

План работы над проектом

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Дальневосточный федеральный университет

Институт математики и компьютерных технологий

Департамент информационных и компьютерных систем

Отчёт о практическом задании по предмету Программная инженерия

Сервис для поиска работы в сфере творческой деятельности (Notjob)

Просина Анна Алексеевна,

Проказюк Савва Юрьевич

Шестовец Егор Васильевич

гр. Б9121-09.03.03пикд

г. Владивосток, 2023 г.

Содержание

- [Содержание](#)
- [Аннотация](#)
- [1. Введение](#)
 - [1.1 Глоссарий](#)
 - [1.2 Описание предметной области](#)
 - [1.3 Неформальная постановка задачи](#)
 - [1.4 Обзор существующих методов решения](#)
- [2. Требования к окружению](#)
 - [2.1. Требования к аппаратному обеспечению](#)
 - [2.2. Требования к программному обеспечению](#)
 - [2.3. Требования к пользователям](#)
- [3. Архитектура системы](#)
- [4. Спецификация данных](#)
 - [4.1 Описание формата или структуры данных](#)
 - [4.2 Описание сущности](#)
 - [4.3 Описание протокола](#)
- [5. Функциональные требования](#)
- [6. Требования к интерфейсу](#)
- [7. Прочие требования](#)
 - [7.1 Требования к надёжности](#)
 - [7.2 Требования к безопасности](#)
 - [7.3 Требования к производительности](#)
- [8. Проект](#)
 - [8.1 Средства реализации](#)

- [8.2 Структуры данных](#)
 - [8.3 Модули и алгоритмы](#)
 - [8.4 Стандарт кодирования](#)
 - [8.5 Проект интерфейса](#)
 - [9. Реализация и тестирование](#)
 - [Заключение](#)
 - [Список литературы](#)
-

Аннотация

"Notjob" - это мультиплатформенный веб-сервис, предназначенный для поиска исполнителей и работодателей с запросом на выполнение работы с творческой направленностью. "Notjob" предоставляет возможность поиска работы и заказов, продвижения своих творческих проектов, их хранения и демонстрации.

1. Введение

Данная работа посвящена разработке проекта в области предоставления услуг связанных с творческой деятельности.

В современном мире существует растет спрос на творческий вид деятельности, появляется всё больше людей которые хотят монетизировать свою творческую деятельность (предоставить услугу творческого направления). Однако отсутствие единой площадки по предоставлению таких услуг осложняет задачу, реализация подобной площадки поможет решить следующие проблемы:

- Поиск партнеров для создания собственного проекта
 - Поиск сотрудников для работы в фирме
 - Поиск специалистов для личного заказа
-

1.1 Глоссарий

Мессенджер - функция для обмена мгновенными сообщениями.

Производитель - пользователь который предоставляет свои услуги.

Потребитель - пользователь который пользуется услугами другого пользователя(производителя).

Юзер - это пользователь. Чаще всего имеется в виду пользователь компьютерной программы, сайта, устройства и т.п.

1.2 Описание предметной области

Последние 10 лет в России наблюдается тенденция по поиску работы в интернете. Многие специалисты предпочитают искать сотрудников на различных интернет-площадках из-за удобства данного способа.

Но именно творческим людям, таким как художники, музыканты, писатели и дизайнеры, действительно тяжело найти работу даже в интернете. Это имеет множество причин:

- Высокая конкуренция;
- Ограниченность мест работы;
- Низкая заработная плата;
- Предвзятость со стороны большинства работодателей в отношении творческих профессий.

В связи с этими факторами, производители не находят потребителя и также в обратную сторону.

Крупные сервисы по поиску работы и работников адаптированы под широкий спектр предоставляемых услуг, по этой причине, чтоб найти команду или одноразовые задачи творческим людям нужно одновременно быть зарегистрированным на множестве площадок и надеяться что они найдут подходящую им работу и их возьмут на её исполнение.

Отсутствие централизованной площадки для предоставления именно творческих услуг - та проблема, которую данный проект предполагает решить. Соответственно, цель проекта - решение проблемы трудоустройства работников многих творческих специальностей и работодателей нуждающихся в таких кадрах путем создания сервиса, адаптированного под поиск потребителя и производителя данного типа работы.

