

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.04 – Программная инженерия
Направленность (профиль) образовательной программы Программная инженерия

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

« ____ » _____ 2025 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка программного обеспечения для веб-системы АО «Прииск Соловьёвский»

Исполнитель

студент группы 1105-об

_____ (подпись, дата)

Е. С. Дудник

Руководитель

доцент, канд. физ-мат. н.

_____ (подпись, дата)

В. В. Ерёмина

Консультант

по безопасности и

экологичности

доцент, канд. техн. наук

_____ (подпись, дата)

А. Б. Булгаков

Нормоконтроль

инженер кафедры

_____ (подпись, дата)

В. Н. Адаменко

Благовещенск 2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов

« ____ » _____ 2025 г.

З А Д А Н И Е

К выпускной квалификационной работе студента Дудник Евгения Сергеевича

1. Тема выпускной квалификационной работы: Разработка программного обеспечения для веб-системы АО «Прииск Соловьёвский»

(утверждена приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025

3. Исходные данные к выпускной квалификационной работе: предметная область, перечень литературных источников, отчёт по преддипломной практике.

4. Содержание выпускной квалификационной работы: (перечень подлежащих разработке вопросов): анализ деятельности предприятия, проектирование веб-сайта и телеграм-бота, разработка программного обеспечения.

5. Перечень материалов приложения: (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): техническое задание, макеты основных страниц веб-сайта, листинг телеграм-бота, листинг блока YaMap.

6. Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов): консультант по безопасности и экологичности
Булгаков А.Б., доцент, канд. техн. наук.

7. Дата выдачи задания: 02.02.2025 г.

Руководитель выпускной квалификационной работы: Еремина В. В. доцент, канд. физ-мат. наук.

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению (дата): 02.02.2025 г.

(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит 84 с., 40 рисунков, 3 таблицы, 23 источника, 4 приложения.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА, ПРОЕКТИРОВАНИЕ ТЕЛЕГРАМ-БОТА, ПРОЕКТИРОВАНИЕ ВЕБ-ИНТЕРФЕЙСА, ТЕХНИЧЕСКОЕ ЗАДАНИЕ, UML ДИАГРАММЫ

В качестве объекта исследования данной работы было выбрано предприятие АО «Прииск Соловьёвский».

Цель работы – разработать программное обеспечение для веб-системы АО «Прииск Соловьёвский».

Для достижения цели работы необходимо произвести: составление технического задания на разрабатываемое программное обеспечение, проектирование веб-сайта и телеграм-бота, разработать спроектированное программное обеспечение.

В ходе выполнения работы на этапе проектирования программного обеспечения была выбрана архитектура программного обеспечения, которая была реализована с использованием HTML, CSS, JavaScript, CMS WordPress, Python и PyCharm.

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты:

СанПиН 2.1.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

ГОСТ Р 50948–2001 Средства отображения информации индивидуального пользования

ГОСТ Р 50571.5.54 Электроустановки низковольтные. Выбор и монтаж электрооборудования. Заземляющие устройства, защитные проводники и защитные проводники уравнивания потенциалов

Федеральный закон № 123-ФЗ «Технический регламент о требованиях пожарной безопасности»

Постановлением Правительства РФ от 16.09.2020 № 1479 «Об утверждении Правил противопожарного режима в Российской Федерации»

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

В настоящей бакалаврской работе применяют следующие термины с соответствующими определениями:

АО – акционерное общество

ГОСТ – государственный стандарт

ДФО – Дальневосточный федеральный округ

ИТ – информационные технологии

ПК – персональный компьютер

ЭДО – электронный документооборот

API – интерфейс программирования приложений

CMS – система управления содержимым

CSS – каскадные таблицы стилей

DDoS – распределённая атака типа «отказ в обслуживании»

HTML – язык разметки гипертекста

ИТ – информационные технологии

UI – пользовательский интерфейс

UML – унифицированный язык моделирования

URL – унифицированный указатель ресурса

СОДЕРЖАНИЕ

Введение	9
1 Анализ предметной области	11
1.1 Общие сведения об организации	11
1.2 Обоснование актуальности темы исследования	12
1.3 Предмет исследования	13
1.4 Объект исследования и структура АО «Прииск Соловьевский»	16
1.5 ИТ-инфраструктура АО «Прииск Соловьевский»	18
1.6 Анализ документооборота предприятия	18
1.7 Обзор существующих методов решения типовых задач	19
1.8 Обзор существующих аналогов разрабатываемого программного обеспечения	25
1.9 Формулировка задачи исследования и общей методики ее решения	27
2 Анализ и выбор программных средств	29
2.1 Выбор языка программирования веб-сайта	29
2.1.1 HTML – язык разметки	29
2.1.2 CSS – язык описания внешнего вида документа	30
2.1.3 Java Script – язык создания сценариев	31
2.1.4 CMS – управление контентом	32
2.2 Технологии программирования телеграм-бота	33
3 Этапы разработки информационной системы	34
3.1 Выбор модели жизненного цикла	34
3.2 Описание функциональных требований	35
3.3 Архитектурный проект	38
3.4 Логическая модель данных	41

3.5 Проектирование интерфейса пользователя	42
3.5.1 Этап исследования и анализа	43
3.5.2 Этап создания структуры и прототипирования	43
3.5.3 Этап разработки визуального стиля	46
3.5.4 Этап оптимизация и интерактивность	48
3.5.5 Этап тестирования и доработки	49
3.6 Разработка веб-системы	50
3.6.1 Разработка веб-сайта	50
3.6.2 Разработка телеграм-бота	52
3.7 Реализация интерактивной анимации на веб-сайте	55
3.7.1 Реализация фонового видео на веб-сайте	55
3.7.2 Анимация появления кнопок на веб-сайте	56
3.7.3 Изменение цвета при наведении на объекты	56
3.8 Интеграция блока обратной связи	57
3.9 Внедрение дополнительных модулей	60
3.10 Адаптация сайта под мобильный формат	62
3.11 Обзор веб-системы	65
3.11.1 Обзор веб-сайта priisk-solov.ru	65
3.11.2 Обзор телеграм-бота @priisksolov_bot	66
4 Вопросы информационной безопасности	68
4.1 Область защиты	68
4.2 Виды угроз и методы защиты от них	69
4.3 Мероприятия по защите веб-системы	70
5 Безопасность и экологичность	72
5.1 Организация безопасного рабочего места	72
5.1.1 Требования к персональным компьютерам	72

5.1.2 Помещения и организация рабочих мест	73
5.1.3 Освещение рабочих мест	73
5.1.4 Микроклимат и чистота воздуха	73
5.1.5 Эргономика и оснащение рабочих мест	74
5.1.6 Нормирование труда и перерывы	75
5.2 Экологичность и утилизация оборудования	75
5.3 Чрезвычайные ситуации при работе за компьютером	77
Заключение	80
Библиографические ссылки	82
Библиографический список	83
Приложение А Техническое задание	85
Приложение Б Макеты основных страниц веб-сайта	90
Приложение В Листинг телеграм-бота	93
Приложение Г Листинг блока YaMap	98

ВВЕДЕНИЕ

Современная цифровая трансформация охватывает практически все сферы промышленности, включая золотодобывающую отрасль, где исторически преобладают традиционные формы управления и взаимодействия. В условиях усиливающейся конкуренции, необходимости повышения прозрачности производственных процессов и укрепления корпоративного имиджа, цифровые инструменты становятся неотъемлемой частью стратегического развития предприятий. Особую актуальность приобретает создание специализированных информационных ресурсов, таких как веб-сайты и мессенджер-боты, позволяющих обеспечить эффективную коммуникацию с внешними и внутренними заинтересованными сторонами.

Золотодобывающие предприятия, несмотря на кажущуюся консервативность, остро нуждаются в современных цифровых решениях. Веб-сайт позволяет представить ключевую информацию о предприятии, от истории и миссии до лицензий и производственных мощностей в структурированной и визуально привлекательной форме. Telegram-бот, в свою очередь, способен выполнять функцию интерактивного помощника предоставляя справочные данные и другую информацию о предприятии.

Цель данной выпускной квалификационной работы – разработка программного обеспечения для веб-системы АО «Прииск Соловьевский», а именно разработка веб-сайта и телеграм-бота, соответствующих требованиям и особенностям золотодобывающего предприятия, с учетом современных подходов к пользовательскому интерфейсу, безопасности, доступности и масштабируемости решений.

Для достижения поставленной цели в работе последовательно решаются следующие задачи:

- проведение анализа предметной области, выявление специфики информационных потребностей золотодобывающего предприятия;

- формирование функциональных и технических требований к веб-сайту и телеграм-боту;
- проектирование архитектуры цифровых решений, включая интерфейсные и серверные компоненты;
- реализация веб-сайта с использованием современных веб-технологий;
- разработка телеграм-бота с необходимым функционалом на основе API Telegram и выбранного языка программирования;
- проведение тестирования и отладки разработанных компонентов.

Актуальность данной работы определяется объективной необходимостью цифровизации производственных компаний. Практическая значимость проекта заключается в создании гибкой, настраиваемой платформы, которую можно адаптировать под конкретные нужды предприятия.

Таким образом, настоящая работа представляет собой не только учебно-исследовательский труд, но и прикладной проект, ориентированный на внедрение современных цифровых технологий в традиционный промышленный сектор.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Общие сведения об организации

АО «Прииск Соловьевский» – это одно из ведущих предприятий горнодобывающей отрасли на территории ДВФО, расположенное в Амурской области.

Основным видом деятельности АО «Прииск Соловьевский» является:

- добыча золота дражным и раздельным открытым способом;
- извлечение металла из рудного месторождения.

Предприятие оснащено современным оборудованием и техникой, а также использует передовые технологии.

Основные подразделения:

Добыча: Ответственна за планирование, организацию и осуществление процесса добычи полезных ископаемых. Включает в себя несколько рудников с соответствующими участками.

Переработка и обогащение: Занимается обработкой добытой руды для извлечения полезных компонентов.

Отдел сбыта: Ответственен за реализацию продукции, заключение контрактов и взаимодействие с клиентами.

Финансовый отдел: Ведет бухгалтерский учет, планирует бюджет и занимается финансовой отчетностью.

Отдел кадров: Управляет персоналом, занимается вопросами найма, оплаты труда и кадровым резервом.

Отдел информационных технологий (ИТ): Обеспечивает бесперебойную работу информационных систем, предоставляет ИТ-сервисы, поддерживает безопасность данных, а также занимается внедрением и развитием новых ИТ-решений.

Юридический адрес:

676271, Амурская область, Тындинский район, с. Соловьевск, ул.Советская 47;

ИНН: 2828002272;

Телефон: (41656) 3-44-16

Факс: (41656) 3-47-15

E-mail: kanc@solov.ru.

1.2 Обоснование актуальности темы исследования

В рамках данного квалификационного исследования рассматривается актуальность веб-системы в условиях современной цифровой среде.

Актуальность темы обусловлена следующими факторами:

– Рост влияния интернет-технологий:

– Цифровизация: в современном мире интернет стал неотъемлемой частью жизни, бизнеса и коммуникации. Онлайн-присутствие становится критически важным для любой организации.

– Доступность информации: веб-сайт и телеграм-бот предоставляет потенциальным клиентам, партнерам, инвесторам и другим заинтересованным сторонам легкий доступ к информации о компании, ее деятельности, продуктах и услугах.

– Улучшение коммуникации: онлайн-платформа позволяет установить более тесные связи с клиентами, отвечать на их вопросы, предоставлять поддержку и укреплять доверие к предприятию.

– Повышение конкурентоспособности:

– Виртуальная витрина: веб-сайт и телеграм-бот являются виртуальной витриной предприятия, которую могут увидеть люди по всему миру. Он позволяет представить компанию в выгодном свете, отличить ее от конкурентов и привлечь новых клиентов.

– Улучшение репутации: качественный веб-система с актуальным контентом, удобной навигацией и современным дизайном повышает репутацию предприятия и создает положительный образ в глазах клиентов и партнеров.

- Информационная прозрачность: веб-система может быть использована для публикации информации о проектах, новости и отчетности, что повышает прозрачность деятельности предприятия.

- Расширение рынка сбыта:

- Глобальное присутствие: веб-сайт и телеграм-бот позволяют предприятию выйти на новый уровень и расширить свой рынок сбыта за счет доступа к глобальной аудитории.

- Привлечение инвесторов: онлайн-платформа может служить инструментом привлечения инвесторов, предоставляя им полную и достоверную информацию о деятельности предприятия и его финансовом состоянии.

- Улучшение взаимодействия с клиентами:

- Онлайн-коммуникация: веб-система может включать формы обратной связи, чаты, форумы и другие инструменты для установления более тесных связей с клиентами и партнерами.

- Управление репутацией: веб-сайт и телеграм-бот позволяют отслеживать отзывы клиентов, отвечать на вопросы и управлять репутацией предприятия в онлайн-пространстве.

В целом, разработка веб-системы для АО «Прииск Соловьевский» является актуальной задачей, способной повысить конкурентоспособность предприятия, улучшить его коммуникацию с клиентами и партнерами, создать более прозрачную и эффективную систему управления и расширить рынок сбыта.

1.3 Предмет исследования

Предметом исследования является веб-система.

Веб-система – это совокупность программного обеспечения, серверов, баз данных и сетевой инфраструктуры, работающая в среде интернет и доступная пользователям через веб-браузеры. Веб-системы предоставляют широкие возможности для бизнеса и общения, позволяя создавать информационные ресурсы,

вести торговлю, управлять процессами реализовывать многочисленные другие задачи.[1]

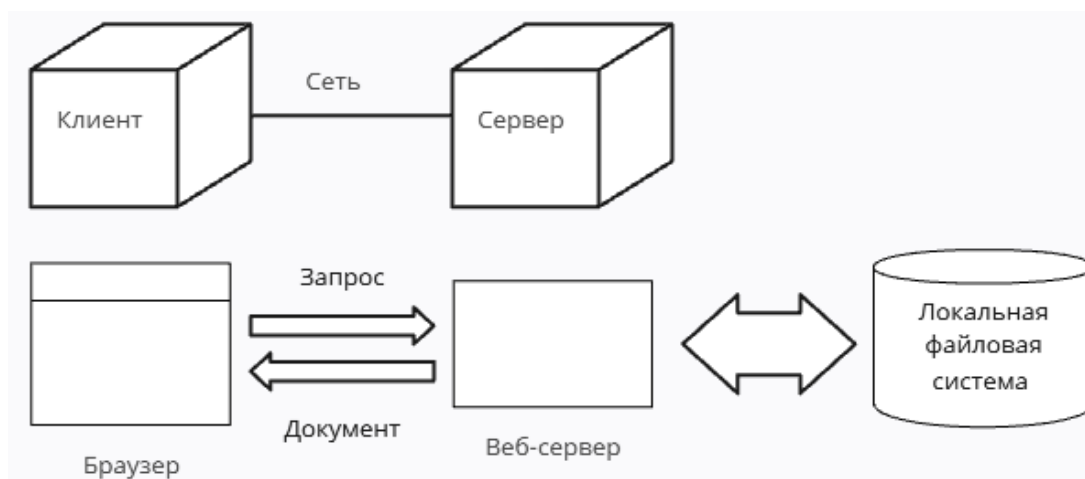


Рисунок 1 – Веб-система

Существует несколько типов веб-систем:

– Веб-сайты:

– Статические: представляют собой набор статических страниц с неизменным контентом. Чаще всего используются для представления информации о компании, продуктах или услугах.

– Динамические: используют скрипты и базы данных для генерации динамического контента, который может меняться в зависимости от запроса пользователя или события. Применяются для создания интерактивных сайтов с возможностью входа в систему, обратной связи, использования форм и т.д.

– Веб-приложения: интерактивные программы, доступные через веб-браузер. Позволяют пользователям выполнять определенные действия: онлайн-покупки, управление аккаунтом, социальное взаимодействие, работа с документами и др.

– Веб-сервисы: программные интерфейсы, позволяющие обмениваться данными между разными веб-приложениями. Например, API для оплаты онлайн, отправки SMS, получения данных с погодных сервисов.

– Мобильные приложения: веб-системы, доступные через мобильные устройства. Позволяют пользователям использовать функции веб-сайта и приложений на своем смартфоне или планшете.

Существует множество видов телеграм-ботов, каждый из которых выполняет определенные функции. Ниже приведены основные типы телеграм-ботов с описанием:

- Информационные боты. Основной функцией данных ботов является предоставление пользователю информации, например новости по интересующей теме, погода, курсы валют.

- Чат-боты/ИИ-боты. Данные боты ведут диалог с пользователем, отвечают на вопросы, могут обучаться.

- Боты для бизнеса и магазинов. Функция данного вида ботов – автоматизация бизнес-процессов (оформление заказов, продажа товаров, чат-менеджеры для поддержки).

- Игровые боты. Основной функцией данных ботов является предоставление игры или участие в игровых процессах (мини-игры, квизы и викторины, геймификация в чатах).

- Утилиты и интеграции. Эти боты используются для соединения телеграм с другими сервисами (уведомления от сайтов или приложений, интеграции с GitHub, Trello, Google Sheets и др.).

Характеристики веб-систем:

Доступность: веб-системы доступны из любой точки мира с доступом к Интернету.

Масштабируемость: веб-системы могут обрабатывать большие объемы данных и пользователей.

Безопасность: важно обеспечить безопасность веб-систем от взлома и несанкционированного доступа к данным.

Обновляемость: веб-системы легко обновлять и добавлять новый функционал.

Экономичность: разработка веб-систем может быть более экономичной, чем разработка традиционного программного обеспечения.

Интерактивность: современные веб-системы предлагают богатый и интерактивный интерфейс для пользователей, что делает их более привлекательными.

В целом, веб-системы являются важной частью современного мира и предлагают широкие возможности для бизнеса, образования, коммуникации и развлечений.

1.4 Объект исследования и структура АО «Прииск Соловьевский»

В рамках данного квалификационного исследования в качестве объекта исследования рассматривается АО «Прииск Соловьевский».

Золотодобывающее предприятие, расположено в Амурской области, Тындинском районе.

АО «Прииск Соловьевский» – крупное золотодобывающее предприятие. Предприятие имеет дивизиональную структуру с четким разделением на производственные и функциональные подразделения.

История одного из старейших приисков современной России – АО «Прииск Соловьевский» – началась в 1868 году. На сегодняшний день АО «Прииск Соловьевский» имеет самый большой дражный флот в стране. В составе предприятия шесть горных карьеров, ведущих разработку месторождений в ДВФО.

Основные подразделения:

Добыча: ответственна за планирование, организацию и осуществление процесса добычи полезных ископаемых. Включает в себя несколько рудников с соответствующими участками.

Переработка и обогащение: занимается обработкой добытой руды для извлечения полезных компонентов.

Отдел сбыта: ответственен за реализацию продукции, заключение контрактов и взаимодействие с клиентами.

Финансовый отдел: ведет бухгалтерский учет, планирует бюджет и занимается финансовой отчетностью.

Отдел кадров: управляет персоналом, занимается вопросами найма, оплаты труда и кадровым резервом.

Отдел информационных технологий (ИТ): обеспечивает бесперебойную работу информационных систем, предоставляет ИТ-сервисы, поддерживает безопасность данных, а также занимается внедрением и развитием новых ИТ-решений.

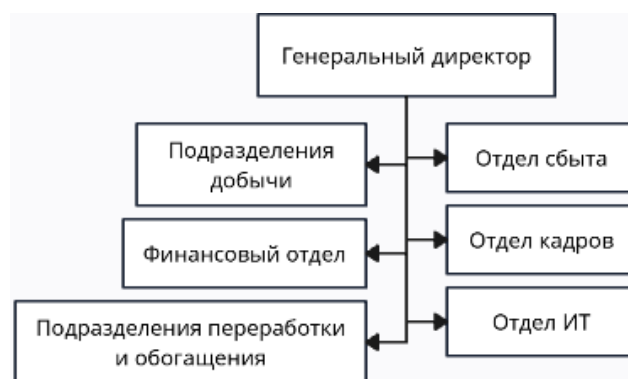


Рисунок 2 – Организационная структура предприятия

Ключевые бизнес-процессы, с которыми тесно связан ИТ-отдел:

– Управление финансами:

– Бухгалтерский учет: автоматизация процессов ввода и обработки финансовых данных, создание отчетности по различным финансовым показателям, взаимодействие с банковскими системами.

– Планирование бюджета: создание и обновление бюджетных моделей, анализ и прогнозирование финансовых показателей.

– Финансовая отчетность: генерация и предоставление финансовой отчетности для внутреннего и внешнего использования.

– Управление персоналом:

– Прием на работу: автоматизированные системы отбора кандидатов, оформление и хранение кадровых документов.

– Оплата труда: расчет и выплата зарплаты, ведение учета отработанного времени.

– Управление кадровым резервом: хранение и обработка информации о кадровом резерве, планирование кадровых передвижений.

