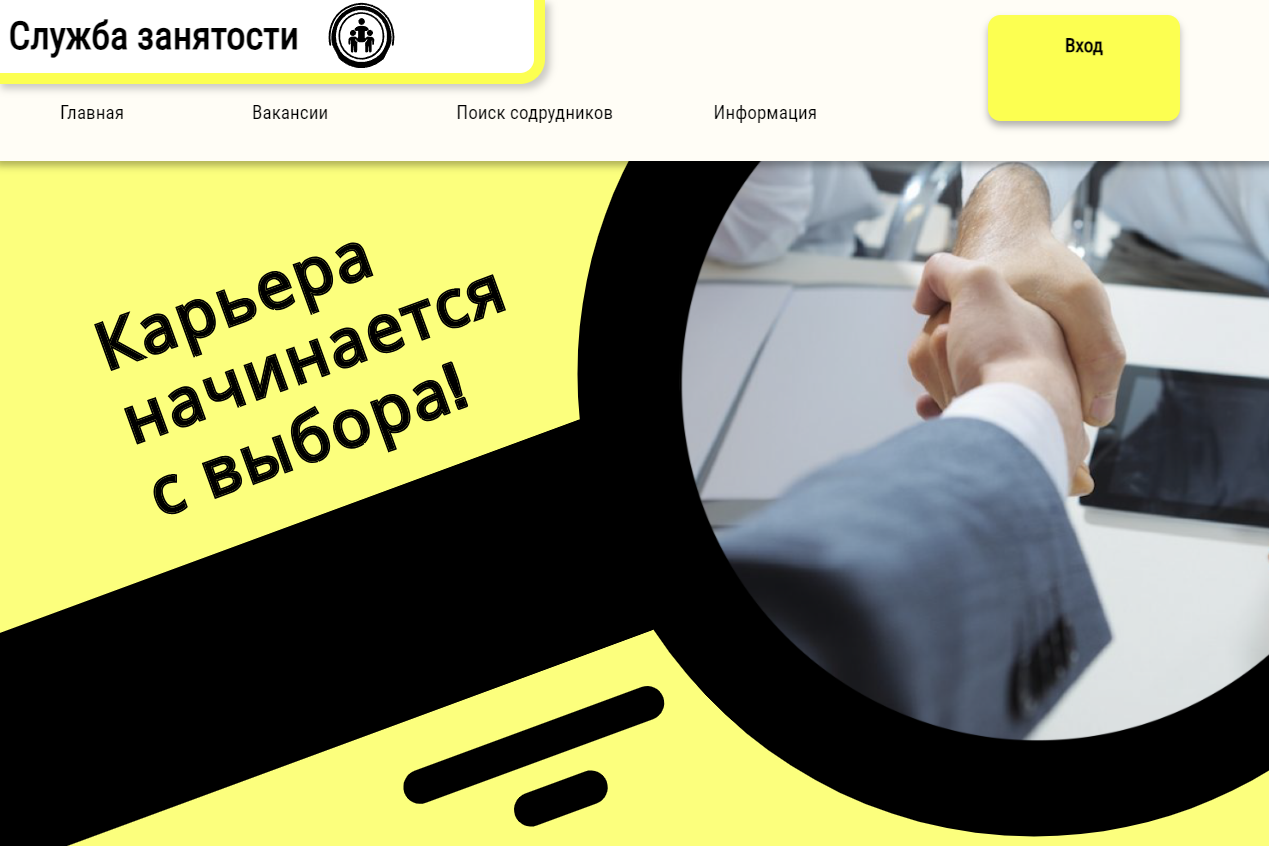


Рисунок 4.3- Вид страницы в браузере Google Chrome

После написания основной структуры страницы на HTML и внешнего стилевого оформления появился вопрос о тестировании. Веб-сайт был открыт при помощи различных браузеров.