1.3 Неформальная постановка задачи

Требуется реализовать проект позволяющий взаимодействовать потребителям и производителям творческих услуг, имеющий следующие возможности:

- Наличие удобного и современного каталога объявлений, а так же форму для заполнения объявления производителем.
- Наличие фильтров для быстрой и удобной навигации по каталогу.
- Наличие личного кабинета.
- Наличие обратной связи, продукт должен предоставлять возможность потребителю оставлять отзывы и оценки производителю услуг.
- Удобная коммуникация при помощи встроенного бесплатного мессенджера для улучшения связи между потребителем и производителем.

1.4 Обзор существующих методов решения

На данный моментг, для решения проблемы описанной в разделе 1.2, используются или глобальные социальные сети, в которых создаются соответствующие чаты и / или сообщества для конкретных социальных групп, или сайты имеющие слишком узкую направленность. Подход с использованием социальных сетей довольно неэффективен, так как пользователи могут не узнать или потерять связь с теми или иными сообществами / чатами, и соответственно, не иметь возможности решить проблемы. Наличие большого количества узкольконаправленных сайтов так же недейственно из-за неудобств доставляемых пользователю. Настоящий проект может решить данную проблему, так как вся необходимая информация будет публиковаться в одном приложении и будет видна всем желающим людям.

Популярные сайты имеющие схожий функционал:

- VK - Самая крупная социальная сеть на территории России
- yesjob - Сервис для поиска творческой работы
- grintern - Сервис для поиска работы молодыми специалистами в городе Москва
- artoffers.ru - Сервис для поиска творческой работы, в котором собраны предложения со сторонних площадок
- theory & practice -
- Telegram - Крупный мессенджер
- event.ru - Площадка связанная с организацией мероприятий, также предоставляет вакансии творческой направленности
- illustrators.ru - Сайт предоставляющий возможность поиска работы для художников
- Дикторы.com - Сайт предоставляющий возможность поиска дикторов

	yesjob	Дикторы.com	illustrators.ru	artoffers.ru	VK	Telegram	event.ru	theory & practice	grintern	Реализуемый проект
Мессенджер	-	+	-	-	+	+	-	-	-	+
Фильтры	-	-	-	-	-	-	-	+	-	+
Личный кабинет	-	-	+	-	-	-	+	+	+	+
Отзывы	-	-	+	-	-	-	-	-	-	+
Широкий спектр творческих направлений	-	-	-	+	-	-	-	-	-	+

2. Требования к окружению

2.1. Требования к аппаратному обеспечению

Требования к аппаратному обеспечению для юзера веб-сайта на компьютере:

- Процессор: Intel Core i3 или аналогичный
- Оперативная память: 4 Гб
- Видеокарта: совместимая с DirectX 11 и более поздней версии
- Жесткий диск: 250 Гб
- Разрешение экрана: 1024 x 768 дюймов и выше

Требования к аппаратному обеспечению для юзера веб-сайта на телефоне:

- Процессор: Apple A9 / Samsung Exynos 8890 или более поздняя версия процессора
- Оперативная память: 2 Гб для Android и iOS
- Хранение: 16 Гб или более

- Разрешение экрана: 720 x 1280 пикселей или больше
- Интернет соединение: 3G или выше, Wi-Fi.

Это минимальные требования, чтобы гладко использовать веб-сайт на компьютере или телефоне. Если требования выше указанных, то пользователь сможет использовать сайт с еще большей производительностью.

2.2. Требования к программному обеспечению

Для пользователя веб-сайта на компьютере требуется следующее программное обеспечение:

- Браузер (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Opera и т.д.) для доступа к веб-сайту.
- Операционная система (Windows, MacOS, Linux и т.д.), чтобы запустить браузер.

Для пользователя веб-сайта на телефоне требуется:

- Мобильный браузер (Google Chrome, Safari, Firefox, Samsung Internet и т.д.), чтобы получить доступ к веб-сайту.
- Операционная система (iOS, Android, Windows Phone и т.д.) для запуска браузера.

2.3. Требования к пользователям

Работодатели/заказчики творческой работы

Функции:

- поиск и выбор исполнителей работы,
- размещение заказов, оплата за выполненную работу.