1.5 ИТ-инфраструктура АО «Прииск Соловьевский»

Основная ИТ-инфраструктура предприятия включает в себя:

- Серверы: серверная комната расположена в отдельном помещении с контролем температуры и влажности, в комнате обеспечивается бесперебойное питание и охлаждение серверов.
- Функции серверов:
 - файловый сервер (Windows Server 2019) для хранения данных сотрудников;
 - почтовый сервер (Mozilla Thunderbird) для обеспечения почтового обслуживания;
 - база данных (Microsoft SQL Server 2019) для хранения данных системы ERP и других приложений.
- Технологии передачи данных:
 - Ethernet 10/100 Мбит/с;
 - Wi-Fi 802.11ac с доступом в беспроводную сеть для мобильных устройств.
- Программное обеспечение:
 - операционные системы: Microsoft Windows Server 2019, Microsoft Windows 10;
 - офисный пакет: Microsoft Office 365 (Word, Excel, PowerPoint, Outlook);
 - система управления базами данных: Microsoft SQL Server 2019;
 - системы безопасности: антивирусное ПО Kaspersky Lab.

1.6 Анализ документооборота предприятия

Основные типы документов, с которыми взаимодействует предприятие с внешними контрагентами:

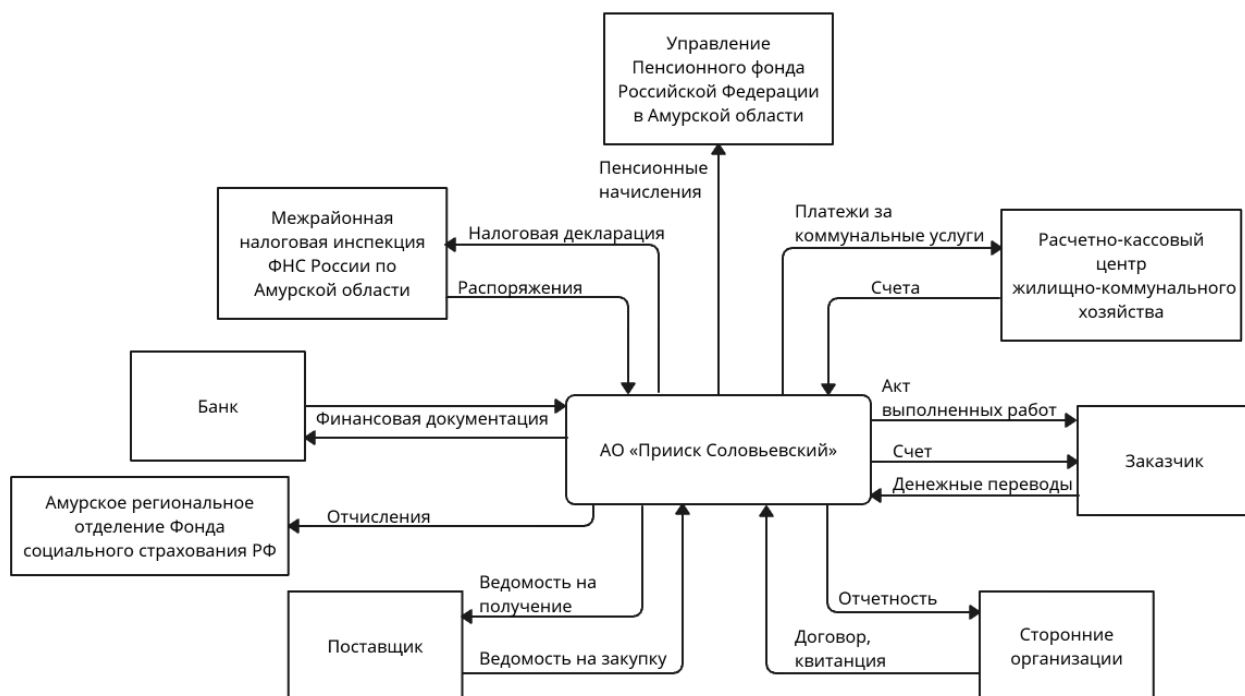


Рисунок 3 – Внешний документооборот предприятия

1.7 Обзор существующих методов решения типовых задач

Разработка программного обеспечения для веб-системы – сложный процесс, который требует учета множества факторов. Чтобы создать эффективный и функциональный веб-сайт, необходимо ознакомиться с методами решения типовых задач, которые успешно применяются на существующих веб-сайтах.

- Представление информации: большинство веб-сайтов золотодобывающих предприятий придерживаются четкой структуры контента.

Обычно используются разделы:

- «О компании» – предоставляет информацию о предприятии, его истории, задачах, ценностях;
- «Продукция» – описывает виды добываемой продукции, её характеристики и свойства;
- «Проекты» – представляет информацию о текущих и будущих проектах предприятия, их этапах реализации;
- «Инвесторам» – содержит финансовую отчетность, информацию о стратегии развития, возможностях инвестирования;

– «Контакты» – предоставляет контактные данные предприятия, форму обратной связи, карту проезда.

Такая структура делает сайт более удобным для пользователей, позволяет им быстро найти нужную информацию.

Использование фотографий, видео, инфографики делает сайт более живым и привлекательным, повышает уровень вовлечения пользователей.

Фотографии могут демонстрировать процессы добычи, оборудование, персонал, ландшафты.

Видео могут показать производственный процесс, рассказать о проектах, предоставить информацию о безопасности и охране окружающей среды.

Инфографика позволяет наглядно представить сложные данные – например, статистику добычи, финансовые показатели, технологические схемы.

Использование интерактивных карт позволяет отобразить местоположение рудников, проектов, офисов и других объектов предприятия.

Пользователи могут увидеть географическое расположение предприятия, что делает информацию более конкретной и понятной.

Предоставление доступа к документам и отчетам (например, финансовая отчетность, регламенты, презентации проектов) повышает уровень прозрачности и доверия к предприятию.

Важным моментом является структурированное хранение и легкий доступ к этим документам для пользователей.

– Взаимодействие с пользователями: формы обратной связи: предоставление возможности отправить вопросы, заявки, отзывы через форму обратной связи на сайте позволяет установить прямую связь с пользователями и оперативно реагировать на их потребности.

– Маркетинг и продвижение: социальные сети: создание профилей предприятия в социальных сетях (ВКонтакте, Телеграм) позволяет установить связь с пользователями и продвигать свои услуги и продукты.

Активное взаимодействие с пользователями в социальных сетях (ответы на вопросы, публикация интересного контента) повышает лояльность аудитории.

Новостная рассылка: отправка новостей предприятия позволяет поддерживать связь с клиентами и инвесторами.

Новостная рассылка может содержать информацию о проектах, финансовых результатах, событиях и т.д.

– Интеграция с другими системами: геоинформационные системы (ГИС): использование ГИС позволяет отобразить геологические данные, местоположение рудников, проектов на интерактивной карте сайта.

Это делает сайт более информативным и наглядным для пользователей, интересующихся географическими данными.

Системы аналитики: интеграция с системами аналитики позволяет собирать и анализировать данные о посещаемости сайта, поведении пользователей, эффективности маркетинговых кампаний.

Аналитические данные могут быть использованы для улучшения сайта, его функциональности, контента и маркетинговых стратегий.

При разработке веб-сайта необходимо учитывать специфику отрасли и требования к информационной безопасности.

Важно провести глубокий анализ конкурентной среды, чтобы определить лучшие практики и создать сайт, который будет отвечать современным требованиям и конкурентным преимуществам.

Необходимо выбрать правильную платформу для разработки сайта (например, WordPress, Drupal, Joomla), чтобы обеспечить его функциональность, безопасность и удобство использования.

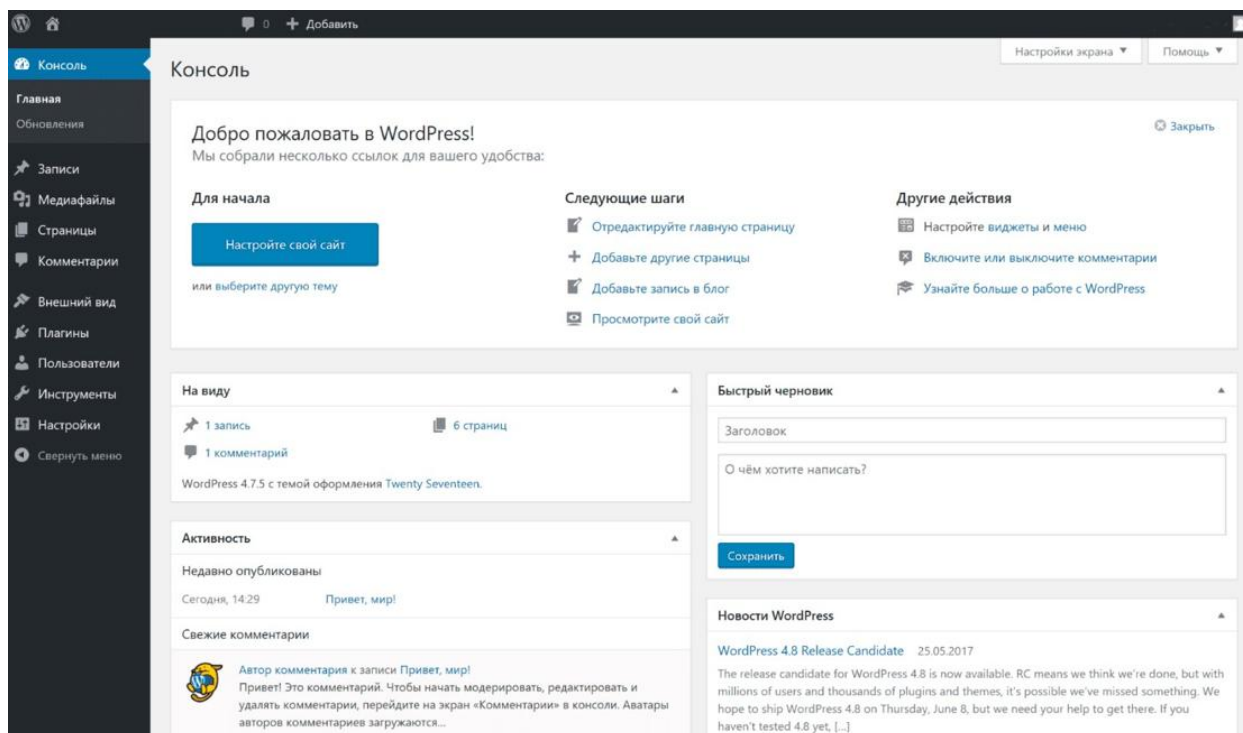


Рисунок 4 – Интерфейс WordPress

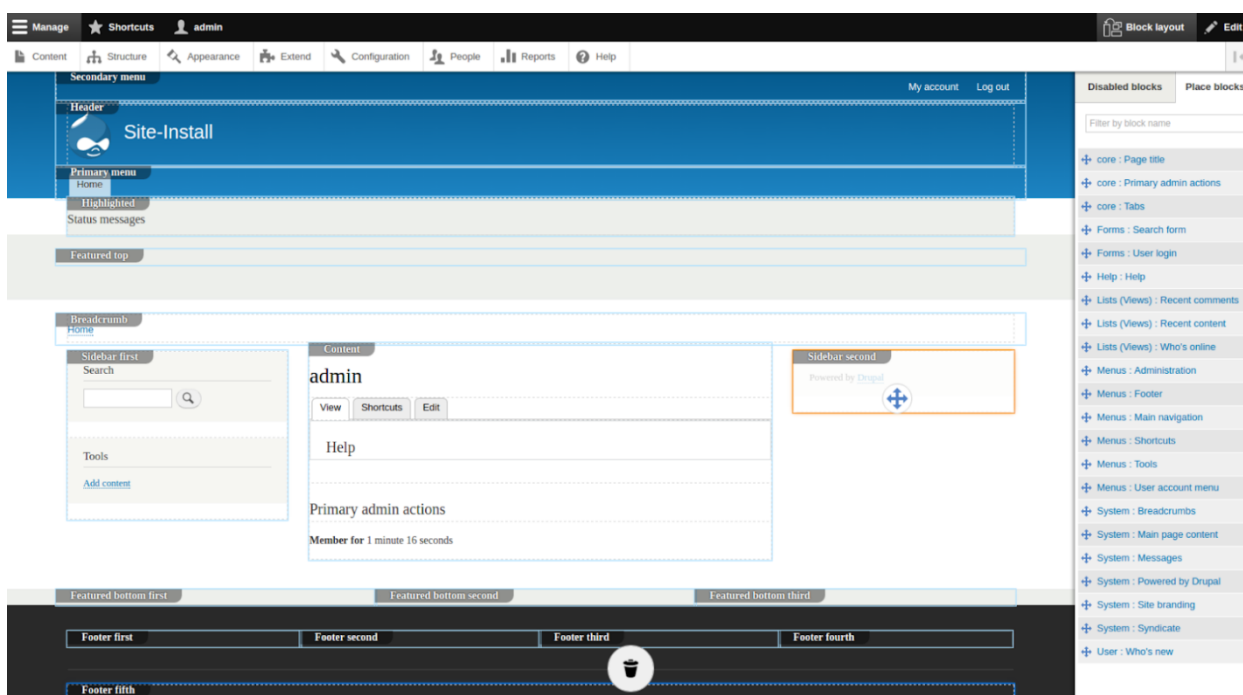


Рисунок 5 – Интерфейс Drupal

Изучение примеров и методов решения типовых задач поможет создать эффективный веб-сайт, который будет отвечать современным требованиям и приносить пользу как предприятию, так и его клиентам, партнерам и инвесторам.

Telegram-боты становятся всё более популярным решением для автоматизации общения. Бот может предоставлять информацию по запросу, направлять документы, уведомлять о новостях, принимать обращения и даже собирать обратную связь.

Telegram-боты – это программные приложения, которые взаимодействуют с пользователями через интерфейс мессенджера телеграм. Они используются для автоматизации общения, выдачи информации, обработки команд, приема данных и т.д. Проектирование телеграм-ботов может осуществляться с использованием различных платформ, фреймворков и инструментов разработки. Рассмотрим основные подходы и сервисы, применяемые при создании ботов.[2]

- Telegram Bot API. Основой для всех телеграм-ботов является Telegram Bot API – официальный интерфейс взаимодействия, предоставляемый разработчикам. С его помощью можно: получать сообщения от пользователей (через polling или webhook), отправлять ответы, обрабатывать команды и события, использовать встроенные клавиатуры, меню и кнопки. API доступен на многих языках программирования, включая Python, JavaScript (Node.js), Java, PHP и другие.[3]

- Языки и библиотеки для разработки:

- Python – один из самых популярных языков для создания телеграм-ботов благодаря своей простоте и множеству готовых библиотек. Основная библиотека:

- python-telegram-bot – удобный и функциональный фреймворк для создания телеграм-ботов. Поддерживает как синхронный, так и асинхронный подход.[4]

- Node.js – используется с библиотеками node-telegram-bot-api, подходит для бэкенд-разработчиков, особенно в случае сложной логики и интеграции с другими сервисами.

- Среды разработки и инструменты:

- PyCharm – популярная среда разработки на Python от компании JetBrains. Поддерживает: автоматическую установку зависимостей,

интеллектуальную подсветку синтаксиса и автодополнение, встроенный терминал и систему контроля версий (Git), виртуальные окружения и управление пакетами.[5]

– Visual Studio Code – альтернатива PyCharm, лёгкая, быстрая, с хорошей поддержкой Python, но менее функциональна в плане крупных проектов.

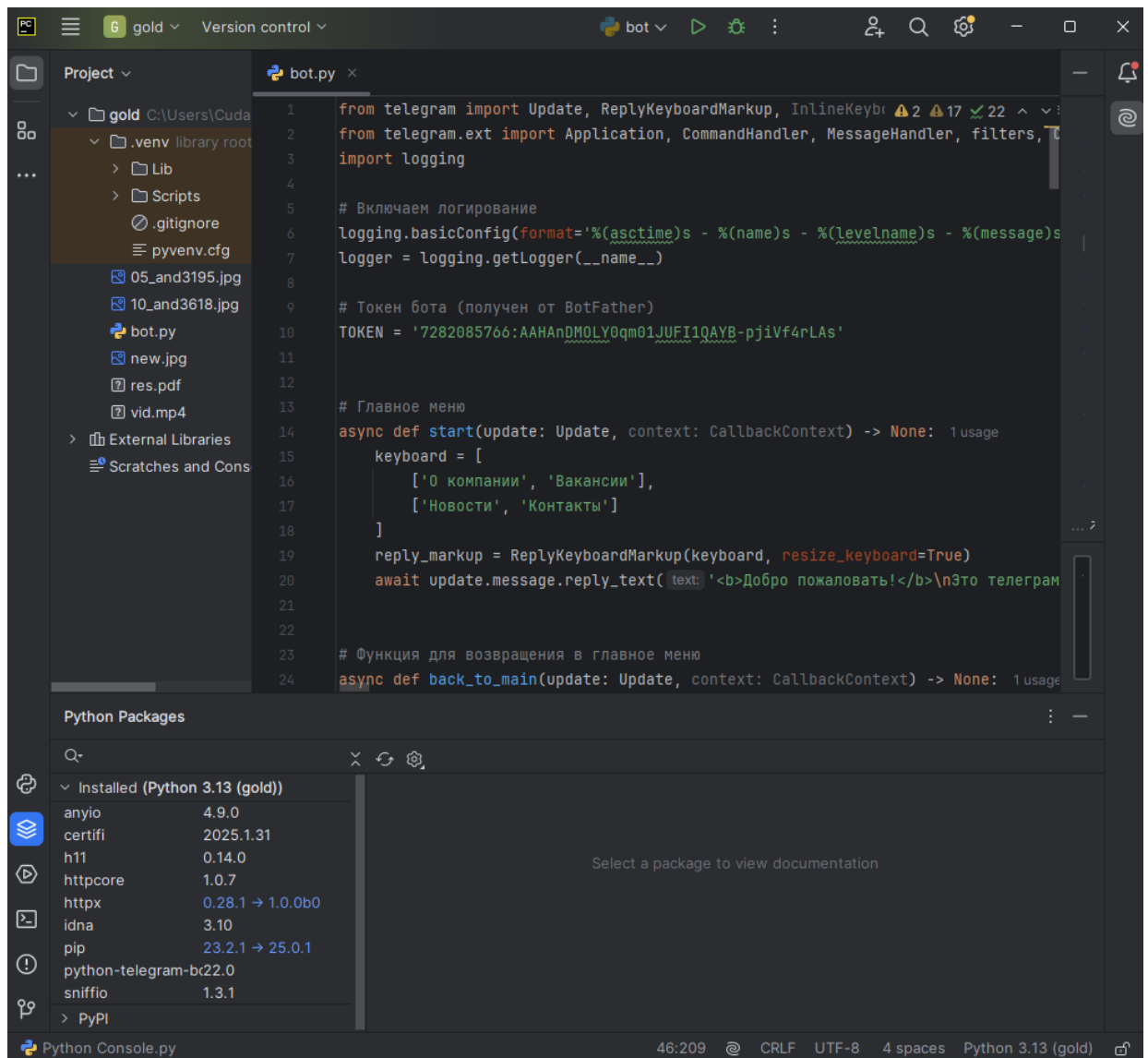


Рисунок 6 – Интерфейс PyCharm

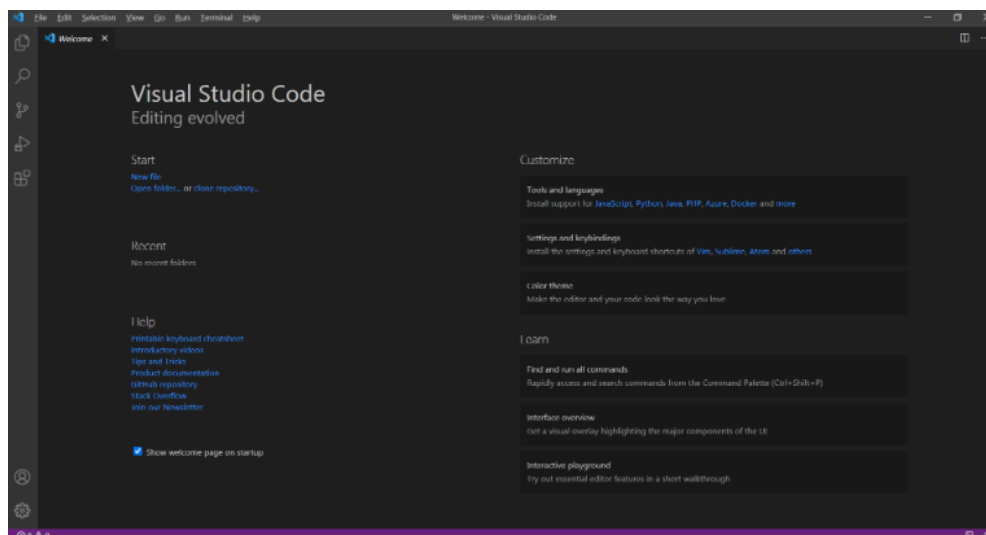


Рисунок 7 – Интерфейс Visual Studio Code

В процессе анализа возможных технологий и инструментов для реализации телеграм-бота были рассмотрены различные языки программирования, библиотеки и среды разработки. Ключевыми критериями выбора стали: доступность документации и обучающих материалов, активное сообщество поддержки, стабильность и надежность, возможность масштабирования функционала. Результат анализа языков программирования представлен в таблице 1.