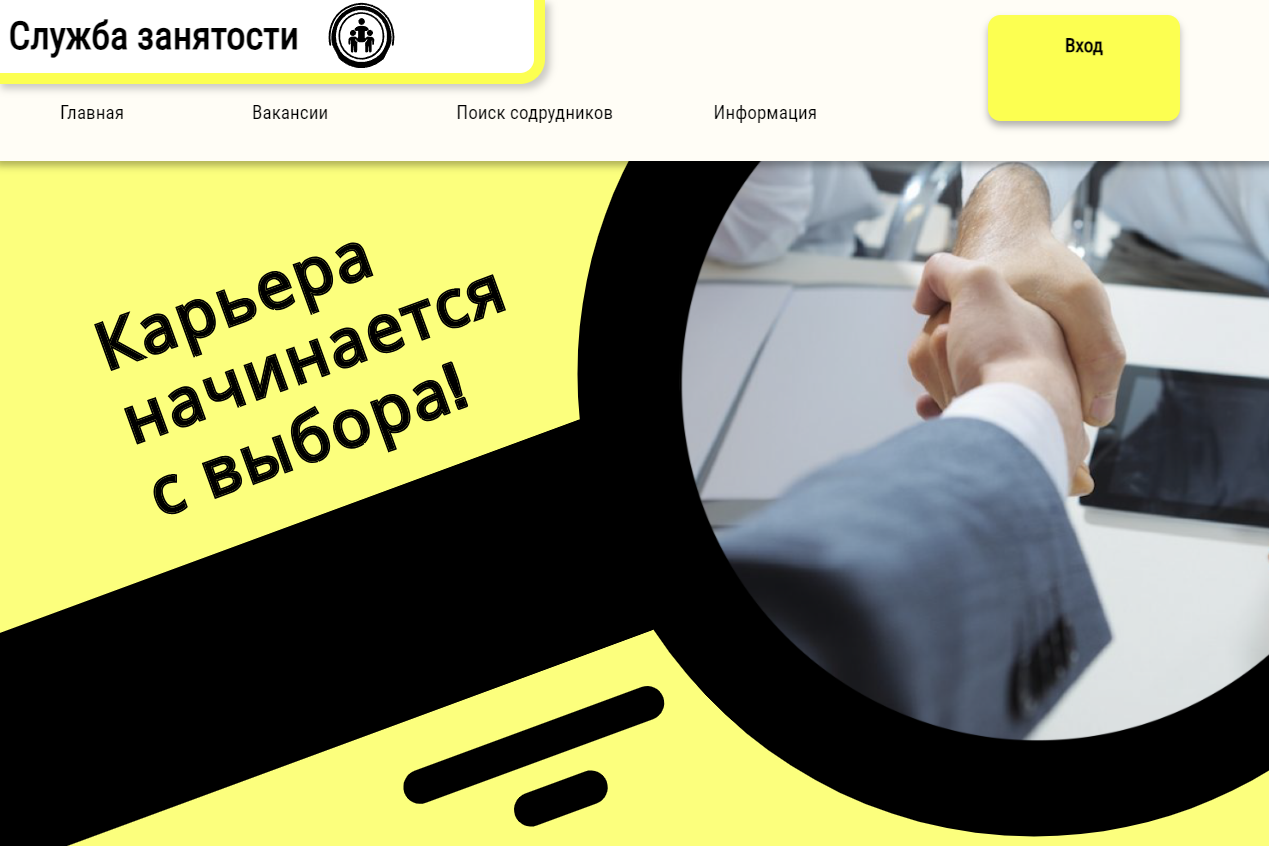


Рисунок 4.4- Вид страницы в браузере Microsoft Edge

Так как проверка на кроссбраузерность возможна на данном этапе разработки только в ручную, некоторые свойства, которые некорректно отображаются в различных браузерах, могли остаться незамеченными.

4.3 Руководство пользователя

Интерфейс сайта интуитивно понятен пользователю. Основное взаимодействие с сайтом в поиске товара удобнее всего осуществлять с помощью ссылок на различные страницы веб-сайта(рис. 4.5).



Рисунок 4.5-Ссылки на информационные страницы

На рисунке представлены страницы навигации по сайту. В навигации пользователь может, например, перейти на страницу вакансий (рис. 4.6)

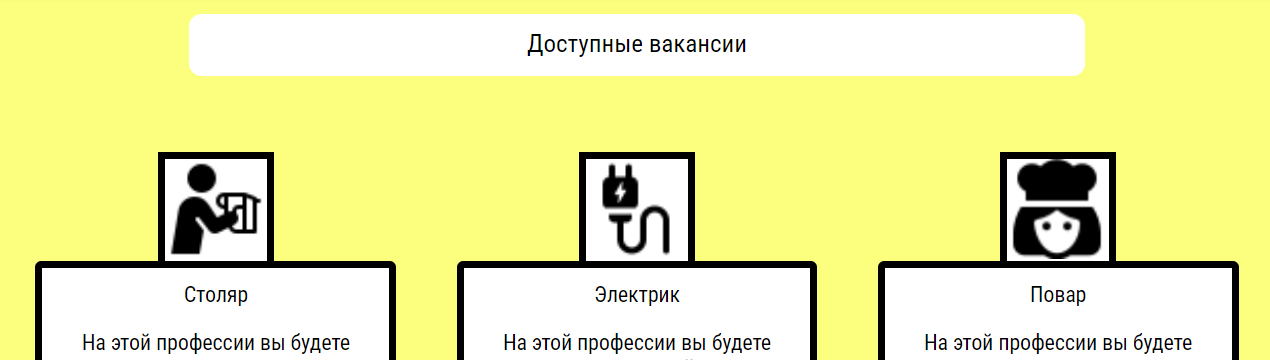


Рисунок 4.6-Страница вакансий

На странице пользователь может ознакомиться с доступными на данный момент вакансиями.