Доступные возможности:

- создание профиля, поиск работников,
- публикация заказов, общение с исполнителями,
- оплата за выполненную работу.

Требования:

- наличие финансовых средств для оплаты работ,
- знание правил оформления заказов,
- умение правильно описывать требования к работе,
- умение вести переговоры.

Исполнители

Функции:

- поиск работы,
- выполнение заказов,
- общение с работодателями.

Доступные возможности:

- создание профиля,
- поиск заказов,
- расширение портфолио,
- общение с работодателями,
- получение оплаты за выполненную работу

Требования:

- наличие творческого потенциала и опыта в выбранной области,
- знание требований к выполнению заказов,
- умение работать с различными программами и инструментами.

3. Архитектура системы

Система веб-сайта по фрилансу может быть разделена на следующие подсистемы:

1. Клиентская подсистема - это фронтенд часть, которая отображается на экране пользователя через браузер. Она отвечает за отображение интерфейса веб-сайта и взаимодействие с пользователем. Ее функции включают:

- Отображение дизайна на экране пользователя;
- Обработка ввода данных от пользователя;
- Отправка данных на серверную подсистему.

2. Серверная подсистема - это бэкенд часть, которая выполняется на сервере и обрабатывает запросы клиентской подсистемы. Она отвечает за обработку данных, хранение информации и взаимодействие с базой данных. Ее функции включают:

- Обработка запросов от клиентской подсистемы;
- Обработка и хранение данных пользователей;
- Взаимодействие с базой данных;
- Аутентификация пользователей.

Общие требования для всех подсистем веб-сайта по фрилансу основаны на принципах безопасности, надежности и удобства использования. Некоторые из требований включают:

- Защита данных пользователей
- Обеспечение быстрой работы веб-сайта;
- Удобный и понятный интерфейс для пользователей;
- Автоматическая аутентификация пользователей;
- Возможность поиска и фильтрации данных;
- Резервное копирование данных и обновление системы;
- Поддержка множества языков и устройств.

Ввиду использования django в данном проекте для проектирования приложения была выбрана следующая архитектура:

MVT (Model View Template) — это шаблон проектирования программного обеспечения. Django MVT — это набор из трех важных компонентов: Model, View и Template (модель, шаблон и представление). Модель помогает обрабатывать базу данных. Это уровень доступа к данным, который обрабатывает данные.

Шаблон — это уровень представления, который полностью обрабатывает часть пользовательского интерфейса. Представление используется для выполнения бизнес-логики и взаимодействия с моделью для переноса данных и отображения шаблона.

Хотя Django следует шаблону MVC, но сохраняет свои собственные соглашения. Таким образом, управление осуществляется самим фреймворком.

Нет отдельного контроллера, и полное приложение основано на представлении модели и шаблоне. Вот почему это называется приложением MVT.

На изображении выше пользователь запрашивает ресурс у Django, Django работает как контроллер и проверяет доступный ресурс в URL-адресе.

Если URL-адреса совпадают, вызывается представление, которое взаимодействует с моделью и шаблоном, оно отображает шаблон. Django отвечает пользователю и отправляет шаблон в качестве ответа.

4. Спецификация данных

4.1 Описание формата или структуры данных

Веб-сайт использует различные форматы файлов данных, такие как:

- HTML файлы для создания страниц сайта
- CSS файлы для оформления страниц (шаблоны, стили и т.д.)
- JavaScript файлы для динамического изменения контента страниц и улучшения пользовательского опыта
- Python файлы для создания backend архитектуры сайта

4.2 Описание сущности

В нашем проекте реализованы следующие сущности:

- Сущность "Пользователь":

Описание структуры данных пользователя в базе данных сайта:

- ID
 - Тип: число;
 - Не изменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется автоматически при регистрации.
- Пароль
 - Тип: строка;
 - Ограничение: не менее 8 символов, не более 128;
 - Изменяемый;
 - Не уникальный;
 - Обязательно;
 - Заполняется пользователем при регистрации.
- Имя пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - Обязательно;
 - Заполняется пользователем при регистрации.
- Логин
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Неизменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется пользователем при регистрации.
- Электронная почта
 - Тип: строка;
 - Ограничение: не более 254 символов;
 - Неизменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется пользователем при регистрации.
- Пол
 - Тип: строка;
 - Изменяемый;
 - Не уникальный;
 - Обязательно;
 - Выбирается пользователем при регистрации.
- Дата рождения
 - Тип: дата;
 - Ограничение: не менее 8 символов, не более 128;
 - Изменяемый;
 - Не уникальный;
 - Обязательно;
 - Заполняется пользователем при регистрации.
- Дата регистрации
 - Тип: дата;
 - Неизменяемый;
 - Не уникальный;

- Обязательно;
- Заполняется автоматически при регистрации.
- Активность
 - Тип: логическая переменная;
 - Изменяемый;
 - Не уникальный;
 - Обязательно;
 - Отображает статус аунтификации пользователя.

2. Сущность "Профиль":

Описание структуры данных профиля в базе данных сайта:

- ID
 - Тип: число;
 - Не изменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется автоматически.
- Рейтинг
 - Тип: число;
 - Не изменяемый;
 - Не уникальный;
 - Обязательно;
 - Заполняется автоматически.
- Имя пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - Обязательно.
- Аватар пользователя
 - Тип: изображение;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - Обязательно.
- Описание профиля
 - Тип: строка;
 - Изменяемый;
 - Не уникальный;
 - Необязательно.
- Ссылка на твиттер пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - НеОбязательно.
- Ссылка на фейсбук пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - НеОбязательно.
- Ссылка на инстаграмм пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - НеОбязательно.

3. Сущность "Услуга":

Описание структуры данных услуги в базе данных сайта:

- ID
 - Тип: число;
 - Не изменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется автоматически
- Наименование услуги
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 255;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Заполняется пользователем при создании услуги.
- Описание услуги
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 255;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Заполняется пользователем при создании услуги.
- Категория услуги
 - Тип: строка;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Выбирается пользователем при создании услуги.
- Цена услуги
 - Тип: число;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Заполняется пользователем при создании услуги.
- Рейтинг пользователя
 - Тип: число;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Вычисляется на основании отзывов других пользователей.

4. Сущность "Пост":

Описание структуры данных поста в базе данных сайта:

- ID
 - Тип: число;
 - Не изменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется автоматически
- Наименование поста
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 255;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Заполняется пользователем при создании поста.
- Описание поста
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 255;

- Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Заполняется пользователем при создании поста.
- Категория поста
 - Тип: строка;
 - Изменяемый;
 - Неуникальный;
 - Обязательно;
 - Выбирается пользователем при создании поста.
- Имя пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - Обязательно.

5. Сущность "Отзыв":

Описание структуры данных отзыва в базе данных сайта:

- ID
 - Тип: число;
 - Не изменяемый;
 - Уникальный;
 - Обязательно;
 - Заполняется автоматически
- Текст отзыва
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 255;
 - Изменяемый;
 - Неуникальный;
 - Обязательно.
- Рейтинг услуги
 - Тип: число;
 - Изменяемый;
 - Неуникальный;
 - Обязательно.
 - Вычисляется на основании отзывов других пользователей.
- Дата регистрации
 - Тип: дата;
 - Неизменяемый;
 - Не уникальный;
 - Обязательно.
 - Заполняется автоматически.
- Имя пользователя
 - Тип: строка;
 - Ограничение: не менее 1 символа, не более 50;
 - Изменяемый;
 - Не уникальный;
 - Обязательно.

4.3 Описание протокола

1. Протокол передачи данных на клиент-серверной стороне:

HTTP (HyperText Transfer Protocol) - протокол, которым пользуются веб сайты. Он используется для передачи гипертекстовых документов, таких как HTML, CSS и JavaScript, между веб-сервером и веб-клиентом (обычно браузером). Базовый протокол более низкого уровня, используемый HTTP, это TCP/IP.

HTTP состоит из запросов и ответов, передаваемых между веб-клиентом и веб-сервером. Запрос содержит заголовок и тело сообщения, которые содержат информацию о том, какой ресурс необходимо получить с сервера и какие параметры запроса должны быть использованы. Ответ содержит заголовок и тело сообщения, который содержит запрошенные данные. При этом, заголовки ответа могут содержать дополнительную информацию, такую как код состояния, который сообщает веб-клиенту о том, был ли запрос успешно выполнен или произошла ошибка.