Таблица 1 – Результат анализа языков программирования

Язык программирования	Документация и обучение	Активность сообщества	Стабильность и надежность	Масштабируемость
Python	Отличная	Очень высокая	Высокая	Хорошая
Node.js (JavaScript)	Хорошая	Высокая	Хорошая	Отличная
Go (Golang)	Средняя	Растущее	Очень высокая	Отличная
Java	Хорошая	Средняя	Очень высокая	Отличная
PHP	Умеренная	Средняя	Средняя	Ограниченная

1.8 Обзор существующих аналогов разрабатываемого программного обеспечения

На данный момент у золотодобывающего предприятия АО «Прииск Соловьевский» существует веб-сайт который может создать конкуренцию

проектируемому веб-сайту. Преимущества проектируемого веб-сайта заключаются в современном дизайне и интерфейсе, повышенной производительности, улучшенной безопасности и интеграцией с новыми инструментами и сервисами. Также обновлена информация о компании.

Предприятие не имеет продуктов которые могут составить конкуренцию разрабатываемому телеграм-боту.

Проектируемый проект – это комплексное решение, сочетающее веб-сайт и телеграм-бота, предназначенное специально для нужд предприятия. Выбор именно проектируемой веб-системы обусловлен рядом значимых преимуществ, которые выгодно отличают его от существующих аналогов:

- индивидуальный подход: в отличие от типовых решений, данный сайт разрабатывается с учётом всех особенностей предприятия АО «Прииск Соловьевский», его специфики, внутренней структуры и потребностей пользователей;

- современные технологии: в разработке используются передовые веб-технологии, что обеспечивает высокую скорость загрузки страниц, адаптивность под мобильные устройства и безупречную работу в различных браузерах;

- интуитивный интерфейс: интерфейс сайта будет простым, понятным и приятным, что снизит порог входа для новых пользователей и повысит удовлетворенность клиентов;

- безопасность данных: проект предусматривает внедрение современных методов защиты информации, защиту от атак и резервное копирование, что особенно важно в условиях цифровой трансформации бизнеса.

Таким образом, создаваемый продукт не только решает актуальные задачи по информатизации предприятия, но и формирует положительный имидж компании как технологически продвинутой и открытой к диалогу.

1.9 Формулировка задачи исследования и общей методики ее решения

Задача исследования: разработать программное обеспечение для веб-системы предприятия, который будет служить эффективным инструментом для представления информации о компании, ее услугах, достижениях и новостях.

Для решения поставленной задачи будет использован следующий подход:

- Изучение предметной области. Проведение анализа целевой аудитории. Выявление потребностей целевой аудитории в информации о предприятии. Определение требований к системе с точки зрения структуры, содержания и функциональности.
- Проектирование системы. Разработка структуры системы, которая будет логичной и удобной для навигации. Разработка дизайна интерфейсов, который будет современным, привлекательным и соответствующим. Разработка системы навигации, которая позволит пользователям легко находить нужную информацию.
- Реализация системы. Написание программного кода сайта с использованием выбранных технологий. Настройка серверной и клиентской частей сайта. Написание программного кода телеграм-бота с использованием выбранных технологий.
- Наполнение контентом. Создание уникального и информативного текстового контента для всех разделов. Подготовка графического контента (фотографии, инфографика, видео) для иллюстрации текстового контента.
- Тестирование и внедрение системы. Проведение функционального тестирования системы для проверки его работоспособности. Проведение нагрузочного тестирования системы для проверки его производительности. Устранение ошибок и внесение необходимых изменений.
- Мониторинг и поддержка системы. Регулярный мониторинг работы сайта. Устранение ошибок и внесение изменений в случае необходимости.

Используемые технологии:

- технологии разработки веб-сайтов: HTML, CSS, JavaScript, PHP и Python;
- среды разработки и инструменты: PyCharm или Visual Studio Code;
- система управления контентом (CMS): WordPress, Joomla или Drupal;

- хостинг: надежный и быстрый хостинг-провайдер;
- аналитика: Яндекс.Метрика.

Ожидаемые результаты:

Современная и информативная веб-система, представляющий предприятие направлен на:

- увеличение узнаваемости бренда и привлечение новых клиентов;
- предоставление актуальной информации о компании, ее услугах и достижениях;
- улучшение коммуникации с целевой аудиторией;
- повышение доверия к предприятию со стороны партнеров и инвесторов;
- увеличение продаж и рост прибыли предприятия.

2 АНАЛИЗ И ВЫБОР ПРОГРАММНЫХ СРЕДСТВ

2.1 Выбор языка программирования веб-сайта

HTML, CSS, JavaScript и CMS – это три основные технологии веб-разработки, которые работают вместе, чтобы создать удобные, красивые, полноценные и функциональные веб-страницы. Они выполняют разные роли и взаимодействуют с пользователями, делая возможным создание динамичного и адаптивного веб-пространства.

- HTML создает структуру и содержимое страницы;
- CSS управляет внешним видом и стилями этой страницы;
- JavaScript добавляет интерактивность и динамическое поведение;
- CMS упрощает управление контентом.

Эти технологии используются вместе, чтобы сделать веб-разработку удобной и эффективной.[6]

2.1.1 HTML – язык разметки

Язык разметки гипертекста HTML (HyperText Markup Language) – это базовый язык разметки для создания структуры веб-страниц. Этот язык служит для описания содержимого веб-страницы и не является языком программирования.

HTML использует «разметку» для аннотирования текста, изображений и другого контента для отображения в веб-браузере. HTML-разметка включает в себя специальные «элементы», такие как `<head>`, `<title>`, `<body>`, `<header>`, `<footer>` и многие другие. Элемент HTML отделяется от остального текста в документе «тегами», которые состоят из имени элемента, окруженного `<` и `>`.

Основными функциями языка HTML являются:

- определяет структуру веб-страницы (заголовки, параграфы, списки, изображения, ссылки, таблицы);
- позволяет вставлять мультимедийные элементы (видео, аудио);
- создает формы для ввода данных;
- использует гиперссылки для навигации между страницами.

Язык HTML имеет несколько основных элементов:

- тег – это основной элемент HTML. Он обычно включает открывающий тег `<tag>` и закрывающий тег `</tag>`. Например, тег `<h1>` используется для заголовков, а `</h1>` завершает этот элемент;
- атрибуты – теги могут иметь дополнительные параметры, называемые атрибутами;
- структура документа – все HTML-документы имеют базовую структуру изображенную на рисунке 8.

```
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <title>Пример</title>
6  </head>
7  <body>
8      <h1>Привет, мир!</h1>
9      <p>Это пример HTML-страницы.</p>
10 </body>
11 </html>
```

Рисунок 8 – Структура документа HTML

Результат выполнения HTML-документа представленного на рисунке 9:

Привет, мир!

Это пример HTML-страницы.

Рисунок 9 – Результат выполнения HTML-документа

HTML – это основа любой веб-страницы, но сам по себе он не задает ни стиль, ни динамику.

2.1.2 CSS – язык описания внешнего вида документа

Каскадные таблицы стилей CSS (Cascading Style Sheets) – это язык, который отвечает за оформление веб-страниц. Он используется для стилизации HTML-элементов, задавая им цвет, размер, положение и другие визуальные параметры.

Основными функциями языка CSS являются:

- определяет цвета, шрифты, отступы и границы;

- делает страницу адаптивной (подстраивает под разные экраны);
- позволяет анимировать элементы;
- разделяет контент и стиль.

Без CSS веб-страницы были бы просто черно-белыми документами без визуального оформления.

В языке CSS используются несколько основных принципов:

- селекторы – при помощи которых можно выбирать HTML-элементы, к которым применяются стили;
- свойства и значения – каждое правило CSS состоит из свойства и значения.

На рисунке 10 показано, что свойство color задает цвет текста, а font-size – размер шрифта;

```
1  h1 {  
2    color: blue;  
3    font-size: 24px;  
4  }
```

Рисунок 10 – Свойства и значения

2.1.3 Java Script – язык создания сценариев

Язык программирования JavaScript – это язык программирования, который добавляет динамичность и интерактивность веб-страницам. Если HTML отвечает за структуру, а CSS за внешний вид, то JavaScript добавляет функциональность. С помощью JavaScript можно обрабатывать действия пользователя (клики, ввод текста и т.д.), изменять содержимое страницы, взаимодействовать с сервером и многое другое.

Основными функциями языка программирования являются:

- управляет событиями (нажатия кнопок, движения мыши);
- динамически изменяет содержимое страницы;
- работает с сервером (запросы, загрузка данных без перезагрузки страницы);
- создает анимации, модальные окна и всплывающие уведомления.

Без JavaScript сайт оставался бы статичным и не реагировал бы на действия пользователя.

Пример кода написанного на языке программирования JavaScript представлен на рисунке 11.

```
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <title>Пример с JavaScript</title>
6  </head>
7  <body>
8      <h1>Кликни на кнопку!</h1>
9      <button id="myButton">Нажми меня!</button>
10     <p id="message"></p>
11
12     <script>
13         document.getElementById("myButton").addEventListener("click", function() {
14             document.getElementById("message").innerText = "Вы кликнули на кнопку!";
15         });
16     </script>
17 </body>
18 </html>
```

Рисунок 11 – Пример кода JavaScript

Результат выполнения кода написанного на языке программирования JavaScript представлен на рисунке 12.

Кликни на кнопку!

Нажми меня!

Вы кликнули на кнопку!

Рисунок 12 – Результат выполнения кода JavaScript

2.1.4 CMS – управление контентом

CMS (Content Management System) – это система управления контентом, которая позволяет создавать, редактировать и управлять веб-сайтами без необходимости писать код вручную.

Основными функциями системы управления контентом являются:

- упрощение создания сайтов без знаний программирования;
- позволяет редактировать контент через удобный интерфейс;
- поддерживает плагины и темы для расширения функционала;
- управляет пользователями, правами доступа и публикациями.

CMS делает управление сайтом доступным даже для пользователей без технических знаний.

Популярные системы управления контентом:

- WordPress – самая популярная CMS, удобная для блогов, корпоративных сайтов, интернет-магазинов;
- Joomla! – мощная система для сложных проектов;
- Drupal – подходит для масштабных сайтов с высокой нагрузкой;
- 1С-Битрикс – популярная в России CMS для бизнеса.

2.2 Технологии программирования телеграм-бота

По результатам анализа было принято решение использовать язык программирования Python. Это обусловлено следующими преимуществами:

- простота и лаконичность синтаксиса, что сокращает время на разработку;
- широкий выбор готовых библиотек для работы с Telegram API (python-telegram-bot, aiogram);
- легкость интеграции с файловыми системами и API других систем.

В качестве среды разработки выбран PyCharm – профессиональный инструмент для написания кода на Python. Преимущества использования PyCharm:

- высокий уровень автоматизации (автозаполнение, отладка, управление зависимостями);
- удобная работа с виртуальными окружениями и системами контроля версий (Git);
- наличие встроенных инструментов тестирования и отладки.[7]

Выбор данной технологии обеспечит: гибкость в проектировании архитектуры телеграм-бота, стабильность и масштабируемость при увеличении числа пользователей, простоту в поддержке и доработке функционала в будущем.

Таким образом, сочетание языка Python и среды разработки PyCharm оптимально подходит для реализации телеграм-бота. Это решение позволяет создать надежный, быстрый и удобный инструмент для взаимодействия с пользователями и автоматизации рутинных процессов.

3 ЭТАПЫ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Выбор модели жизненного цикла

Перед началом разработки программного обеспечения важно определить подходящую модель жизненного цикла (ЖЦ) – набора этапов, через которые проходит проект от идеи до завершения. Существует несколько основных моделей ЖЦ, каждая из которых применяется в зависимости от специфики проекта, требований заказчика, бюджета, сроков и других факторов.

Классическая водопадная модель предполагает строгое следование этапам – от анализа требований до внедрения и поддержки, без возврата к предыдущим шагам. Она подходит для проектов с чётко определёнными задачами и минимальными рисками изменений.

Каскадная модель с обратной связью допускает возврат на один или несколько предыдущих этапов, что делает её более гибкой по сравнению с классическим "водопадом".

Инкрементная модель реализует продукт по частям – каждая версия расширяет функциональность предыдущей. Это позволяет быстро выводить минимально работоспособные версии и постепенно улучшать их.

Итеративная модель схожа с инкрементной, но делает акцент на повторении циклов – итераций. Каждая итерация включает анализ, проектирование, реализацию и тестирование части функционала, с возможностью переосмысления архитектуры по мере развития проекта.

Также распространены спиральная модель, ориентированная на риск-менеджмент, и гибкие методологии, предполагающие постоянное взаимодействие с заказчиком и быструю адаптацию к изменениям.

Для проектирования веб-сайта и телеграм-бота предприятия была выбрана итеративная модель жизненного цикла проектирования программного обеспечения. Данная модель предполагает реализацию проекта поэтапно, с постепенным добавлением функциональности (рисунок 4). Это позволяет:

- быстро развернуть базовую версию продукта;
- тестировать каждую версию независимо;
- получать обратную связь от заказчика и вносить правки на ранних стадиях;
- повысить гибкость при изменениях требований.

Процесс проектирования включает следующие фазы: анализ требований, написание технического задания, создание логической модели данных, создание архитектурного проекта, проектирование баз данных, выбор средств реализации проекта.

Инкременты – расширения по мере необходимости.

Такой подход особенно эффективен для взаимодействия с пользователями через мессенджеры и веб-интерфейсы, где важна удобная навигация, стабильная работа и своевременное обновление контента.

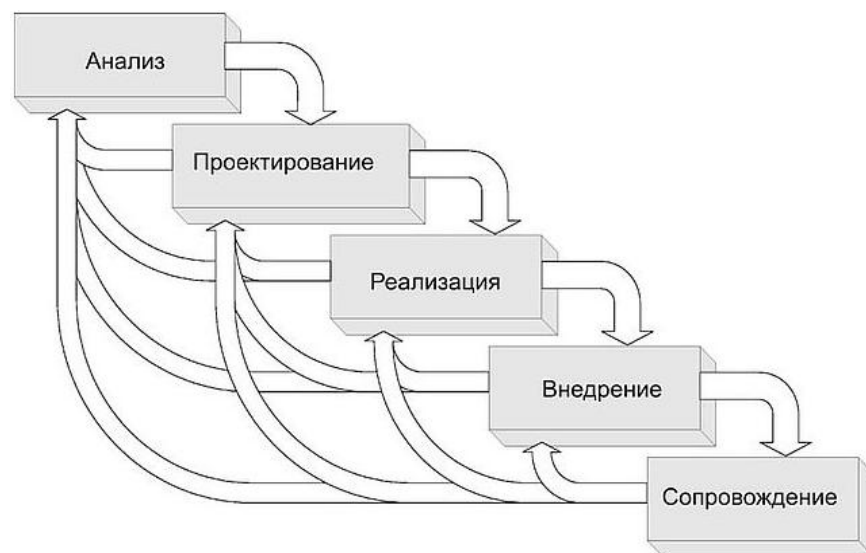


Рисунок 14 – Итеративная модель жизненного цикла

3.2 Описание функциональных требований

Разрабатываемая веб-система предназначена для предоставления в сети Интернет всей необходимой информации для работников и инвесторов, а также для цифровизации, улучшения коммуникации, повышения конкурентоспособности, информационной прозрачности и расширения рынка сбыта.

Функционал веб-сайта – это совокупность возможностей и инструментов, которые определяют, как работает сайт и какие задачи он выполняет.

Основной функционал веб-сайта:

- навигация и интерфейс – удобное меню, поиск, кнопки, ссылки, которые позволяют пользователям легко перемещаться по сайту;
- контент и мультимедиа – отображение текстов, изображений, видео, анимаций и других элементов;
- интерактивность – формы обратной связи, комментарии, лайки, подписки, калькуляторы;
- интеграции с внешними сервисами – соцсети, платежные системы, карты, CRM;
- администрирование и безопасность – защита данных, управление пользователями, аналитика посещаемости.

Функциональные требования к веб-сайту золотодобывающего предприятия АО «Прииск Соловьевский» являются:

- предоставление информации о предприятии;
- форма для обратной связи;
- вопрос-ответ;
- разделение прав доступа;
- создание, редактирование и удаление новостей.

В ходе анализа существующих веб-сайтов были выявлены следующие требования к интерфейсу веб-сайта:

- Удобство и простота использования:
 - интуитивно понятная навигация;
 - единый стиль оформления и логичная структура страниц;
 - быстрый доступ к основным разделам и функциям.
- Адаптивность и кросс-браузерность:
 - корректное отображение на мобильных устройствах, планшетах и ПК;
 - поддержка всех популярных браузеров (Chrome, Safari, Firefox, Microsoft Edge).
- Визуальная привлекательность и дизайн:

- современный и эстетичный интерфейс;
 - читаемый шрифт, удобные цветовые решения;
 - реализация анимаций;
 - использование качественных изображений и графики.
- Интерактивность:
- формы обратной связи;
 - модуль с геолокацией предприятия;
 - ссылки на социальные сети предприятия.

Проектирование современной веб-системы требует четкой структуризации всех компонентов и функций, обеспечивающих удобство взаимодействия пользователей с ресурсом.

На веб-сайте необходимо создать следующие страницы:

- главная страница сайта – основная информационная страница с основной информацией и ссылками на другие страницы, также данная страница содержит анимационный фон;
- страница «Контакты» – расположена вся контактная информация предприятия и модуль с картой;
- страница «Политика конфиденциальности» – политика конфиденциальности при отправки личных данных в форме обратной связи;
- страница «Новости» – страница где расположены новостные блоки;
- страница «О нас» – данная страница включает в себя ссылки на подстраницы;

Меню должно содержать следующие элементы:

- «Контакты»;
- «Новости»;
- «О нас».

Все разделы веб-сайта должны быть доступны всем пользователям.

В телеграм-боте необходимо создать следующие страницы:

- главная страница;

- страница «О компании» – ссылки на подстраницы;
- страница «Новости» – информационное сообщение с пятью последними новостями предприятия;

- страница «Вакансии» – ссылки на подстраницы;
- страница «Контакты» – контактная информация;

Главное меню должно содержать следующие элементы:

- страница «О компании»;
- страница «Новости»;
- страница «Вакансии»;
- страница «Контакты».

Все разделы должны быть доступны всем пользователям.

На каждом разделе, кроме главного меню, должна присутствовать кнопка «Назад» для возвращения к предыдущему разделу.

Основываясь на требованиях заказчика к веб-системе было составлено техническое задание (Приложение А). Техническое задание требуется для определения очереди разработки, приемки и ввода в действие веб-системы.

3.3 Архитектурный проект

В процессе проектирования программного обеспечения архитектурное моделирование играет ключевую роль, обеспечивая наглядное представление структуры и поведения системы. Данный раздел посвящён архитектурному проекту, в рамках которого представлены различные UML-диаграммы, отражающие важнейшие аспекты взаимодействия компонентов, пользователей и процессов. Каждая из диаграмм фокусируется на определенной стороне архитектуры разрабатываемого решения: от внешнего взаимодействия с системой до внутренней организации модулей и возможных состояний объектов.

Диаграмма вариантов использования (или Use Case Diagram) – это один из видов UML-диаграмм, который показывает, как внешние пользователи (акторы) взаимодействуют с системой через ее функциональность. На рисунке 15 представлена диаграмма вариантов использования телеграм-бота.



Рисунок 15 – Диаграмма вариантов использования веб-сайта

Диаграмма активностей – это диаграмма в UML, которая показывает последовательность действий, выполняемых в системе или процессе. Она помогает понять что, когда и в каком порядке происходит, и кто выполняет действия. На рисунке 16 описан сценарий запроса документации.



Рисунок 16 – Диаграмма активностей на странице веб-сайта

Диаграмма пакетов – это структурная диаграмма UML, которая показывает, как организованы классы, модули или компоненты в логические группы, называемые пакетами. Структура модулей сайта по разделам представлена на рисунке 17.

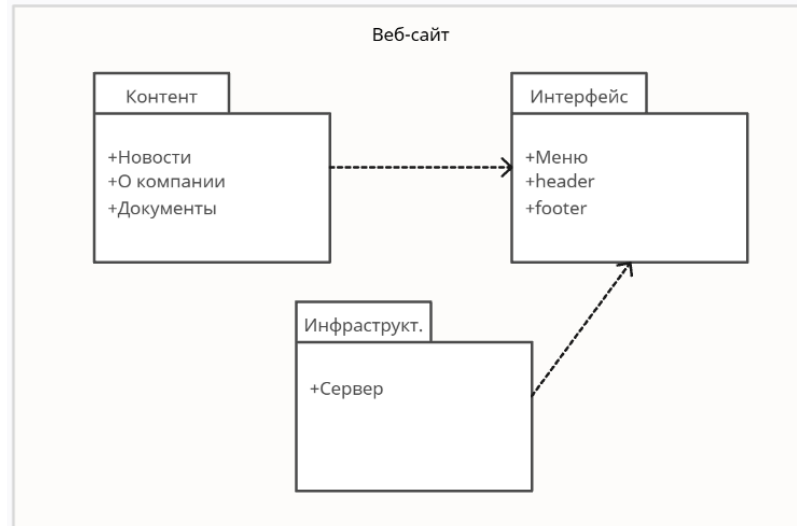


Рисунок 17 – Диаграмма пакетов веб-сайта

Диаграмма состояний – это поведенческая диаграмма UML, которая показывает, в какие состояния может переходить объект, что вызывает эти переходы, и как он себя ведёт в каждом состоянии. Диаграмма состояний для элемента «Форма обратной связи» представлена на рисунке 18.