Таким образом, был разработан сайт с целью обеспечить максимально простое и интуитивно понятное использование.

4.4 Тестирование кода

Тестирование кода веб-страниц является критически важной частью процесса веб-разработки. Это процесс проверки программного кода, чтобы убедиться, что он работает так, как должен работать, и что он соответствует определенным стандартам и требованиям.

Тестирование позволяет убедиться, что код работает правильно и соответствует заданным требованиям. Это помогает обеспечить высокое качество и надежность сайта. Тестирование помогает обнаружить уязвимости и потенциальные угрозы безопасности, что позволяет разработчикам принимать меры для их устранения до того, как они могут быть использованы злоумышленниками. Ошибка в коде, которая обнаруживается на ранней стадии, может быть исправлена быстрее и с меньшими затратами, чем если бы она была обнаружена позже в процессе разработки или после выпуска сайта. Правильно работающий сайт, который не вызывает ошибок и проблем, улучшает опыт пользователя, что может привести к большей лояльности и повторному использованию сайта.

Весь код был протестирован с помощью онлайн-сервиса linter. Результат представлен на рисунке ниже.

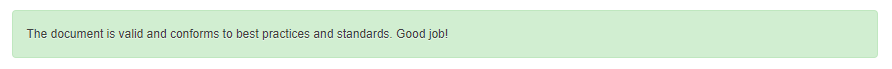


Рисунок 4.7-Результат тестирования кода на сайте linter.

В целом, тестирование кода веб-страниц является неотъемлемой частью процесса веб-разработки и помогает гарантировать высокое качество, безопасность и удобство использования сайта.

4.5 Выводы

В данном разделе я было рассмотрено, как сайт ведет в себя на разных устройствах и в разных браузерах. С помощью тестирования были выявлена некоторые недочеты для дальнейшего усовершенствования продукта. Так же создано руководство пользователя, где кратко расписано, как пользоваться сайтом.

Заключение

В результате проделанной работы был разработан полноценный веб-сайт для службы занятости, который предоставляет широкий спектр услуг для поиска работы и трудоустройства.

При разработке данного веб-сайта был использован широкий ряд языков разметки и таблиц стиля, который позволил увеличить функциональность и многократно повысил качество конечного продукта. Была использована адаптивная верстка для корректного отображения на всех устройствах.

В ходе выполнения данной задачи были рассмотрены плюсы и минусы аналогичных решений. В процессе осмотра выяснилось, каких тенденции в разработке дизайна веб-сайта стоит придерживаться.

При формировании технического задания были продумано содержание основных страниц веб-сайта.

Были выбраны инструменты для разработки веб-сайта.

При проектировании были разработаны прототипы, а далее созданы макеты дизайна страниц веб-сайта, при опоре на который был реализован сам веб-сайт. Для разработки прототипов и макетов станиц было использована программа Figma, ранее не изученная в ходе учебного процесса.

Были разработаны пользовательские элементы, спецэффекты и логотип, для идентификации веб-сайта, среди миллионов других интернет ресурсов.

Для качественного отображения картинок, на любом устройстве, было решено применять SVG-изображения, также был создан XML-документ для последующей вставки на веб-страницу HTML.

В ходе тестирования была продемонстрирована валидность, кроссбраузерность и адаптивность веб-сайта.

Было создано подробное руководство пользователя для использования веб-сайта.

В пояснительной записке описана реализация поставленных в рамках дипломного проекта ряда задач:

* проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта;
* разработать макет и прототип сайта;
* разработать структуру веб-сайта;
* наполнить сайт информацией по теме;
* протестировать веб-сайт;
* разработать руководство пользователя.

Таким образом была полностью достигнута поставленная цель по разработке веб-сайта, были учтены все требования, все задачи курсового проекта выполнены.

Список использованных источников

1. Документация по Figma [Электронный ресурс]. — Режим доступа: <https://help.figma.com>
2. Аналог «gsz.gob» [Электронный ресурс]. — Режим доступа: https://gsz.gov.by
3. Аналог «rostrud.gov» [Электронный ресурс]. — Режим доступа: https://rostrud.gov.ru
4. Документация по JavaScript/HTML [Электронный ресурс]. — Режим доступа: <https://developer.mozilla.org>
5. Введение в кросс-браузерное тестирование [Электронный ресурс]. — Режим доступа:

<https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Cross_browser_testing/Introduction>