Список сообщений протокола HTTP:

1. Запрос - используется для передачи информации от веб-клиента к веб-серверу, включая метод запроса (например, GET, POST), URI-адрес запрашиваемого ресурса, заголовки запроса и тело сообщения (если есть).
2. Ответ - используется для передачи информации от веб-сервера к веб-клиенту, включая код состояния (например, 200 OK, 404 Not Found), заголовки ответа и тело сообщения (если есть).
3. Ошибка - используется для сообщения об ошибке в процессе выполнения запроса, включая код состояния, заголовки ответа и тело сообщения (если есть).

В непредвиденных ситуациях, таких как непредвиденное прекращение функционирования одной из сторон или разрыв канала связи, протокол HTTP может привести к ошибкам и искажению данных. Для устранения этих проблем введены дополнительные механизмы, такие как переподключение, обнаружение ошибок и повторная отправка сообщений.

2. Протокол передачи данных внутри системы:

Протокол передачи данных внутри системы (API) для веб-приложения - это стандартизированный набор правил, специфических для веб-окружения, которые определяют, как приложения на стороне клиента (браузер) и на стороне сервера должны взаимодействовать друг с другом.

Для веб-приложений наиболее распространенные протоколы API включают в себя REST (Representational State Transfer) и SOAP (Simple Object Access Protocol). REST - это протокол передачи данных, основанный на использовании стандартных HTTP-методов, таких как GET, POST, PUT и DELETE, для запросов на сервер и получения ответов от него. При реализации нашего проекта мы будем использовать REST API.

API для веб-приложений может использовать различные форматы данных, такие как JSON, XML или HTML, и различные методы передачи данных, такие как AJAX (Asynchronous JavaScript and XML) и WebSockets.

С помощью REST API можно создавать такие функциональные возможности веб-приложения, как:

- Регистрация и авторизация пользователей.
- Создание, редактирование и удаление профилей пользователей.
- Поиск и фильтрация проектов по различным параметрам.
- Поиск и отбор фрилансеров для выполнения задач.
- Создание, редактирование и удаление задач для фрилансеров.
- Отслеживание выполнения задач и оплаты работ.

5. Функциональные требования

Система должна позволять **Администратору**:

- Использовать весь доступный функционал зарегистрированных пользователей;
- В отдельной для него вкладке просматривать данные об услугах, пользователях, отзывах и т.д.;
- Изменять данные услуг, пользователей и т.д.

Система должна позволять **Незарегистрированному пользователю**:

- Создание нового аккаунта (регистрация);
- Поиск интересующего профиля пользователя;
- Просмотр актуального списка услуг.
- Фильтрация услуг по категориям.

Система должна позволять **Пользователю**:

- Авторизация;
 - Удаление своего аккаунта;
 - Поиск интересующего профиля пользователя;
 - Просмотр актуального списка услуг;
 - Фильтрация услуг по категориям;
 - Загрузка своих работ на сайт и их воспроизведение в профиле;
 - Редактирование профиля;
 - Коммуникация с другими заказчиками;
 - Просмотр актуального списка постов;
 - Добавление новых постов.
-

6. Требования к интерфейсу

Интерфейс реализуемого приложения является Web-интерфейсом. В данном случае будет рассматриваться только клиентская часть, именно она осуществляет взаимодействие между Пользователем и сервером.

Обозначим требования к веб интерфейсу:

Обице требования к веб интерфейсу

- Интерфейс должен быть интуитивно понятным, простым в использовании и привлекательным для пользователя.
- Должна быть поддержка различных языков и национальных стандартов.
- Интерфейс должен быть адаптивным и работать на различных устройствах с разными разрешениями экранов.
- Должна быть поддержка различных браузеров и операционных систем.

Специальные требования к веб интерфейсу

- Интерфейс должен быть доступным при низкой скорости интернета для пользователей из удаленных районов.

Цветовая гамма в основе цветовой гаммы лежат чёрный, белый и синий цвета, а также отдельные элементы ярких цветов для привлечения внимания.