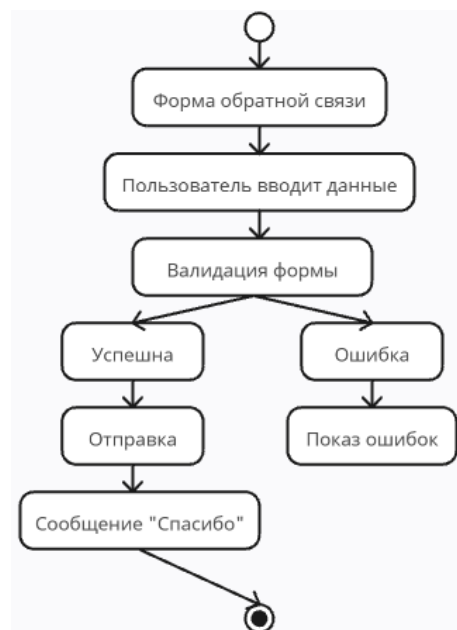


Рисунок 18 – Диаграмма состояний формы обратной связи

Использование UML-диаграмм позволяет не только формализовать архитектуру, но и облегчить коммуникацию между разработчиками, аналитиками и другими заинтересованными сторонами. Диаграммы, представленные в этом разделе, дают целостное представление о функционировании программного обеспечения, демонстрируя их логику, структуру и сценарии поведения в различных ситуациях.

3.4 Логическая модель данных

Разработка веб-сайта и телеграм-бота требует четкого понимания взаимосвязей между элементами системы, что делает построение логической модели данных ключевым этапом проектирования. Эта модель формируется на основе анализа пользовательских запросов и ожидаемых результатов работы системы.

В качестве выходных данных рассматриваются интерфейсные и функциональные компоненты, отображаемые как на веб-платформе, так и в телеграм-боте. Архитектурная реализация осуществляется через взаимодействие front-end и back-end компонентов, а также модулей, обеспечивающих работу бота. Управление данными, сопровождение процессов тестирования и обновления представляют собой управляющие воздействия. В результате проведенного анализа была создана функциональная модель системы, представленная с использованием нотации IDEF0 (рисунок 19) и последующей декомпозицией верхнего уровня (рисунок 20), что позволяет наглядно отразить структуру и логику функционирования всех компонентов.

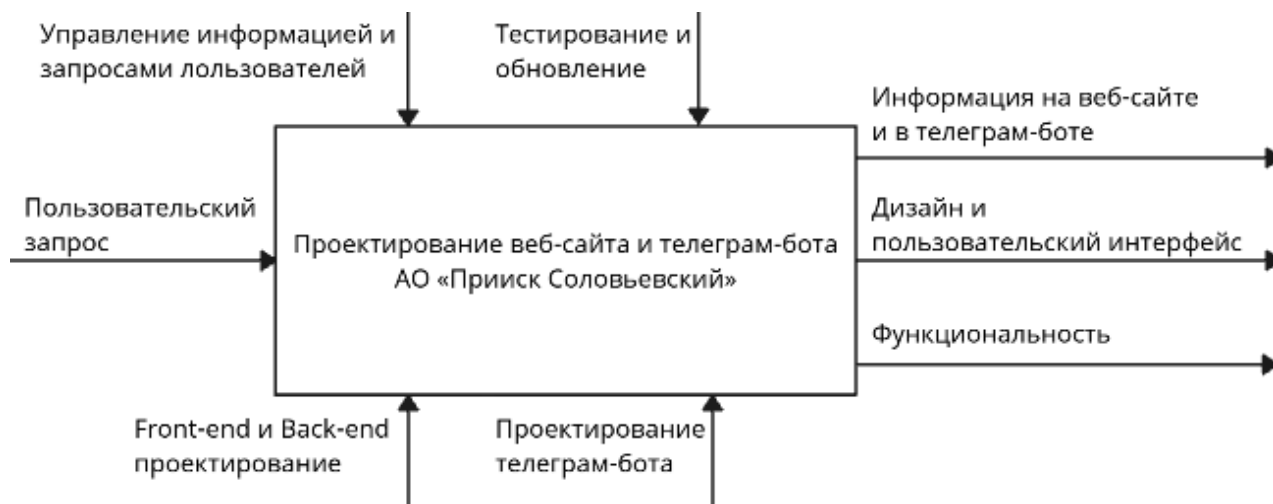


Рисунок 19 – Нотация IDEF0

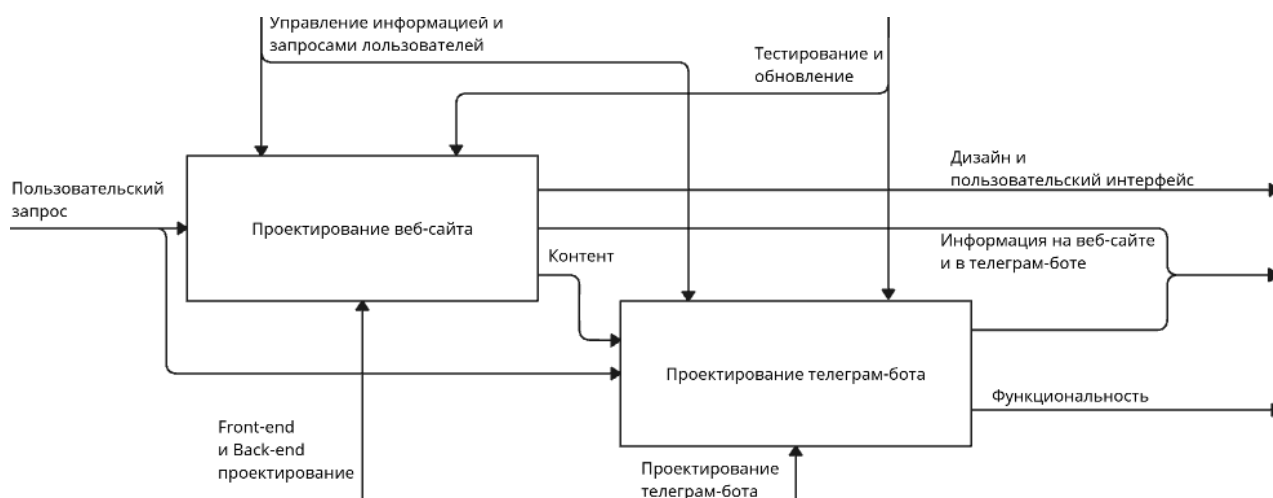


Рисунок 20 – Декомпозиция первого уровня

3.5 Проектирование интерфейса пользователя

Проектирование пользовательского интерфейса (UI) веб-системы – это процесс создания удобной, интуитивно понятной и эстетически привлекательной среды для пользователей. Включает в себя разработку навигации, элементов взаимодействия и визуального оформления.

Проектирование пользовательского интерфейса веб-системы можно разделить на следующие этапы:

- этап исследования и анализа;
- этап создания структуры и прототипирования;
- этап разработки визуального стиля;
- этап оптимизация и интерактивность;

– этап тестирования и доработки.

Создание интерфейса пользователя – это процесс, который требует анализа, проектирования, тестирования и постоянных улучшений. Чем удобнее интерфейс, тем больше пользователей останутся довольны и вернутся снова.

3.5.1 Этап исследования и анализа

Перед тем как приступить к дизайну интерфейса, важно понять аудиторию и цели веб-системы. Разрабатываемый веб-сайт и телеграм-бот является информационной системой, на котором будет представлена информация о компании и её проектах, также веб-система будет использоваться для привлечения инвесторов через демонстрацию финансовых отчетов и достижений. В результате веб-сайт и телеграм-бот преимущественно рассчитан на инвесторов, партнеров, журналистов и работников предприятия.

Веб-система золотодобывающей компании должен быть надежным, удобным и информативным.

3.5.2 Этап создания структуры и прототипирования

На этом этапе создается логическая структура веб-сайта, а также телеграм-бота.

Карта сайта определяет, какие страницы должны присутствовать и как они связаны между собой. Карта разрабатываемого веб-сайта:

- главная страница;
- о нас;
- новости;
- контакты.

Подробная схема карты веб-сайта представлена на рисунке 21.

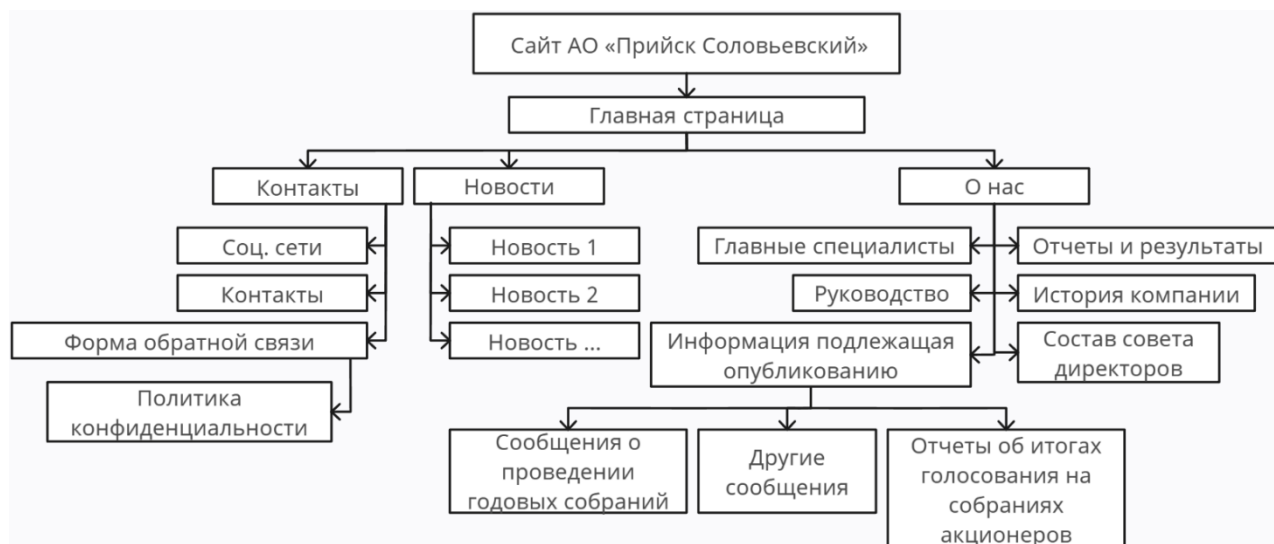


Рисунок 21 – Карты веб-сайта

Для верстки сайта предварительно будет создан макет каждой страницы. В качестве основного инструмента для прототипирования выбран Mockflow. Mockflow – бесплатный онлайн-сервис, предназначенный для быстрого создания прототипов и визуализации пользовательского интерфейса (UI/UX-дизайна). Mockflow позволяет собирать макеты из готовых блоков, что значительно ускоряет процесс проектирования и обеспечивает наглядность.

Визуальное представление каждой страницы помогает более точно определить структуру контента, расположение элементов и общий стиль сайта до начала этапа верстки. Это снижает вероятность ошибок и доработок в будущем, а также упрощает коммуникацию между дизайнером и разработчиком.[8]

Макет страницы «О нас» представлен на рисунке 22.



Рисунок 22 – Макет страницы «О нас»

Макеты всех страниц представлены в приложении Б

Структура телеграм-бота, включает основные разделы и сценарии взаимодействия пользователя с интерфейсом. В отличие от классического веб-сайта, телеграм-бот использует текстовое меню и кнопки для навигации, реализуемые с помощью ReplyKeyboardMarkup.

Основные разделы телеграм-бота АО «Прииск Соловьёвский» включают:

- «О компании»;
- «Вакансии»;
- «Новости»;
- «Контакты».

Подробная карта разделов телеграм-бота АО «Прииск Соловьёвский» изображена на рисунке 23.



Рисунок 23 – Карты телеграм-бота

Для проектирования структуры меню был использован принцип блочного прототипирования, где каждая кнопка запускает отдельную функцию с текстом, медиафайлами или вложениями.

Прототип бота реализуется с использованием языка Python и библиотеки python-telegram-bot, позволяющей обрабатывать команды и текстовые сообщения. В качестве основы использовалась асинхронная модель взаимодействия, обеспечивающая стабильную работу даже при высокой нагрузке.

Визуальный прототип телеграм-бота создается не в виде графического макета, как в веб-дизайне, а в виде иерархии кнопок и сценариев (Flow-chart). Однако логика проектирования схожа с веб-картой: каждому действию пользователя соответствует определённый экран (сообщение), а кнопки играют роль ссылок между ними.

3.5.3 Этап разработки визуального стиля

На этом этапе создается дизайн, который соответствует бренду и удобен для пользователя. На данном этапе проектирование интерфейса пользователя подбираются шрифты и выбираются кнопки, формы, иконки.

Также на этом этапе проектирования выбирается цветовая палитра. Цветовая палитра – это набор цветов, которые используются на веб-сайте для оформления его внешнего вида.

В цветовую палитру сайта входит:

- основной цвет (главный акцентный цвет, который чаще всего используется для кнопок, ссылок, важных элементов);

- дополнительный цвет (поддерживающий цвет, который дополняет основной и используется для второстепенных акцентов);
- фоновый цвет (цвет, который используется для основного фона страниц);
- цвет текста (цвет основного текста, подзаголовков, ссылок и других текстовых элементов).

Проанализировав деятельности предприятия были сформированы следующие требования:

- цвет имеет хороший контраст между текстом и фоном;
- цвета должны быть удобны для восприятия на разных устройствах и экранах;
- цвет должен отражать сферу деятельности.

Сформированная цветовая панель представлена на рисунке 24.



Рисунок 24 – Цветовая палитра

Также на данном этапе была проведена работа по оформлению визуального образа телеграм-бота АО «Прииск Соловьёвский», обеспечивающего удобное взаимодействие с пользователем.

Были определены следующие элементы интерфейса:

- типографика (подбор удобочитаемых шрифтов);
- кнопки и клавиатуры – в виде интуитивно понятных блоков, структурирующих доступ к разделам;
- иконки – сопровождающие текстовые элементы, упрощающие восприятие информации (например: 🏠, 📞, 📧 и др.);
- структура меню – реализована через клавиатуру с разбивкой по логическим разделам (например: «О компании», «Вакансии», «Контакты», «Новости» и др.).

Пример реализации визуального подхода можно увидеть в меню бота, где используется адаптивная клавиатура с иконками и заголовками, а также форматирование текста (жирный шрифт, эмодзи) для лучшего акцента на важной информации (рисунок 25). Все элементы адаптированы под интерфейс телеграм, что делает навигацию и восприятие информации максимально комфортными.

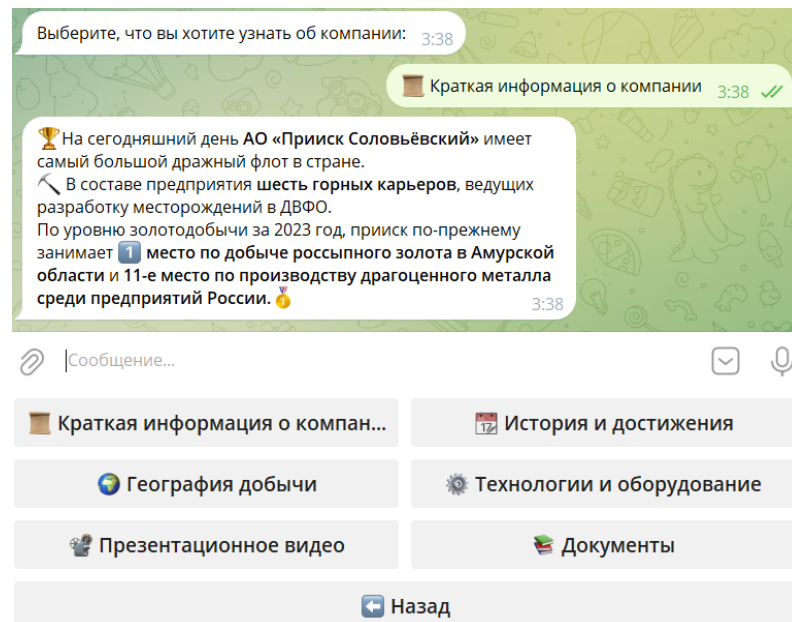


Рисунок 25 – Меню телеграм-бота

В результате, визуальный стиль бота стал не только отражением корпоративного имиджа предприятия, но и эффективным инструментом для удобного взаимодействия с пользователями.

3.5.4 Этап оптимизация и интерактивность

На данном этапе производится оптимизация навигации, которая помогает минимизировать количество кликов до целевого действия и создается простая иерархия разделов.

Далее осуществляется проектирование анимации и эффектов. Анимация делает сайт более живым, но не должна мешать пользователю.

Также на этом этапе добавляются интерактивные элементы, такие как кнопки, всплывающие окна и чат-поддержки.

Особое внимание уделяется удобству взаимодействия с сайтом: продумываются сценарии поведения элементов при наведении, нажатии и прокрутке страницы.

Проводится тестирование интерактивных элементов на разных устройствах для обеспечения корректной работы и быстрой загрузки.

Все решения принимаются с учетом пользовательского опыта (UX) и общей концепции дизайна проекта.

3.5.5 Этап тестирования и доработки

Это финальная часть проектирования, в которой оценивается, насколько удобно, понятно и привлекательно система выглядит и работает для пользователей. Этот этап позволяет убедиться, что всё функционирует корректно, удобно и безопасно для пользователей.

На этом этапе проводится несколько видов работ:

- тестирование разных версий дизайна, чтобы выбрать наиболее эффективную;
- тестирование на реальных пользователях;
- оптимизация изображений.

Тестирование пользовательского интерфейса проводится по следующим пунктам:

- визуальное оформление (единый стиль и цветовая палитра на всех страницах, все шрифты читаемы и достаточно контрастны, умеренное количество анимаций и эффектов);
- адаптивность и отзывчивость (интерфейс хорошо отображается на разных устройствах, адаптируются размеры шрифтов, кнопок, блоков);
- навигация (меню логичное, удобный доступ к важным разделам);
- интерактивные элементы (кнопки кликабельны, имеют визуальную обратную связь при наведении, формы удобно заполняются, анимации не мешают восприятию информации);

- иконки и изображения (все иконки понятны и дополняют смысл, изображения загружаются корректно);
- контент (нет опечаток и грамматических ошибок, текст не обрезается и не наезжает на другие элементы).

3.6 Разработка веб-системы

3.6.1 Разработка веб-сайта

Первым этапом в разработке веб-сайта является выбор хостинга.

Хостинг – это услуга по размещению файлов и данных веб-сайта на сервере, который подключен к интернету. Благодаря хостингу сайт становится доступным для пользователей в любое время и из любой точки мира.

В ходе анализа различных сервисов предоставляющих услуги хостинга, был выбран сервис «timeweb» – российский хостинг-провайдер, отличающийся стабильным функционированием, безопасностью, быстродействием и множеством предложений для размещений сайтов. Сервис имеет структурированную панель администрирования.

Особенности данного сервиса:

- быстрая и качественная поддержка;
- бесперебойная работа серверов на уровне 99.98 %;
- удобная и понятная панель управления.

Также в сервисе «timeweb» было приобретено и привязано к веб-сайту доменное имя «priisk-solov.ru»

Разработка веб-сайта – это процесс создания сайта, который включает несколько этапов, от идеи до запуска. Разработка начинается с этапа планирования, который включает в себя: определение цели сайта, анализ целевой аудитории, определение структуры сайта.

В процессе анализа платформ для разработки веб-сайтов была выбрана платформа CMS WordPress. WordPress – это бесплатная и открытая платформа по разработке веб-сайтов. Платформа WordPress представляет собой мощную систему управления контентом (CMS), которая обеспечивает гибкость, простоту

использования и масштабируемость проекта. CMS делает управление сайтом доступным даже для пользователей без технических знаний. У выбранной платформы интуитивно понятный интерфейс и редактор который позволяет создавать сложные макеты. Интерфейс платформы CMS WordPress представлен на рисунке 5.

Процесс установки CMS WordPress осуществляется на сайте хостинг-провайдера. После установки происходит базовая настройка (выбор языка, установка названия сайта и слогана, создание администратора).

Далее через меню «Внешний вид → Темы» выбирается тема оформления.

После того как установили CMS и выбрали тему, происходит процесс установки плагинов. Плагины – это расширения, добавляющие функциональность сайту. Плагины устанавливаются через меню «Плагины → Добавить новый».

Для данной веб-системы были выбраны следующие плагины:

- Elementor – визуальный конструктор страниц (рисунок 26);
- Yoast SEO – оптимизация сайта для поисковых систем;
- Contact Form 7 – создание форм обратной связи;
- UpdraftPlus – резервное копирование сайта.

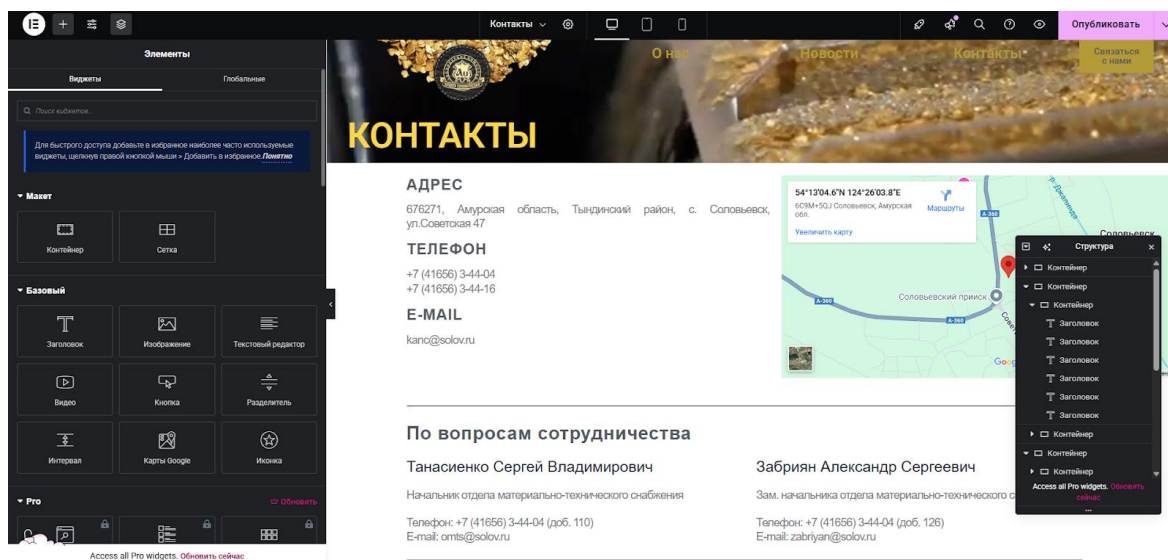


Рисунок 26 – Интерфейс Elementor

После выбора и установки плагинов создаются основные страницы сайта:

- главная страница;

- о нас;
- новости;
- контакты.

Также настраиваются меню и виджеты для удобной навигации: шапка сайта (header), подвал сайта (footer).

Важно настроить параметры сайта, такие как:

- постоянные ссылки (URL-структура);
- часовой пояс и формат времени;
- настройка отображения главной страницы.

Также необходимо подключить аналитику Яндекс Метрика. Яндекс Метрика – это бесплатный сервис веб-аналитики от компании Яндекс, предназначенный для сбора статистики и анализа поведения пользователей на сайте.

Основные функции Яндекс Метрики: счетчик посещаемости, карта кликов (визуализация, которая показывает, куда чаще всего кликают посетители) и отчеты в реальном времени (можно видеть, что происходит на сайте прямо сейчас: сколько людей онлайн, с каких устройств, на каких страницах находятся).

После наполнения контентом и проверки всех разделов сайт можно опубликовать. На этом этапе также важно:

- протестировать адаптивность и отображение на разных устройствах;
- проверить корректность всех ссылок и форм.

Разработка веб-сайта с помощью WordPress – это интуитивно понятный и доступный процесс, не требующий глубоких знаний программирования. Благодаря широкому выбору тем и плагинов можно быстро создать сайт, соответствующий современным требованиям дизайна и функциональности.

3.6.2 Разработка телеграм-бота

Первым этапом в создании телеграм-бота является регистрация и получение уникального токена доступа. Для этого используется специальный официальный бот в телеграм под названием BotFather. С его помощью можно создать нового бота,

задать ему имя, описание и получить токен, необходимый для работы с Telegram API.

После получения токена необходимо перейти к процессу разработки. Для написания бота используется язык программирования Python, благодаря своей простоте, универсальности и наличию мощной библиотеки `python-telegram-bot`, предоставляющей готовый функционал для работы с Telegram API.

Средой разработки выбран PyCharm - профессиональная IDE от компании JetBrains. Она обладает широкими возможностями для разработки на Python: интеллектуальная подсветка синтаксиса, отладка кода, управление зависимостями и встроенный терминал.

Перед началом работы в PyCharm необходимо установить библиотеку `python-telegram-bot`. Это можно сделать с помощью команды в терминале, команда для установки библиотеки представлена на рисунке 27.

```
1  pip install python-telegram-bot --upgrade
2
```

Рисунок 27 – Команда для установки библиотеки `python-telegram-bot`

Этап разработка архитектуры телеграм-бота начинается с настройки логирования и подключения библиотеки (рисунок 28).

```
1  import logging
2  from telegram.ext import Application
3
```

Рисунок 28 – Подключения библиотеки

Весь код строится на асинхронных функциях, что позволяет эффективно обрабатывать запросы пользователей и отвечать на них в режиме реального времени.

Основными элементами архитектуры телеграм-бота являются:

- командный обработчик (`CommandHandler`) - реагирует на команды, такие как `/start`, `/about_company` и другие;
- обработчик сообщений (`MessageHandler`) - отвечает на текстовые сообщения, выбранные пользователем из меню;

– ReplyKeyboardMarkup - создаёт интерфейс с кнопками, отображаемыми прямо в чате телеграм;

– Inline элементы - позволяют прикреплять интерактивные кнопки, документы, видео, фотографии.

Бот реализует информационные разделы компании АО «Прииск Соловьёвский», такие как: «Информация о компании», «Презентационное видео», «Документы», «Новости», «Контакты», «Вакансии».

Каждый из разделов открывает свое подменю с детализированным описанием и вложенными медиа-файлами: изображениями, PDF-документами и видео. Пример функции, выводящей краткую информацию о компании представлен на рисунке 29.

```
1 v async def company_info(update: Update, context: CallbackContext) -> None:
2     text = "👤 На сегодняшний день АО «Прииск Соловьёвский» имеет ..."
3     await update.message.reply_text(text, parse_mode='HTML')
```

Рисунок 29 – Пример функции

Все данные структурированы и обрабатываются через единый хендлер handle_message, который по тексту пользователя перенаправляет его в нужный раздел.

телеграм-бот также поддерживает отправку: изображений (update.message.reply_photo), видео (update.message.reply_video), документов (update.message.reply_document). Изображения, видео и документы которые необходимы для отправки хранятся в файловой системе телеграм-бота.

Пример кода для отправки ботом карты с месторождениями представлен на рисунке 30.

```
1 v with open('map.jpg', 'rb') as photo:
2     await update.message.reply_photo(photo=photo)
```

Рисунок 30 – Отправка изображения

Для запуска бота в PyCharm достаточно вызвать основную функцию main(), которая инициализирует приложение и запускает polling-режим, позволяющий боту получать обновления от телеграм:

Перед запуском необходимо убедиться, что токен корректен, а все функции бота асинхронные (`async def`), иначе бот не сможет обработать запросы. Итоговый код телеграм-бота представлен в приложении В.

Разработка телеграм-бота – это удобный и гибкий способ автоматизировать общение с пользователями. С помощью Python и библиотеки `python-telegram-bot` можно реализовать как простые команды, так и полноценные интерактивные меню. Интеграция мультимедийного контента делает бота не только функциональным, но и удобным для восприятия информации.

Процесс разработки в среде PyCharm делает управление проектом интуитивно понятным и удобным даже для начинающих разработчиков.

3.7 Реализация интерактивной анимации на веб-сайте

Современные технологии веб-разработки предполагают активное использование интерактивной анимации как средства улучшения пользовательского опыта, привлечения внимания и визуального обогащения интерфейса. Elementor предоставляет гибкие средства визуального редактирования, в том числе встроенные анимации, эффекты при наведении и возможность внедрения видеоконтента в качестве фонового элемента.

3.7.1 Реализация фонового видео на веб-сайте

Одним из центральных визуальных элементов сайта является видеофон, расположенный в верхней секции главной страницы. Использование видеоконтента позволяет создать динамичный, эмоционально насыщенный первый экран, формируя у пользователя первое впечатление о сайте.

Этапы добавления видеофона:

- добавление секции: в редакторе Elementor создается новая секция, растянутая на всю ширину экрана;
- настройка фонового видео: в панели редактирования выбирается вкладка «Фоновое видео», а затем вставляется ссылка на видео;
- оптимизация: устанавливаются параметры автозапуска и отключения звука.

Также прежде чем интегрировать видеоконтент в сайт с помощью Elementor, необходимо подготовить соответствующее видео. Фоновое видео должно соответствовать общей визуальной концепции проекта.

3.7.2 Анимация появления кнопок на веб-сайте

Анимация появления применяется к интерактивным кнопкам и элементам управления, с целью постепенного, визуально плавного вовлечения пользователя в интерфейс.

В рамках конструктора Elementor такие эффекты реализуются с помощью встроенного модуля Motion Effects, который обеспечивает гибкое управление анимацией появления, включая её тип, направление, скорость и поведение при прокрутке.

Пример настройки анимации появления кнопок на веб-сайте представлен в таблице 2.

Таблица 2 – Настройки анимации появления кнопок на веб-сайте

Параметр	Значение
Тип анимации	Zoom in
Направление	Центр
Длительность	700 мс
Задержка	250 мс
Повтор	Отключено (анимация запускается один раз при загрузке/появлении элемента в области видимости)
Область активации	При прокрутке до элемента

3.7.3 Изменение цвета при наведении на объекты

Анимации при наведении курсора мыши являются важным аспектом интерактивности веб-сайта. Они усиливают обратную связь и улучшают восприятие навигационных и функциональных элементов.

Для того чтобы создать задать объекту изменение цвета при наведении необходимо в редакторе Elementor выбрать нужный элемент (заголовок, кнопка, иконка), далее в разделе «Стиль» задать следующие параметры:

- цвет текста/фона по умолчанию;
- цвет текста/фона при наведении.

Пример конфигурации объектов на веб-сайте представлен в таблице 3.

Таблица 3 – Конфигурация объектов на веб-сайте

Элемент	Цвет (обычный)	Цвет (при наведении)
Кнопка	#FFD945	#B59B37
Заголовок	#FFD945	#B59B37
Подзаголовок	#1E293B	#E6BB12
Иконка соцсети	#54595F	#4B8DB2

3.8 Интеграция блока обратной связи

Для обеспечения быстрой и удобной связи пользователей с владельцем сайта был разработан блок обратной связи, с помощью плагина Forminator для CMS WordPress. Forminator – это мощный и удобный инструмент для создания форм различного типа на сайте без необходимости программирования.

Блок обратной связи должен включать:

- форму отправки сообщения с полями: имя, электронная почта, сообщение;
- возможность прикрепить файл;
- кнопку отправки с подтверждением отправки.

Техническая реализация блока с помощью плагина Forminator:

- установка плагина через панель WordPress;
- создание новой формы через меню Forminator;
- выбор типа формы;
- настройка полей формы через редактор;
- установка уведомлений;
- публикация формы на любой странице или записи через шорткод.

Разработанная форма обратной связи включает в себя несколько типов полей, обязательные к заполнению (поле для ввода имени, поле email) и необязательные (поле для ввода номера телефона и поле для ввода сообщения). Также в форме есть

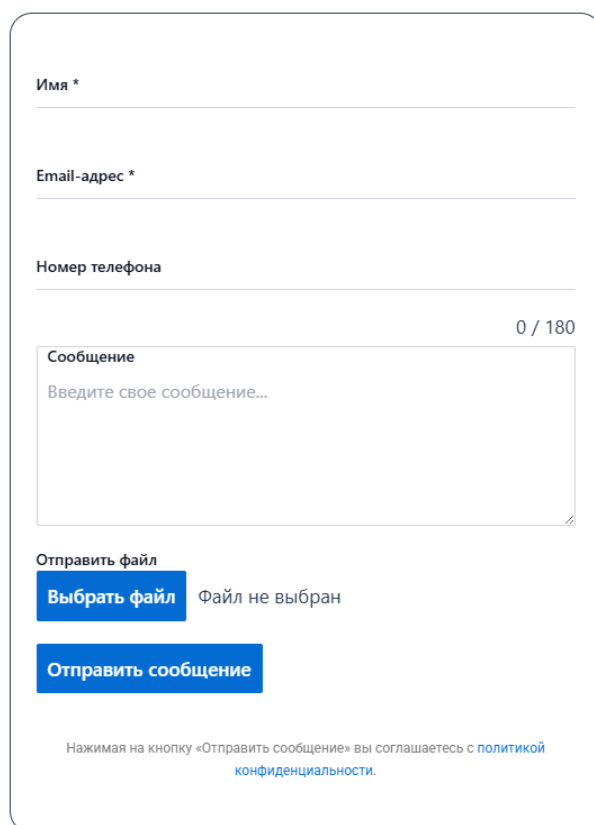
кнопка «Выбрать файл» при нажатии на которую открывается окно для загрузки файла.

Кнопка «Отправить сообщение» отвечает за проверку введенных данных и если они верны появляется текст который информирует об успешной отправки. Если введенные данные не соответствуют формату, то выводится сообщение с просьбой исправить введенные данные.

В нижней части формы расположена ссылка на страницу с политикой конфиденциальности.

Внешний вид формы обратной связи представлен на рисунке 31.

Для удобства поиска данной формы на каждой странице сайта в шапке расположена кнопка «Связаться с нами» нажатие на которую ведёт к форме.



The image shows a contact form with the following elements:

- Input field for "Имя *" (Name).
- Input field for "Email-адрес *" (Email address).
- Input field for "Номер телефона" (Phone number) with a character count "0 / 180".
- Text area for "Сообщение" (Message) with placeholder text "Введите свое сообщение..." (Enter your message...).
- Section "Отправить файл" (Send file) containing a blue button "Выбрать файл" (Choose file) and the text "Файл не выбран" (File not selected).
- A blue button "Отправить сообщение" (Send message).
- Footer text: "Нажимая на кнопку «Отправить сообщение» вы соглашаетесь с [политикой конфиденциальности](#)." (By clicking the "Send message" button, you agree with the [privacy policy](#)).

Рисунок 31 – Форма обратной связи

Когда пользователь заполняет и отправляет форму обратной связи на сайте, созданную при помощи плагина Forminator, начинается процесс обработки и сохранения введенной информации.

Сначала браузер пользователя отправляет данные формы на сервер сайта. Вместе с пользовательскими данными отправляются также служебные параметры, такие как скрытые идентификаторы формы. Эти дополнительные данные помогают серверу проверить, что отправка формы действительно была выполнена с нужной страницы сайта.

На стороне сервера WordPress данные сначала проходят этап проверки. Система проверяет, заполнены ли обязательные поля, правильный ли формат у введенных данных. Если всё в порядке, происходит переход к следующему шагу – сохранению данных.

Плагин Forminator сохраняет отправленную информацию в отдельные таблицы базы данных WordPress. Основная таблица хранит общую информацию об отправке формы: дату, IP-адрес пользователя и номер формы. Все отдельные поля формы, которые заполнил пользователь, записываются в другую таблицу – каждое поле отдельно с привязкой к основной записи отправки (рисунок 32).

☐	ID	Дата представления	Имя	Email-адрес	
☐	7	Май 5, 2025 @ 2:22 ДП	Иван	ivan@mail.ru	+ 3 другие поля ^
#7 Дата представления Май 5, 2025 @ 2:22 ДП Имя Иван Email-адрес ivan@mail.ru Номер телефона +7 1234567890 Сообщение Сообщение. Отправить файл Hn0f00Qp44ta-Файл.docx					
 УДАЛИТЬ  ПОВТОРНО ОТПРАВИТЬ УВЕДОМЛЕНИЕ ПО ЭЛЕКТРОННОЙ ПОЧТЕ					

Рисунок 32 – Отчет формы обратной связи

После сохранения данных плагин дополнительно отправляет уведомление администратору сайта на корпоративную почту.

Если на каком-то этапе возникает ошибка – например, данные не прошли проверку или токен формы оказался недействительным – процесс прерывается, и пользователю показывается сообщение об ошибке. В этом случае данные в базу не записываются.

Диаграмма последовательности процесса отправки формы представлена на рисунке 33.

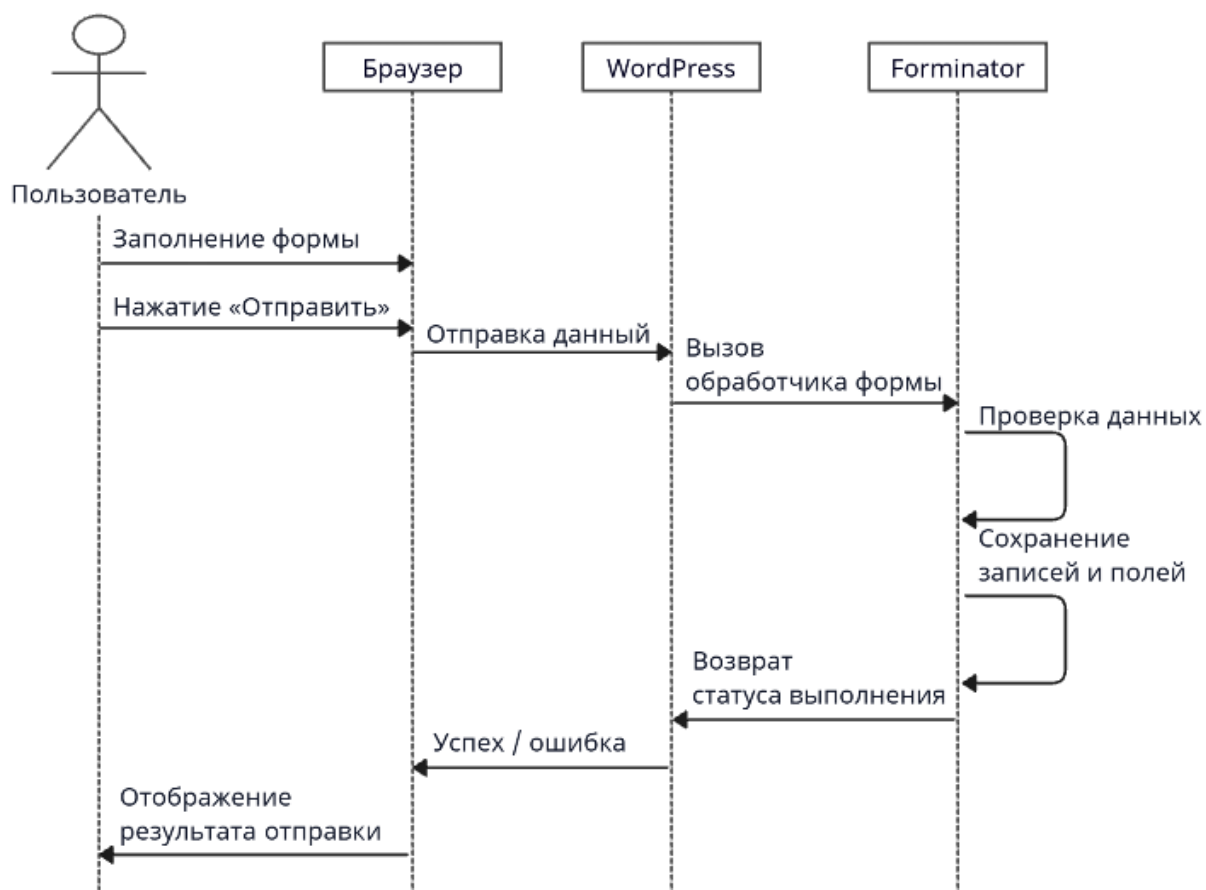


Рисунок 33 – Диаграмма последовательности процесса отправки формы

3.9 Внедрение дополнительных модулей

В рамках расширения функциональности веб-сайта был разработан и интегрирован модуль YaMap, основанный на сервисе Яндекс.Карты. Этот модуль позволяет пользователям быстро находить местоположение компании прямо на сайте и улучшая навигацию.

Основные характеристики модуля YaMap:

- отображение адреса компании с точной отметкой на карте;
- адаптивность (карта корректно отображается на любых устройствах);

– интерактивность (возможность увеличивать, уменьшать карту и просматривать ближайшие ориентиры).

Этапы создания и внедрения модуля YaMap:

– Использование сервиса «Конструктор карт Яндекс»

Для создания карты был использован официальный инструмент – Конструктор карт Яндекс, который позволяет без программирования сформировать карту с необходимыми настройками.

На этом этапе производится выбор местоположения компании, ввод адреса, установка и настройка маркера, настройка параметры отображения, выбор масштаба карты (рисунок 34). Далее для получения кода необходимо нажать на кнопку «Получить код», выбрать формат кода и скопировать полученный HTML-код, код карты представлен в приложении Г.

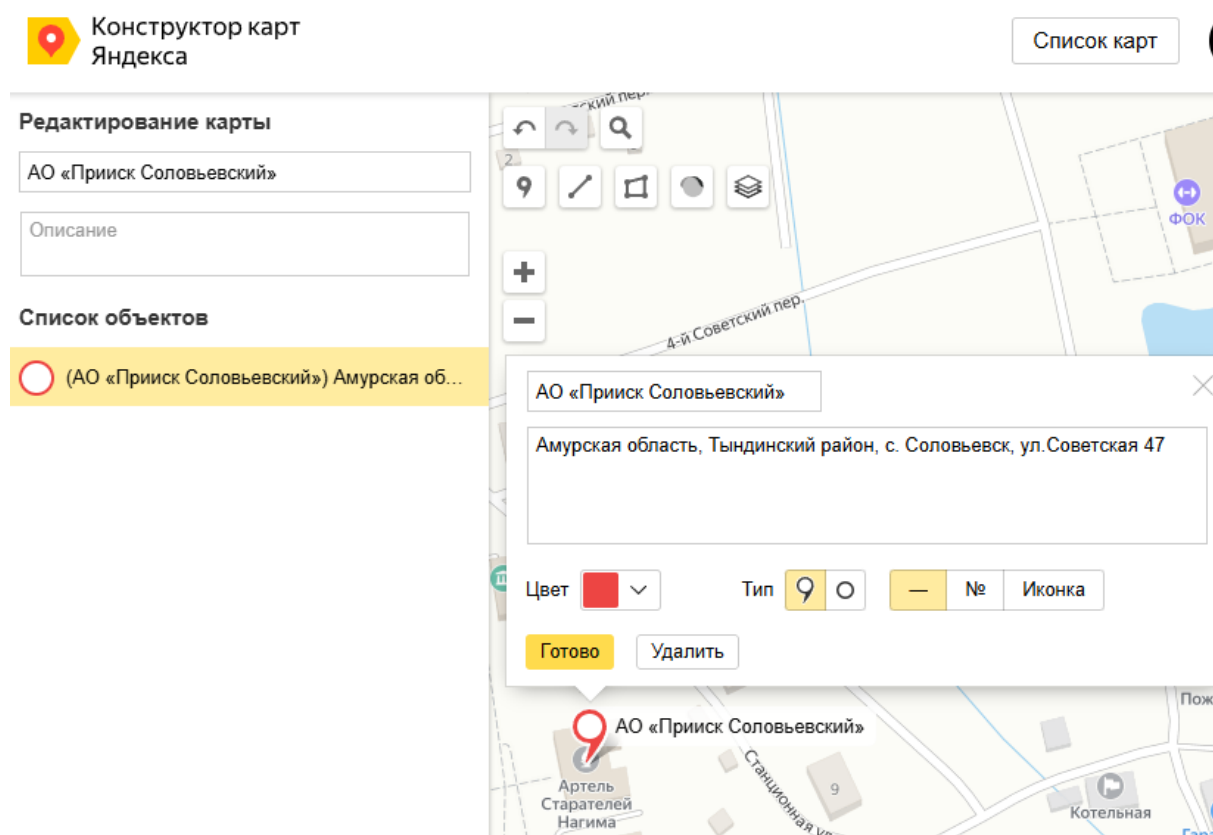


Рисунок 34 – Создание карты

- Вставка кода на сайт

После генерации и копирования кода следующими шагами была выполнена интеграция карты на сайт:

- открытие нужной страницы в редакторе WordPress;
- создание элемента «HTML-код»;
- вставка полученного кода.

Внешний вид карты представлен на рисунке 35.

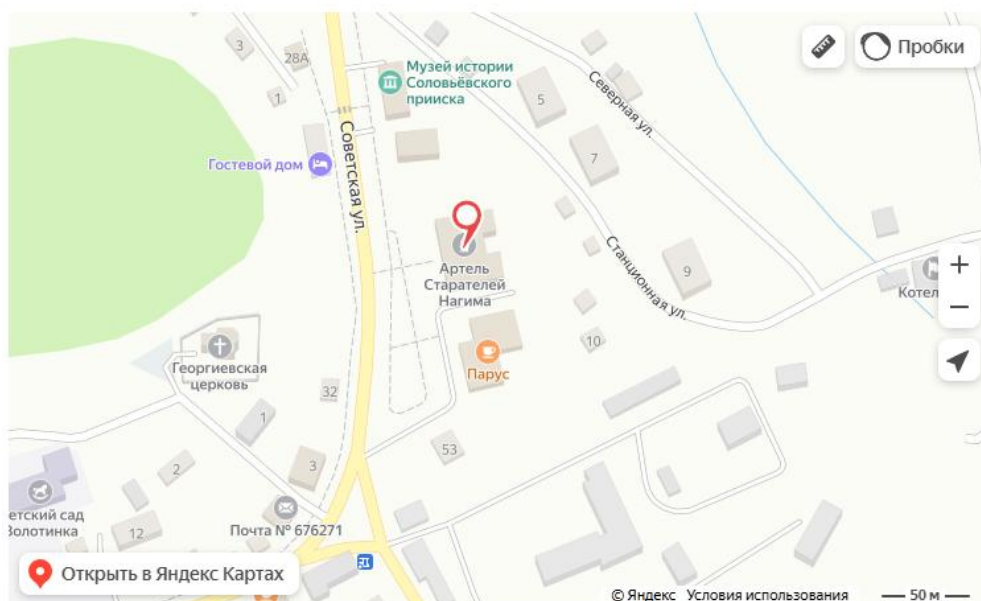


Рисунок 35 – Внешний вид блока

- Тестирование работы карты

После внедрения карта была проверена:

- на различных устройствах (смартфоны, планшеты, компьютеры);
- в разных браузерах (Chrome, Safari, Firefox, Edge);
- проверено правильное положение маркера и работоспособность элементов управления.

3.10 Адаптация сайта под мобильный формат

С переходом глобального трафика в сторону мобильных устройств, становится очевидной необходимость внедрения механизмов адаптации веб-контента к экранам с различным разрешением и ориентацией. Основной целью адаптации сайта под мобильные устройства является обеспечение интуитивного

взаимодействия, сохранения структуры контента и визуальной целостности при минимальной загрузке ресурсов.

В современной практике веб-разработки наиболее эффективным методом обеспечения кросс-платформенной совместимости является адаптивная верстка. Платформа WordPress предоставляет широкий инструментарий для реализации адаптивности. Одним из таких инструментов является визуальный конструктор Elementor, обладающий средствами для детальной настройки отображения элементов сайта под различные разрешения.

Elementor позволяет использовать принципы mobile-first разработки, предоставляя гибкие инструменты настройки блоков, шрифтов, отступов, выравнивания, видимости и поведения элементов интерфейса.

Основные функции Elementor для адаптации:

- режимы предпросмотра – позволяют переключаться между видами отображения: Компьютер, Планшет, Мобильный;
- индивидуальные настройки для каждого устройства – можно задать разный размер шрифта или отступ для мобильного экрана, не затрагивая версию для компьютера;
- скрывание и отображение элементов – возможность скрывать или показывать блоки на определённых устройствах.

Процесс адаптации сайта, разработанного в Elementor, под мобильные устройства можно условно разделить на следующие этапы:

- анализ существующего макета – выявление элементов, которые нарушают читаемость или структуру при уменьшении ширины экрана;
- активация адаптивного режима Elementor и настройка всех ключевых компонентов страницы в режиме Мобильный;
- проверка структуры контейнеров и обеспечение вертикального стека элементов, настройка порядка их отображения;
- оптимизация типографики – уменьшение размера заголовков, выбор читабельных шрифтов и интерлиньяжа;

- настройка отступов для устранения горизонтальной прокрутки;
- тестирование кроссбраузерности и функционала на реальных устройствах и в эмуляторах (Chrome DevTools, BrowserStack и др.).

Пример адаптированной главной страницы разработанного веб-сайта приведен на рисунке 36.

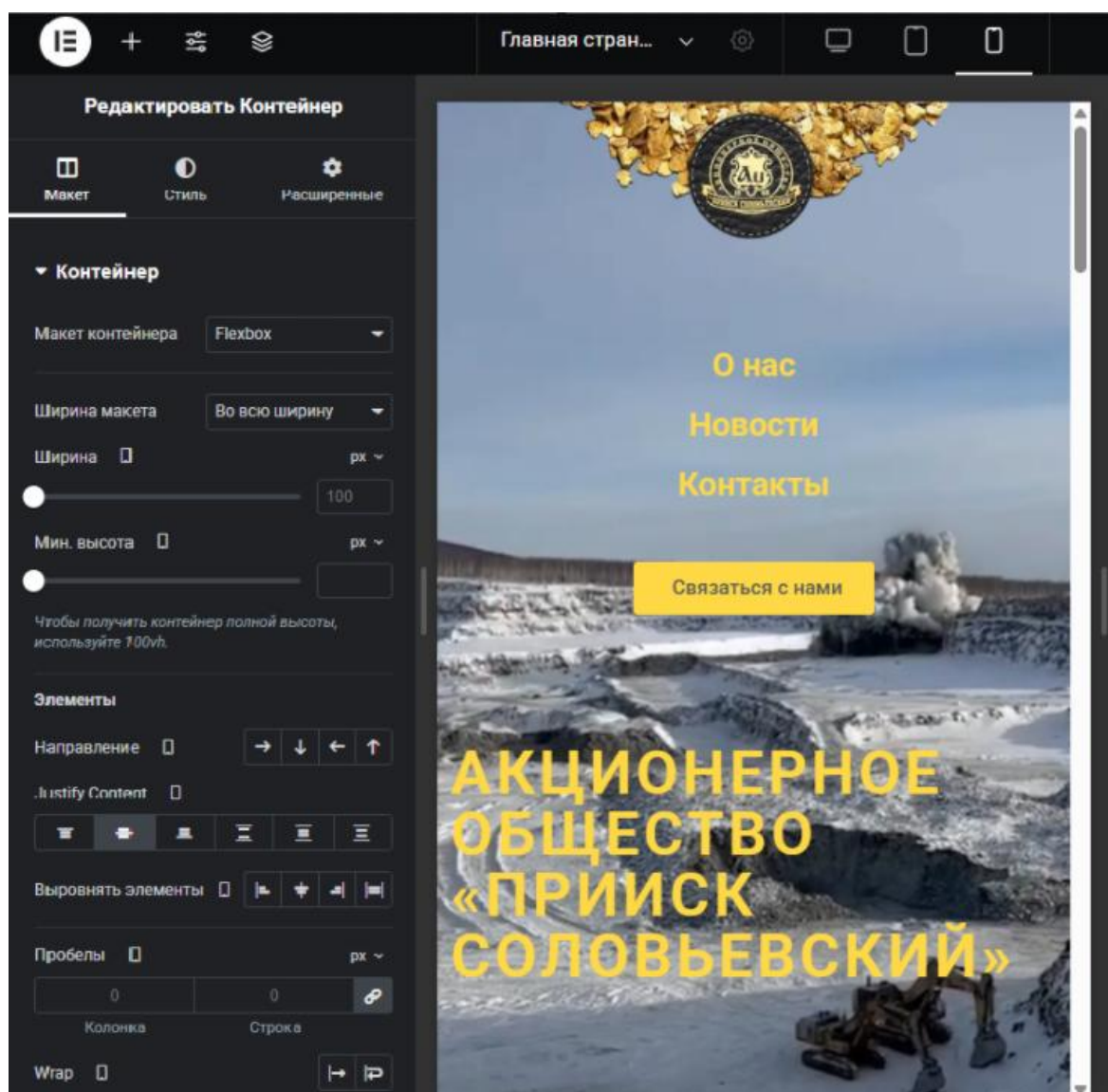


Рисунок 36 – Главная страница веб-сайта адаптированная под мобильный формат

Практическая реализация адаптации на основе главной страницы разработанного веб-сайта показала, что даже при наличии сложной структуры и насыщенного визуального наполнения возможно достижение высокой степени адаптивности без ущерба для функциональности. Таким образом Elementor

представляет собой не только мощный инструмент визуального проектирования, но и полноценную среду для создания профессионально адаптированных веб-интерфейсов.

3.11 Обзор веб-системы

3.11.1 Обзор веб-сайта priisk-solov.ru

Веб-сайт priisk-solov.ru представляет собой компактный, информативный интернет-ресурса, основной задачей которого является предоставление краткой информации о предприятии или частном лице. В данном случае речь идет о золотодобывающем предприятии АО «Прииск Соловьевский».

Цель веб-сайта – донести ключевую информацию до целевой аудитории: потенциальных партнеров, клиентов, соискателей, а также структур, заинтересованных в логистике, геологоразведке, инвестициях и пр.

Веб-сайт структурирован по стандартам UX-дизайна, имеет интуитивно понятную структуру и навигацию, а также минималистичное оформление, ориентированное на взрослую, преимущественно профессиональную аудиторию.

Веб-сайт содержит следующие ключевые разделы:

- главная страница – обзорная страница с кратким описанием деятельности;
- о нас – история, миссия, сведения о лицензиях и географии работ;
- новости – актуальные события, достижения, пресс-релизы;
- контакты – информация о местонахождении, телефонах, электронной почте.

Веб-сайт не перегружен визуальными эффектами и не содержит тяжелых скриптов, что обеспечивает высокую скорость загрузки даже при нестабильном соединении в удалённых регионах.

Веб-сайт реализован с использованием базовых технологий HTML, CSS, JavaScript и CMS WordPress. Применяется адаптивная вёрстка, корректно отображается на различных устройствах, включая смартфоны (рисунок 37).

Графическое оформление выдержано в сдержанной цветовой гамме. Шрифты читаемы, иерархия визуальных элементов соблюдена.

Сайт не содержит навязчивой рекламы и сторонних трекеров, что подчеркивает ориентацию на профессиональную этику взаимодействия с аудиторией.



Рисунок 37 – Главная страница веб-сайта в мобильном формате

3.11.2 Обзор телеграм-бота @priisksolov_bot

Телеграм-бот @priisksolov_bot – это вспомогательный инструмент цифрового взаимодействия, расширяющий каналы коммуникации предприятия с внешней средой. Он выполняет функцию цифровой визитки в мессенджере Телеграм – одном из самых популярных средств обмена информацией в России.

Основная задача бота — оперативное предоставление контактной информации и сведений о предприятии.

Телеграм-бот реализован на базе API Telegram Bot Platform и использованием языка программирования Python. Логика построена по принципу "кнопочного интерфейса". Реализованы следующие основные команды:

- «/start» – приветственное сообщение и вывод основного меню;
- «О компании» – история, миссия, сведения о лицензиях и географии работ;

- «Новости» – актуальные события, достижения, пресс-релизы;
- «Вакансии» – актуальные вакансии на предприятии;
- «Контакты» – информация о местонахождении, телефонах, электронной почте.

Стартовый экран телеграм-бота @priisksolov_bot представлен на рисунке 38.

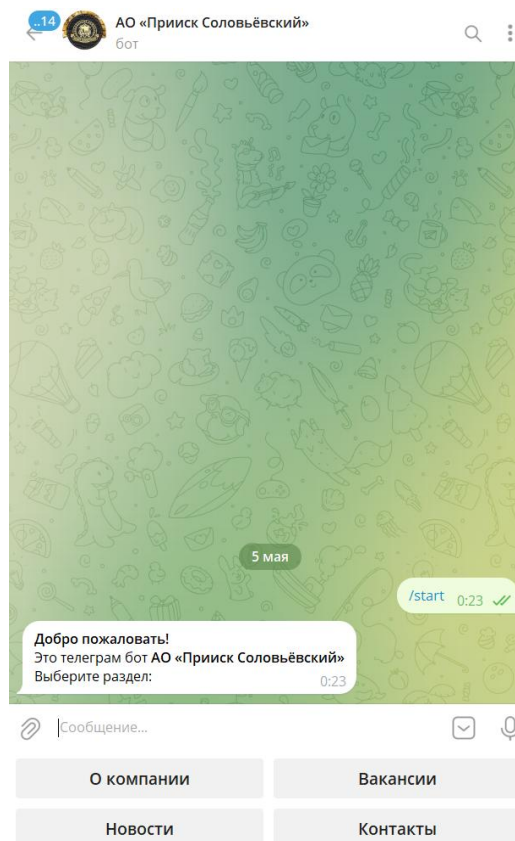


Рисунок 38 – Стартовый экран телеграм-бота @priisksolov_bot

Простой пример взаимодействия с телеграм-ботом:

- Пользователь нажимает кнопку "Контакты";
- Бот отправляет сообщение с актуальными данными (контактная информация предприятия, адрес и ссылки на веб-сайт и социальные сети).

Telegram-бот визитка является удачным дополнением к веб-ресурсу priisk-solov.ru, обеспечивая дополнительный мобильный канал связи и способ представления информации. Он демонстрирует простоту, надёжность и функциональность, что особенно важно в условиях добывающей отрасли, где необходимо быстро и безошибочно донести информацию до заинтересованных сторон.

4 ВОПРОСЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

4.1 Область защиты

В условиях стремительного развития информационных технологий и роста числа киберугроз обеспечение информационной безопасности становится одной из приоритетных задач любого предприятия. Особую актуальность эта задача приобретает в контексте цифровых продуктов, таких как веб-сайты и телекоммуникационные приложения, включая телеграм-ботов. Учитывая высокую степень вовлеченности пользователей и критичность обрабатываемых данных, необходимо тщательно определить и проанализировать области, подлежащие защите, в рамках функционирования указанных компонентов.

Данный этап работы охватывает основные направления, в которых необходимо реализовать меры информационной безопасности для поддержания целостности, конфиденциальности и доступности ресурсов предприятия. Рассматриваются как технические, так и организационные аспекты защиты веб-сайта и телеграм-бота предприятия, включая серверную инфраструктуру, каналы передачи данных, пользовательский контент и взаимодействие с внешними сервисами.

Для веб-сайта основное внимание уделяется защите хостинговой платформы и операционных систем, безопасности сетевого взаимодействия, а также мерам по защите персональных данных пользователей. телеграм-бот требует особого подхода к обеспечению безопасности в части взаимодействия с API Telegram и защите серверной инфраструктуры.

Таким образом, область защиты охватывает весь цикл функционирования цифровых решений предприятия, от серверного уровня до взаимодействия с конечным пользователем. Это позволяет создать всестороннюю и надежную систему защиты, минимизирующую возможные угрозы и обеспечивающую устойчивость информационных ресурсов к внешним и внутренним рискам.

4.2 Виды угроз и методы защиты от них

Современные информационные системы сталкиваются с множеством киберугроз, каждая из которых может нанести серьезный ущерб данным, имуществу и репутации компании. Ниже представлены основные виды угроз, а также проверенные методы защиты от них.

Основные угрозы информационной безопасности:

- Несанкционированный доступ. Попытки злоумышленников проникнуть в систему без наличия соответствующих прав. Это может быть результатом слабых паролей, отсутствия контроля доступа или утечек учетных данных. Например, хакер, использующий украденный логин и пароль сотрудника, получает доступ к корпоративной почте.

- Вредоносное программное обеспечение (вредоносное ПО). Программы, созданные для нанесения вреда – вирусы, черви, трояны, шпионские и рекламные программы. Они могут красть данные, уничтожать файлы, шпионить за пользователем. Троян, замаскированный под PDF-файл, который при открытии заражает компьютер.

- DDoS-атаки (Distributed Denial of Service). Массовая перегрузка сервера запросами с разных устройств, в результате чего ресурс становится недоступен для реальных пользователей. Например веб-сайт онлайн-магазина недоступен в период распродаж из-за DDoS-атаки конкурентов.

- Фишинг и социальная инженерия. Обман пользователей с целью получения конфиденциальной информации (паролей, номеров банковских карт и др.). Примером данного вида угрозы служит письмо якобы от банка с просьбой перейти по ссылке и ввести данные.

Методы защиты от угроз:

- Аутентификация и авторизация. Проверка личности пользователя и ограничение его прав. Надёжная защита начинается с настройки прав доступа. На практике применяется использование двухфакторной аутентификации (2FA) и назначение ролей с минимально необходимыми правами.

– Шифрование данных. Преобразование данных в форму, недоступную без ключа расшифровки. Методом защиты является использование протоколов SSL/TLS, которые защищают передаваемую по сети информацию.

– Регулярное обновление ПО. Производители программ часто выпускают обновления, устраняющие уязвимости. Несвоевременное обновление увеличивает риск взлома.

– Мониторинг и аудит. Отслеживание активности пользователей и системных событий позволяет выявлять подозрительное поведение.

Надёжная защита информации – это не одно решение, а комплексная стратегия. Комбинация технических мер, организационных процессов и повышения осведомленности сотрудников – ключ к эффективной кибербезопасности.

4.3 Мероприятия по защите веб-системы

С учётом многообразия современных угроз информационной безопасности, перед внедрением информационной системы были приняты комплексные организационные и технические меры, направленные на защиту как от внешних, так и от внутренних рисков.

Для предотвращения несанкционированного доступа к системе были внедрены механизмы аутентификации и авторизации. Настроены уровни доступа на основе принципа минимальных прав, что ограничивает возможности пользователей только теми функциями, которые им действительно необходимы. Дополнительно реализована двухфакторная аутентификация (2FA), что существенно повышает устойчивость к атакам, связанным с утечками паролей.

На всех компьютерах и серверах организации установлено сертифицированное антивирусное программное обеспечение с возможностью автоматического обновления баз сигнатур. Это позволяет оперативно реагировать на новые типы вирусов, троянов и шпионских программ. Дополнительно на сервере реализован запрет на хранение и запуск исполняемых файлов в общих папках, что минимизирует риск распространения вредоносного ПО внутри корпоративной сети.

Для защиты от DDoS-атак была приобретена услуга по фильтрации и анализу входящего трафика от специализированного поставщика. Эта система позволяет обнаруживать и блокировать подозрительную активность, направленную на перегрузку ресурсов компании, что гарантирует стабильную работу веб-приложений и сервисов, даже в условиях внешнего давления.

Сотрудники компании проходят регулярные инструктажи по информационной безопасности, в ходе которых рассматриваются типовые сценарии социальной инженерии и фишинга. Также были введены организационные меры по проверке подлинности входящих писем и ссылок, в том числе настройка фильтров на корпоративной почте и блокировка подозрительных доменов. Это позволяет эффективно выявлять мошеннические попытки сбора конфиденциальной информации.

Организована регулярная система резервного копирования данных с хранением копий в изолированном сегменте сети. Это позволяет быстро восстановить информацию в случае потери данных из-за сбоев оборудования, человеческого фактора или атак злоумышленников.

Комплекс реализованных мероприятий обеспечивает многослойную защиту информационной системы от наиболее актуальных угроз – от несанкционированного доступа и вирусов до атак на веб-приложения и фишинга. Эффективная кибербезопасность достигается не только с помощью технических средств, но и благодаря постоянному повышению квалификации персонала и соблюдению внутренней политики информационной безопасности.

5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

5.1 Безопасность

АО «Прииск Соловьевский» активно применяет цифровые инструменты в своей деятельности, включая эксплуатацию корпоративного веб-сайта и телеграм-бота. Работа с этими информационными системами требует постоянного взаимодействия сотрудников с персональными компьютерами, поэтому обеспечение безопасных и комфортных условий труда является приоритетной задачей. Контроль за соблюдением норм осуществляет специализированный отдел по технике безопасности предприятия.

5.1.1 Требования к персональным компьютерам

Рабочие места, на которых осуществляется эксплуатация цифровых систем, должны быть оснащены персональными компьютерами, соответствующими требованиям по уровню шума, вибрации и электромагнитного излучения. Совокупное воздействие всех устройств в зоне нахождения работника не должно превышать предельно допустимых значений, установленных СанПиН 2.1.3685-21.

Технические требования к ПК включают:

- допустимые уровни шума (не более 55 дБА);
- ограничения по электромагнитному полю (например, для диапазона 30–300 кГц — не более 25 В/м);
- регулировка яркости и контрастности изображения на экране;
- отсутствие бликов и отражающих поверхностей.

Мониторы должны соответствовать ГОСТ Р 50948–2001, в частности:

- яркость не менее 35 кд/м²;
- контрастность не ниже 3:1;
- равномерность подсветки;
- устойчивость изображения.

Оборудование, создающее повышенные шумы (серверы, источники бесперебойного питания), должно размещаться в отдельных помещениях или вне зоны постоянного пребывания сотрудников.

5.1.2 Помещения и организация рабочих мест

АО «Прииск Соловьевский» организует рабочие пространства с учётом норм по площади и освещенности. Для одного рабочего места с ЖК-монитором выделяется не менее 4,5 м², однако фактически предусмотрено около 10 м², что улучшает эргономику и микроклимат.

Все помещения, где размещены ПК, соответствуют требованиям по внутренней отделке: светлые потолки (коэффициент отражения 0,7–0,8), стены умеренного тона (0,5–0,6), полы с коэффициентом отражения 0,3–0,5. Электроснабжение соответствует ГОСТ Р 50571.5.54, с обязательным заземлением и отсутствием электромагнитных помех.

Отдел по технике безопасности регулярно проводит оценку соответствия рабочих зон требованиям охраны труда.

5.1.3 Освещение рабочих мест

В помещениях используются комбинированные системы освещения. Естественный свет регулируется с помощью жалюзи, а искусственное освещение обеспечивает уровень освещенности от 300 до 500 лк. Для исключения ослепления и бликов светильники расположены сбоку от рабочих мест.

Используемые источники света — преимущественно светодиодные и люминесцентные лампы с коэффициентом пульсации не более 5 %. Очистка светильников и замена ламп выполняется согласно графику технического обслуживания.[9]

5.1.4 Микроклимат и чистота воздуха

Параметры микроклимата в помещениях, где ведется работа с веб-сервисами и телеграм-ботом, поддерживаются на уровне, рекомендованном СанПиН 2.1.3685-21. Это включает:

- температуру воздуха: 22–24 °С зимой, 23–25 °С летом;

- влажность: 40–60 %;
- скорость движения воздуха: до 0,1 м/с.

Аэроионный состав и концентрация вредных веществ находятся под контролем, с учётом регулярной вентиляции и проверки воздуха на соответствие санитарным нормам. Администрация следит за соблюдением этих требований совместно с отделом охраны труда.

5.1.5 Эргономика и оснащение рабочих мест

Все рабочие места оборудованы эргономичной мебелью. Столы соответствуют требованиям по глубине (не менее 800 мм), высоте (725 мм для нерегулируемых, 680–800 мм – для регулируемых), а кресла – по всем параметрам регулируемости и комфорта.

Каждое место дополнительно оснащается подставками для ног и специальными поверхностями для размещения клавиатуры. Расстояние от глаз до экрана составляет не менее 600 мм.

Для работы, требующей концентрации, применяются мобильные перегородки, снижающие уровень отвлекающих факторов. Пример организации рабочего места с учетом норм эргономики представлен на рисунке 39.

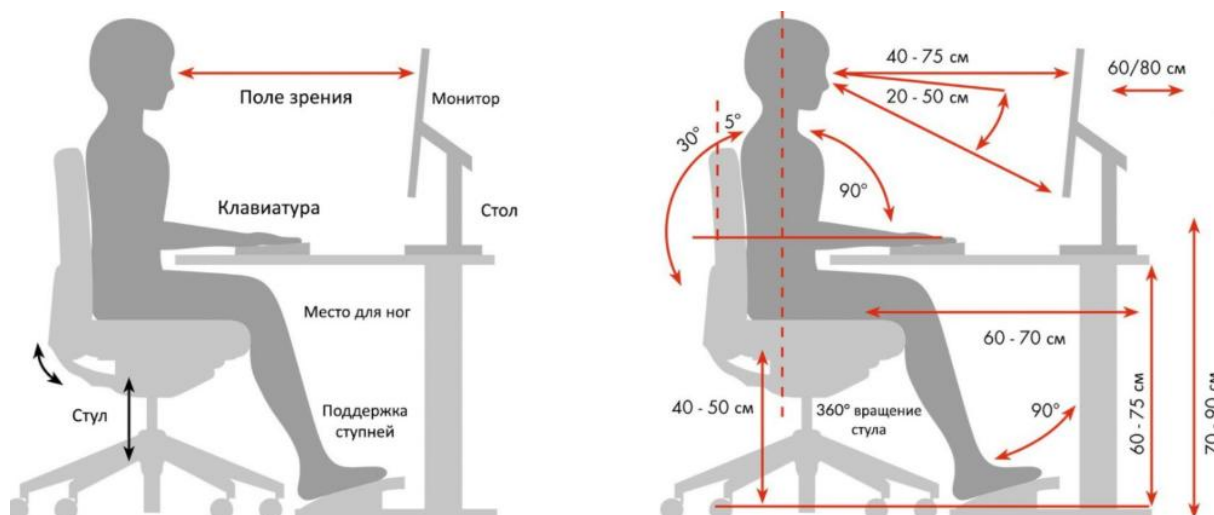


Рисунок 39 – Рабочее места с учетом норм эргономики

Пример стола отвечающего эргономическим требованиям представлен на рисунке 40.

отходов различного класса опасности. В соответствии со статьей 14 Федерального закона № 89-ФЗ «Об отходах производства и потребления», АО «Прииск Соловьевский» обязуется проводить идентификацию и классификацию образующихся отходов в установленных категориях:

- I класс – чрезвычайно опасные отходы (например, ртутьсодержащие лампы);
- II класс – высокоопасные отходы;
- III класс – умеренно опасные (электронные компоненты);
- IV класс – малоопасные (пластик, металл, оргтехника);
- V класс – практически неопасные отходы (макулатура, сломанная мебель и пр.).

Функции экологического контроля и обращения с отходами на предприятии возложены на отдел по технике и охране безопасности, который в тесном взаимодействии с IT-службой реализует меры по минимизации негативного воздействия на окружающую среду. Среди таких мер:

- вывод из эксплуатации устаревшей компьютерной и периферийной техники с последующей передачей в лицензированные организации, имеющие разрешение на утилизацию отходов I–IV классов опасности;
- постоянный мониторинг состояния оборудования, позволяющий своевременно выявлять устройства, подлежащие замене, и сокращать риск образования опасных отходов вследствие аварий;
- списание офисной мебели сопровождается оценкой возможности её ремонта или вторичного использования; в противном случае она направляется в специализированные пункты приёма как отходы V класса;
- в рамках повсеместного внедрения электронного документооборота значительно сокращены объемы бумажных документов, что снижает потребление бумаги, картриджей, тонеров и электроэнергии.

Несмотря на активное использование ЭДО, ряд административных и производственных процессов в АО «Прииск Соловьевский» по-прежнему требует обращения с бумажными документами (например, при отчётности, согласовании

технической документации). В таких случаях компания применяет принципы рационального использования:

- использование двусторонней печати по умолчанию;
- отказ от лишнего копирования;
- сортировка и централизованный сбор макулатуры для последующей передачи на переработку в лицензированные предприятия.

АО «Прииск Соловьевский» также предпринимает шаги к отказу от ртутьсодержащих источников света, заменяя их на энергоэффективные светодиодные светильники, которые имеют увеличенный срок службы и не представляют угрозы при утилизации. До полного отказа от люминесцентных ламп осуществляется их сбор, временное хранение в специально оборудованных помещениях и последующая утилизация в соответствии с нормами обращения с отходами I класса опасности.

С точки зрения цифровых решений, разработка веб-сайта и Telegram-бота велась с учётом энергосберегающих принципов программирования. Приоритет был отдан оптимизированным алгоритмам и эффективной архитектуре. Это способствует не только снижению эксплуатационных затрат, но и уменьшению непрямого углеродного следа, связанного с потреблением энергии на уровне дата-центров.

Таким образом, экологическая составляющая в рамках функционирования цифровой инфраструктуры АО «Прииск Соловьевский» не ограничивается техническими мерами – она интегрирована в корпоративную культуру устойчивого развития, где ответственность за окружающую среду рассматривается как важнейший аспект современного производственного подхода.

5.3 Чрезвычайные ситуации

В процессе эксплуатации информационных систем, таких как веб-сайт и телеграм-бот, функционирующих в рамках цифровой инфраструктуры АО «Прииск Соловьевский», крайне важно учитывать возможные риски, связанные с возникновением ЧС. Несмотря на то, что цифровые сервисы сами по себе не

являются источниками повышенной опасности, эксплуатация серверного оборудования, офисной техники и элементов ИТ-инфраструктуры сопряжена с определёнными техногенными и организационными угрозами.

Одним из наиболее вероятных сценариев ЧС в офисной среде по-прежнему остаётся возникновение пожара. Причинами могут выступать короткое замыкание, перегрузка электросети, неисправность или перегрев электрооборудования, а также человеческий фактор — нарушение инструкций эксплуатации или неосторожное обращение с огнем.

В соответствии с положениями Федерального закона № 123-ФЗ «Технический регламент о требованиях пожарной безопасности» и Постановлением Правительства РФ от 16.09.2020 № 1479 «Об утверждении Правил противопожарного режима в Российской Федерации», в АО «Прииск Соловьевский» действует регламентированная система обеспечения пожарной безопасности.

Организация располагает собственным отделом по технике безопасности, в компетенцию которого входит реализация комплекса профилактических и организационно-технических мероприятий, направленных на предупреждение пожаров и защиту жизни сотрудников. В частности, обеспечивается:

- разработка и ежегодное утверждение инструкций о мерах пожарной безопасности;
- постоянный контроль за состоянием автоматических систем пожарной сигнализации;
- наличие и исправность первичных средств пожаротушения (огнетушители, пожарные краны, огнезащитные покрывала);
- регулярное обновление схем эвакуации, размещаемых в офисных и серверных помещениях на видных местах;
- обеспечение свободного доступа к эвакуационным выходам и контроль их незагроможденности.

Кроме того, все помещения, где осуществляется эксплуатация ИТ-инфраструктуры, отделаны трудногорючими материалами, исключающими быстрое

распространение огня, а также оснащены системами аварийного освещения и сигнализацией с голосовым оповещением.[10]

Руководство АО "Прииск Соловьевский", согласно требованиям законодательства, обеспечивает:

- проведение обязательного противопожарного инструктажа для всех сотрудников, включая специалистов, работающих с веб-приложениями и ботами;
- организацию практических тренировок по эвакуации персонала не реже двух раз в год;
- ежегодную оценку состояния пожарной безопасности зданий, используемых под офисные и серверные нужды;
- эксплуатационные испытания эвакуационных конструкций (лестничных клеток, аварийных выходов и т.д.) – раз в 5 лет;
- своевременную маркировку всех эвакуационных путей и установку стандартных знаков пожарной безопасности.

Так как инфраструктура цифровых сервисов компании, в том числе веб-сайт и телеграм-бот, развернута в защищенной среде, включая локальные серверы и облачные хостинги, также предусмотрены меры информационной устойчивости на случай пожара или отключения электроэнергии. Используются системы резервного копирования данных, дублирующие критически важную информацию на удаленные носители и облачные хранилища. Это позволяет минимизировать риски утраты данных и быстро восстановить работу сервисов после устранения последствий ЧС.

Также в компании утвержден локальный план реагирования при ЧС, в который включены не только пожарные угрозы, но и сценарии отключения электроэнергии, обрушения конструкций, утечки воды и другие возможные инциденты, способные затронуть функционирование ИТ-систем.

ЗАКЛЮЧЕНИЕ

Результатом проведённой выпускной квалификационной работы стало создание функционального веб-сайта и телеграм-бота для золотодобывающего предприятия АО «Прииск Соловьевский». Выполненная разработка позволила не только сформировать цифровое присутствие предприятия в информационном пространстве, но и обеспечить оперативное и взаимодействие с пользователями.

В ходе работы была достигнута поставленная цель – разработка программного обеспечения для веб-системы АО «Прииск Соловьевский», а именно разработка веб-сайта и телеграм-бота, соответствующих требованиям и особенностям золотодобывающего предприятия, с учетом современных подходов к пользовательскому интерфейсу, безопасности, доступности и масштабируемости решений. Созданные программные решения обладают высокой степенью адаптивности, простотой в эксплуатации и возможностью масштабирования.

В рамках работы были последовательно решены следующие задачи:

- проведен анализ предметной области и выявлена специфика информационных потребностей золотодобывающего предприятия;
- сформированы функциональные и технические требования к создаваемым цифровым продуктам;
- спроектирована архитектура цифровых решений, включая интерфейсные и серверные компоненты;
- реализован веб-сайт с использованием современных веб-технологий;
- разработан телеграм-бот с необходимым функционалом на основе API Telegram на языке программирования Python;
- выполнено тестирование компонентов на соответствие требованиям и стабильность работы.

Таким образом, проделанная работа демонстрирует практическую реализацию методов и средств разработки программного обеспечения в условиях отраслевой специфики. Полученные результаты могут быть масштабированы и адаптированы

для других предприятий горнодобывающей и смежных отраслей. Кроме того, проект может стать основой для дальнейших научных и инженерных разработок в направлении цифровизации традиционных производств.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Литвиненко А. А. Информационные системы на предприятии. Москва: КноРус, 2020. 368 с.
- 2 Чекмарев Д. Ю. Разработка Telegram-ботов на Python. Москва: Эксмо, 2022. 192 с.
- 3 Telegram Bot API [Электронный ресурс]. URL:<https://core.telegram.org/bots/api> (дата обращения: 08.03.2025).
- 4 Python Telegram Bot [Электронный ресурс]. URL:<https://github.com/python-telegram-bot/python-telegram-bot> (дата обращения: 30.03.2025).
- 5 JetBrains [Электронный ресурс]. URL:<https://www.jetbrains.com/pycharm> (дата обращения: 11.04.2025).
- 6 Бережной В. П. Современные технологии веб-разработки: HTML, CSS, JavaScript. Москва: Диалектика, 2022. 278 с.
- 7 Python [Электронный ресурс]. URL:<https://www.python.org> (дата обращения: 12.04.2025).
- 8 Кормушин И. В. Web-дизайн: теория и практика. Москва: ДМК Пресс, 2022. 304 с.
- 9 Резчиков Е. А., Рязанцева А. В. Безопасность жизнедеятельности: учебник для вузов. Москва: Юрайт, 2025. 638 с.
- 10 Беляков Г. И. Пожарная безопасность, безопасность в чрезвычайных ситуациях и оказание первой помощи: учебник для вузов. Москва: Юрайт, 2025. 529 с.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Беляков, Г. И. Пожарная безопасность, безопасность в чрезвычайных ситуациях и оказание первой помощи: учебник для вузов./ Г. И. Беляков. – 5-е изд., перераб. и доп. – Москва: Изд-во Юрайт, 2025. – 529 с.
- 2 Бережной, В. П. Современные технологии веб-разработки: HTML, CSS, JavaScript./ В. П. Бережной. – Москва: Диалектика, 2022. – 278 с.
- 3 Блинов И. Ю. Основы работы с API Telegram./ И. Ю. Блинов. – Москва: Инфра-М, 2020. – 192 с.
- 4 Бочаров, В. В. Основы информационных технологий./ В. В. Бочаров. – Москва: Инфра-М, 2020. – 384 с.
- 5 Бушманов, А. В. Методические указания к выполнению и защите выпускной квалификационной работы бакалавра/ А. В. Бушманов, С. Г. Самохвалова, Т. А. Галаган. – Б.: ФГБОУ ВО «АмГУ», 2025. – 57 с.
- 6 Григорьев, Д. В. Современные веб-технологии: HTML5, CSS3, JavaScript, React./ Д. В. Григорьев. – СПб.: Питер, 2023. – 400 с.
- 7 Кормушин, И. В. Web-дизайн: теория и практика./ И. В. Кормушин. – Москва: ДМК Пресс, 2022. – 304 с.
- 8 Литвиненко, А. А. Информационные системы на предприятии./ А. А. Литвиненко. – Москва: КноРус, 2020. – 368 с.
- 9 Лобанов А. В. Основы веб-программирования: учебное пособие./ А. В. Лобанов. – Москва: КНОРУС, 2020. – 256 с.
- 10 Макаров, С. Н. Проектирование информационных систем./ С. Н. Макаров. – СПб.: Питер, 2021. – 352 с.
- 11 Николаев, А. П. Создание сайтов с использованием CMS: WordPress, Joomla, Drupal./ А. П. Николаев. – Москва: КноРус, 2022. – 296 с.
- 12 Резчиков, Е. А. Безопасность жизнедеятельности: учебник для вузов./ Е. А. Резчиков, А. В. Рязанцева. – 4-е изд., перераб. и доп. – Москва: Изд-во Юрайт, 2025. – 638 с.

- 13 Самойленко В. А. HTML, CSS и JavaScript для начинающих./ В. А. Самойленко. – Санкт-Петербург: БХВ-Петербург, 2020. – 320 с.
- 14 Смирнов, С. Н. Основы охраны труда: учебное пособие./ С. Н. Смирнов. – СПб.: Питер, 2019. – 288 с.
- 15 Сэммерс, Д. Python. К вершинам мастерства./ Д. Сэммерс. – Москва: ДМК Пресс, 2021. – 432 с.
- 16 Чекмарев, Д. Ю. Разработка Telegram-ботов на Python./ Д. Ю. Чекмарев. – Москва: Эксмо, 2022. – 192 с.
- 17 Drupal [Электронный ресурс]. – Режим доступа: <https://www.drupal.org> – 07.04.2025.
- 18 JetBrains [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/pycharm> – 11.04.2025.
- 19 Joomla [Электронный ресурс]. – Режим доступа: <https://www.joomla.org> – 07.04.2025.
- 20 Python [Электронный ресурс]. – Режим доступа: <https://www.python.org> – 12.04.2025.
- 21 Python Telegram Bot [Электронный ресурс]. – Режим доступа: <https://github.com/python-telegram-bot/python-telegram-bot> – 09.04.2025.
- 22 Telegram Bot API [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api> – 08.04.2025.
- 23 WordPress [Электронный ресурс]. – Режим доступа: <https://wordpress.org> – 07.04.2025.

ПРИЛОЖЕНИЕ А

Техническое задание

1 Введение

Работа выполняется в рамках дипломной работы на тему «Разработка программного обеспечения для веб-системы АО «Прииск Соловьевский»»

2 Основания для разработки

– Основания для разработки является задание на производственную практическую подготовку. Наименование темы разработки – «Проектирование web-сайта и телеграм бота АО «Прииск Соловьевский»».

– Наименование работы:

«Проектирование web-сайта и телеграм бота АО «Прииск Соловьевский»».

– Исполнители: студент группы 1105-об, института компьютерных и инженерных наук, кафедры информационных и управляющих систем, Амурского государственного университета Дудник Евгений Сергеевич.

– Соисполнители: нет.

3 Назначение разработки

Предоставление информации о предприятии, процессах добычи, новостях компании, взаимодействие с сотрудниками и партнерами через сайт и телеграм-бот.

4 Технические требования

4.1. Требования к функциональным характеристикам.

4.1.1 Состав выполняемых функций. На веб-сайте необходимо создать следующие страницы:

– главная страница сайта – основная информационная страница с основной информацией и ссылками на другие страницы, также данная страница содержит анимационный фон;

– страница «Контакты» – расположена вся контактная информация

Продолжение ПРИЛОЖЕНИЯ А

предприятия и модуль с картой;

– страница «Политика конфиденциальности» – политика конфиденциальности при отправки личных данных в форме обратной связи;

– страница «Новости» – страница где расположены новостные блоки;

– страница «О нас» – данная страница включает в себя ссылки на подстраницы;

– подстраница «Главные специалисты» – содержит блоки с информацией о главных специалистах предприятия;

– подстраница «Руководство» – содержит блоки с информацией о руководстве предприятия;

– подстраница «Состав совета директоров» – содержит блоки с информацией о составе совета директоров предприятия;

– подстраница «История компании» – на данной странице представлена история предприятия;

– подстраница «Отчеты и результаты» – страница на которой опубликованы годовые отчеты предприятия;

– подстраница «Информация, подлежащая опубликованию» – информационная страница, на которой публикуются файлы необходимые акционерам предприятия.

Меню должно содержать следующие элементы:

– «Контакты»;

– «Новости»;

– «О нас».

Все разделы веб-сайта должны быть доступны всем пользователям.

В телеграм-боте необходимо создать следующие страницы:

– главная страница сайта;

Продолжение ПРИЛОЖЕНИЯ А

- страница «О компании» – ссылки на подстраницы;
- страница «Новости» – информационное сообщение с пятью последними новостями предприятия;
- страница «Вакансии» – ссылки на подстраницы;
- страница «Контакты» – контактная информация;
- подстраница «Краткая информация о компании» – сообщение с краткой информацией о предприятии;
- подстраница «География добычи» – информация о местах добычи и иллюстрация;
- подстраница «Презентационное видео» – презентационное видео предприятия;
- подстраница «История и достижения» – информационное сообщение с историей предприятия;
- подстраница «Технологии и оборудование» – описание используемых технологий и оборудования с иллюстрацией;
- подстраница «Документы» – сообщение с годовыми отчетами за последние 5 лет;
- подстраница «Актуальные вакансии» – информационное сообщение с актуальными вакансиями предприятия;
- подстраница «Часто задаваемые вопросы» – информационная подстраница с часто задаваемыми вопросами и ответами на них;
- подстраница «Связаться с HR-отделом» – сообщение с контактной информацией HR-отдела;
- подстраница «Отправить резюме» – сообщение с информацией по отправке резюме.

Главное меню должно содержать следующие элементы:

Продолжение ПРИЛОЖЕНИЯ А

- страница «О компании»;
- страница «Новости»;
- страница «Вакансии»;
- страница «Контакты».

Все разделы должны быть доступны всем пользователям.

На каждом разделе, кроме главного меню, должна присутствовать кнопка «Назад» для возвращения к предыдущему разделу.

4.1.2 Организация входных и выходных данных.

Данные которые пользователь отправляет через модуль обратной связи хранятся в почтовом ящике.

Данные необходимые для скачивания хранятся в файловой системе.

4.2. Требования к надежности.

SSL-сертификат. Надежность и безопасность информационной системы являются ключевыми аспектами при её проектировании, разработке и эксплуатации. Система должна обеспечивать бесперебойную работу, защиту данных от потерь и устойчивость к внешним и внутренним угрозам. Система должна быть защищена от: SQL-инъекций и DDoS атак.

4.3. Условия эксплуатации и требования к составу и параметрам технических средств.

Веб-сайт должен эксплуатироваться в браузерах Google Chrome, Яндекс Браузер, Microsoft Edge, Safari, Opera, Mozilla Firefox. Также веб-сайт должен корректно отображаться в мобильных версиях браузера.

4.4. Требования к информационной и программной совместимости.

Кросс-браузерность: веб-сайт должен эксплуатироваться в браузерах Google Chrome, Яндекс Браузер, Microsoft Edge, Safari, Opera, Mozilla Firefox. Также веб-сайт должен корректно отображаться в мобильных версиях браузера.

Продолжение ПРИЛОЖЕНИЯ А

Скорость загрузки: Сайт должен иметь оптимизированные изображения, CSS и JavaScript, чтобы обеспечить быструю загрузку страниц.

Система должна обеспечивать бесперебойную работу, защиту данных от потерь и устойчивость к внешним и внутренним угрозам. Система должна быть защищена от: SQL-инъекций и DDoS атак.

4.5. Требования к маркировке и упаковке.

Программное изделие передается по сети Internet. Специальных требований к маркировке не предъявляется.

4.6. Требования к транспортированию и хранению.

Специальных требований не предъявляется/

4.7. Специальные требования.

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы Windows 10/11.

5 Требования к программной документации

Предварительный состав программной документации:

- техническое задание;
- программа и методика испытаний;
- руководство системного программиста;
- руководство программиста.

6 Техничко-экономические показатели

Программное обеспечение для веб-системы АО «Прииск Соловьевский» разрабатывается для конкретного предприятия - АО «Прииск Соловьевский». Экономический эффект обеспечивается за счет реализации.

7 Требования к программной документации

Срок начала работ: 07.04.2025

Срок окончания работ: 19.04.2025

ПРИЛОЖЕНИЕ Б

Макеты основных страниц веб-сайта



Рисунок Б.1 – Макет хедера и футера



Рисунок Б.2 – Страница «О нас»

Продолжение ПРИЛОЖЕНИЯ Б



Рисунок Б.3 – Страница «Новости»

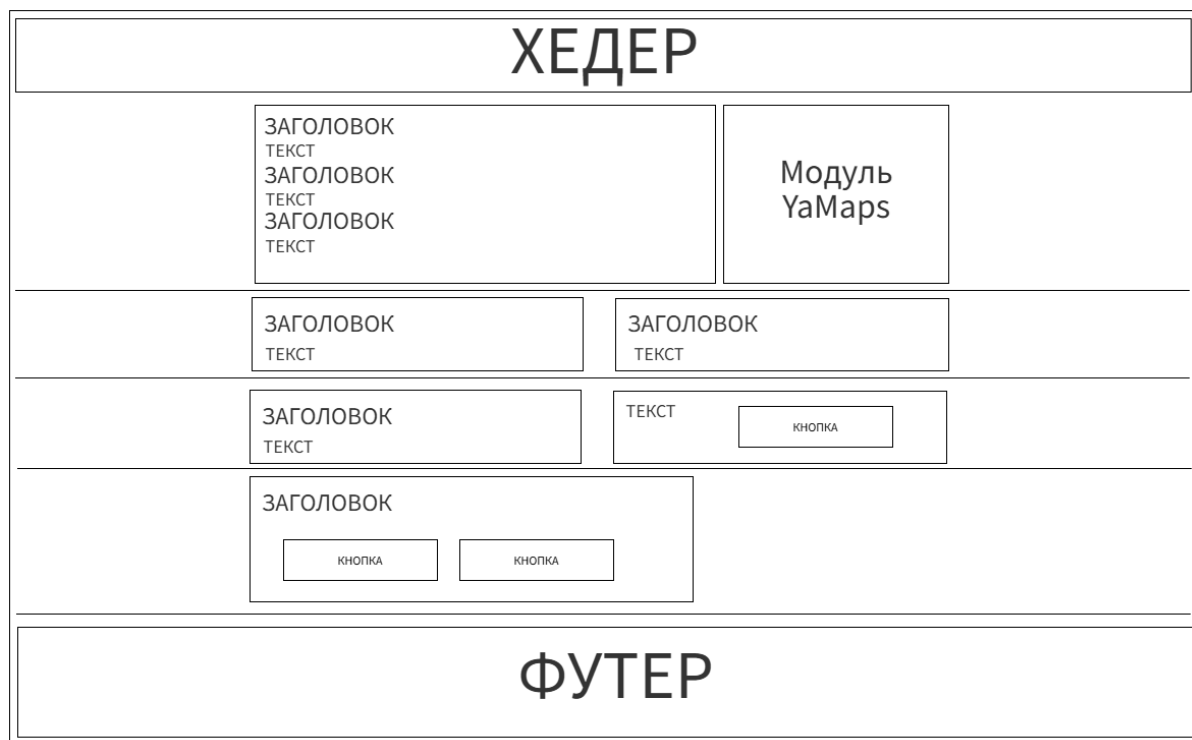


Рисунок Б.4 – Страница «Контакты»

Продолжение ПРИЛОЖЕНИЯ Б

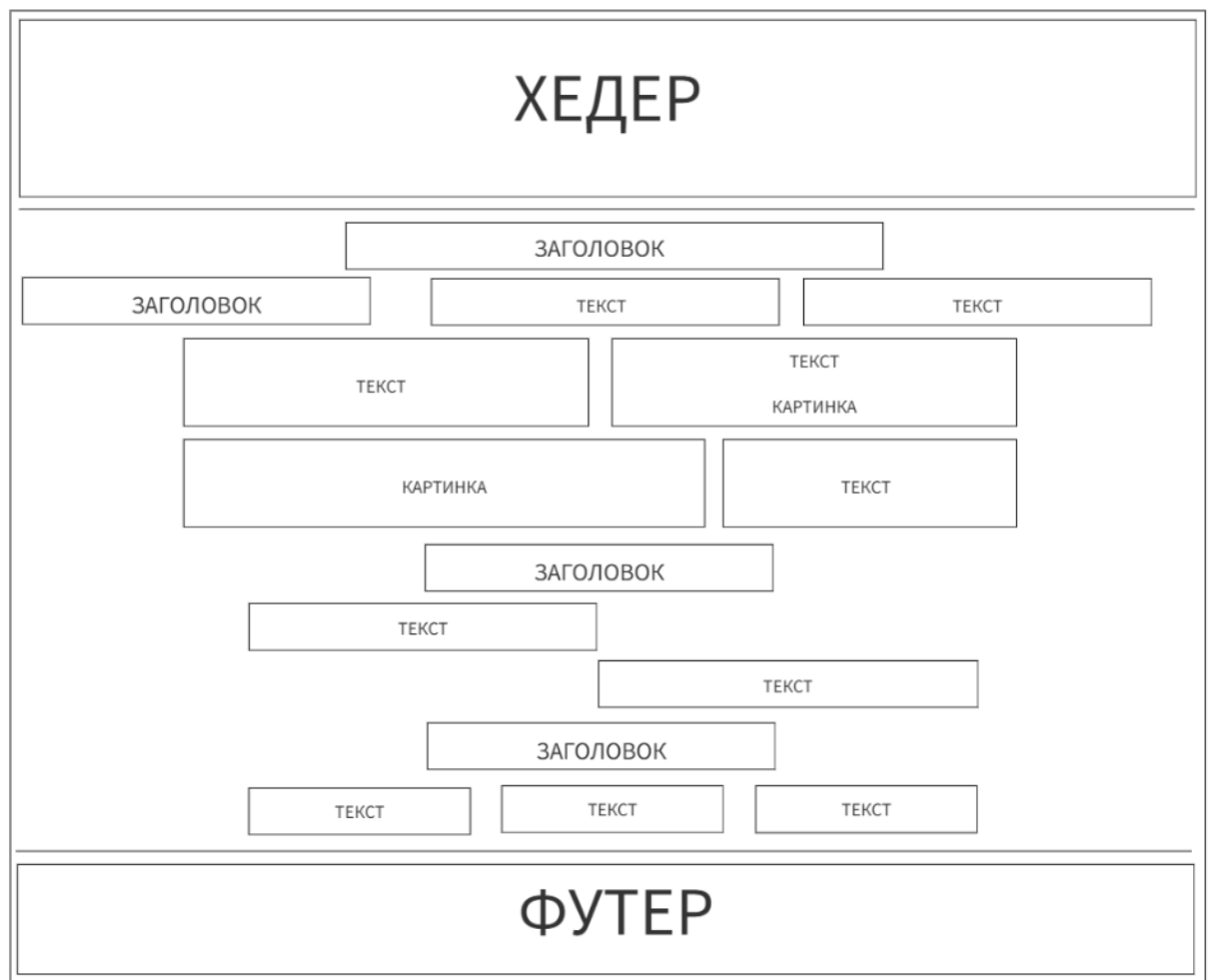


Рисунок Б.5 – Страница «Главная страница»

ПРИЛОЖЕНИЕ В

Листинг телеграм-бота

```
from telegram import Update, ReplyKeyboardMarkup, InlineKeyboardButton,
InlineKeyboardMarkup
from telegram.ext import Application, CommandHandler, MessageHandler, filters,
CallbackContext
import logging

# Включаем логирование
logging.basicConfig(format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
level=logging.INFO)
logger = logging.getLogger(__name__)

# Токен бота (получен от BotFather)
TOKEN = '7282085766:AAHAnDMOLY0qm01JUFI1QAYB-pjiVf4rLAs'

# Главное меню
async def start(update: Update, context: CallbackContext) -> None:
    keyboard = [
        ['О компании', 'Вакансии'],
        ['Новости', 'Контакты']
    ]
    reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)
    await update.message.reply_text('<b>Добро пожаловать!</b>\nЭто телеграм бот <b>АО  
«Прииск Соловьёвский»</b>\nВыберите раздел:', reply_markup=reply_markup,
parse_mode='HTML')

# Функция для возвращения в главное меню
async def back_to_main(update: Update, context: CallbackContext) -> None:
    keyboard = [
        ['О компании', 'Вакансии'],
        ['Новости', 'Контакты']
    ]
    reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)
    await update.message.reply_text('Вы вернулись в главное меню. Выберите раздел:',
reply_markup=reply_markup)

# О компании
async def about_company(update: Update, context: CallbackContext) -> None:
    keyboard = [
        ['📄 Краткая информация о компании', '🏆 История и достижения'],
        ['🌐 География добычи', '🔧 Технологии и оборудование'],
        ['🎥 Презентационное видео', '📁 Документы'],
        ['⏪ Назад']
    ]
    reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)
    await update.message.reply_text('Выберите, что вы хотите узнать об компании:',
```

Продолжение ПРИЛОЖЕНИЯ В

```
reply_markup=reply_markup)

# Краткая информация о компании
async def company_info(update: Update, context: CallbackContext) -> None:
    text = "🏢 На сегодняшний день <b>АО «Прииск Соловьёвский»...</b>"
    await update.message.reply_text(text, parse_mode='HTML')

# История и достижения
async def history(update: Update, context: CallbackContext) -> None:
    text = "📖 <b>История одного из старейших приисков современной России...</b>"
    await update.message.reply_text(text, parse_mode='HTML')

# География добычи
async def geography(update: Update, context: CallbackContext) -> None:
    regions = "🌐 На сегодняшний добыча полезных ископаемых ведется в двух регионах  
России:\n1. 📍 <b>Амурская область</b>\n2. 📍 <b>Забайкальский край</b>"
    await update.message.reply_text(regions, parse_mode='HTML')

# Отправка изображения с картой
with open('05_and3195.jpg', 'rb') as photo:
    await update.message.reply_photo(photo=photo)

# Технологии и оборудование
async def technologies(update: Update, context: CallbackContext) -> None:
    text = "🔧 📄 <b>Разработка рудного карьера ведется при помощи...</b>"
    await update.message.reply_text(text, parse_mode='HTML')
    with open('10_and3618.jpg', 'rb') as photo:
        await update.message.reply_photo(photo=photo)

# Презентационное видео
async def presentation_video(update: Update, context: CallbackContext) -> None:
    with open('vid.mp4', 'rb') as video_file:
        await update.message.reply_video(video=video_file)

# Документы (отчёты)
async def documents(update: Update, context: CallbackContext) -> None:
    with open('res.pdf', 'rb') as pdf_file:
        await update.message.reply_document(document=pdf_file)

# Вакансии
async def vacancies(update: Update, context: CallbackContext) -> None:
    keyboard = [
        ['🔥 Актуальные вакансии', '📧 Отправить резюме'],
        ['🏠 Условия проживания, график, питание', '❓ Часто задаваемые вопросы'],
        ['➡️ 📞 Связаться с HR-отделом'],
        ['🏠 Назад']
    ]
```

Продолжение ПРИЛОЖЕНИЯ В

```
reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)
await update.message.reply_text('Выберите интересующий вас раздел :',
reply_markup=reply_markup)

# 🚩 Актуальные вакансии
async def current_vacancies(update: Update, context: CallbackContext) -> None:
    vacancies_text = "📄Вакансия номер 1\n📄Вакансия номер 2"
    await update.message.reply_text(vacancies_text, parse_mode='HTML')

# ❏ Часто задаваемые вопросы
async def faq(update: Update, context: CallbackContext) -> None:
    faq_text = "❏<b>Как трудоустроиться в АО «Прииск Соловьёвский»?..."
    await update.message.reply_text(faq_text, parse_mode='HTML')

# ➡📄 Связаться с HR-отделом
async def contact_hr(update: Update, context: CallbackContext) -> None:
    contact_text = "☎❏<b> Телефон:</b>+7 (41656) 3-44-04\n✉<b>Email:</b> ok@solov.ru"
    await update.message.reply_text(contact_text, parse_mode='HTML')

# Отправить резюме
async def send_resume(update: Update, context: CallbackContext) -> None:
    text = "✉️Отправьте свое резюме в на email: ok@solov.ru"
    await update.message.reply_text(text, parse_mode='HTML')

# Новости
async def news(update: Update, context: CallbackContext) -> None:
    text = "📰<b>Последние новости компании:</b>"
    await update.message.reply_text(text, parse_mode='HTML')

    with open('new.jpg', 'rb') as photo:
        await update.message.reply_photo(photo=photo)

# Контакты
async def contacts(update: Update, context: CallbackContext) -> None:
    text = "☎<b> Телефон:</b>+7 (41656) 3-44-04\n<b>Email:</b> kanc@solov.ru"
    await update.message.reply_text(text, parse_mode='HTML')
    text = "📍<b>Адрес:</b> 676271, Амурская область, Тындинский район, с. Соловьёвск,
ул.Советская 47"
    await update.message.reply_text(text, parse_mode='HTML')
    text = "🌐<b>Сайт:</b>\npriisk-solov.ru\n🌐<b>Социальные
сети:</b>\nhttps://vk.com/priisk_solov\nhttps://t.me/priisksolov"
    await update.message.reply_text(text, parse_mode='HTML')

# Основная функция для запуска бота
def main() -> None:
    # Создаем экземпляр Application
```

Продолжение ПРИЛОЖЕНИЯ В

```
application = Application.builder().token(TOKEN).build()
# Обработчики команд
application.add_handler(CommandHandler("start", start))
application.add_handler(CommandHandler("about_company", about_company))
application.add_handler(CommandHandler("vacancies", vacancies))
application.add_handler(CommandHandler("news", news))
application.add_handler(CommandHandler("contacts", contacts))

# Обработчик сообщений
application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND,
handle_message))

# Запуск бота
application.run_polling()

# Обработка текстовых сообщений
async def handle_message(update: Update, context: CallbackContext) -> None:
    text = update.message.text.strip()

    if text == 'О компании':
        await about_company(update, context)
    elif text == 'Вакансии':
        await vacancies(update, context)
    elif text == 'Новости':
        await news(update, context)
    elif text == 'Контакты':
        await contacts(update, context)
    elif text == '📄 Краткая информация о компании':
        await company_info(update, context)
    elif text == '📅 История и достижения':
        await history(update, context)
    elif text == '🌐 География добычи':
        await geography(update, context)
    elif text == '🔧 Технологии и оборудование':
        await technologies(update, context)
    elif text == '📺 Презентационное видео':
        await presentation_video(update, context)
    elif text == '📁 Документы':
        await documents(update, context)
    elif text == '📌 Актуальные вакансии':
        await current_vacancies(update, context)
    elif text == '❓ Часто задаваемые вопросы':
        await faq(update, context)
    elif text == '➡️ Связаться с HR-отделом':
        await contact_hr(update, context)
    elif text == '📧 Отправить резюме':
```


Продолжение ПРИЛОЖЕНИЯ В

```
await send_resume(update, context)
elif text in ['⏮ Назад', 'Назад']:
    await back_to_main(update, context)
else:
    await update.message.reply_text('Я вас не понял. Пожалуйста, выберите один из пунктов меню.')

if __name__ == '__main__':
    main()
```

ПРИЛОЖЕНИЕ Г

Листинг блока YaMap

```
<iframe
src="https://yandex.ru/mapwidet/v1/?um=constructor%3Ad5194b048a91d1d2edb8634b9dad0376
966d08f3324198bdcad7b3045f589541&
source=constructor"
width="600"
height="450"
frameborder="0"
></iframe>
```