Особенности интерфейса

- В профиле пользователя находится его портфолио выраженное изображениями которые пользователь может выложить;
- Возможность поиска профилей зарегистрированных пользователей для просмотра отзывов о них и их портфолио;
- Каталог услуг возможностью фильтрации по категориям;

7. Прочие требования

Среди прочих требований необходимо отметить следующие:

1. **Быстродействие и производительность.**

Необходимо обеспечить высокую скорость загрузки страницы и быстрое выполнение всех операций на сайте. Долгие задержки и медленная работа могут отпугнуть потенциальных заказчиков и негативно повлиять на репутацию исполнителя.

2. **Надежность и безопасность.**

Сайт должен быть защищен от взлома, а также обеспечивать безопасность и конфиденциальность пользовательских данных. Надежность работы сайта является одним из важнейших критериев для привлечения и удержания заказчиков.

3. **Удобство использования.**

Система управления должна быть интуитивно понятной и удобной в использовании. Для этого необходимо обеспечить понятную структуру сайта, легкий поиск интересующей информации, удобный интерфейс и быстрый доступ к необходимым функциям.

4. **Адаптивность и мобильность.**

Сайт должен адаптироваться к различным устройствам и браузерам. В настоящее время все больше пользователей используют мобильные устройства для доступа к интернету, поэтому важно обеспечить корректное отображение сайта на мобильных устройствах.

5. **SEO оптимизация.**

Сайт должен быть оптимизирован для поисковых систем, чтобы привлечь больше трафика и увеличить количество заказов. Необходимо обеспечить правильную структуру для страниц сайта, использование ключевых слов, мета-тегов и других инструментов для оптимизации контента.

7.1 Требования к надёжности

Проанализируем возможные риски связанные с надёжностью сайта, выявим вероятность их возникновения, последствия и убытки которые они принесут.

1. **Отказ оборудования**

Вероятность: средняя.

Последствия: простой сайта на период устранения проблемы, возможная потеря клиентов и престижа.

Убытки: потеря прибыли и репутации.

2. **Обнаружение ошибок разной степени тяжести**

Вероятность: высокая.

Последствия: возможны проблемы с функционированием сайта, негативный опыт клиентов и уменьшение престижа.

Убытки: потери в репутации, потеря прибыли.

3. **Утеря или повреждение данных**

Вероятность: низкая.
Последствия: потеря заказов, негативный опыт клиентов.
Убытки: потеря прибыли и репутации.

4. **Физическое нарушение безопасности**

Вероятность: низкая.
Последствия: несанкционированный доступ к данным клиентов и исполнителей, утечка конфиденциальных данных.
Убытки: потери в репутации и правовых проблемы.

5. **Резкое изменение экономических условий или производственного процесса**

Вероятность: высокая.
Последствия: существенное снижение спроса на услуги, ограничение финансовых возможностей клиентов.
Убытки: потеря прибыли и возможность оказания услуг.

7.2 Требования к безопасности

Для предотвращения угроз, связанных с злоумышленным несанкционированным доступом к веб-сайту, необходимо рассмотреть риски и возможные векторы атаки, а также применить соответствующие меры безопасности. Различные уязвимости и атаки могут привести к различным угрозам безопасности сайта, таким как:

1. Доступ на чтение к конфиденциальным данным: злоумышленники могут получить доступ к конфиденциальной информации, такой как личные данные пользователей, платежные данные и т.д.

Векторы атаки:

- SQL-инъекции
- Межсайтовое подделывание запросов (CSRF)
- Кража куки-файлов и сессий

Изменения функциональности:

- Применение механизмов валидации входных данных
- Подключение систем защиты от SQL-инъекций
- Использование криптографии при передаче данных

2. Доступ на запись к открытым, но важным данным: злоумышленники могут изменять данные на сайте, такие как страницы, посетителей и другие открытые данные.

Векторы атаки:

- Подделка данных формы и GET-параметров
- Межсайтовое подделывание запросов (CSRF)
- Слабые пароли и учётные записи

Изменения функциональности:

- Реализация систем защиты от CSRF и XSS
- Настройка механизмов контроля доступа
- Использование сильных паролей и хранение их в безопасной форме (хеширование и соль)

3. Перехват управления критическими процессами: злоумышленники могут получить доступ к контроллерам и системам управления сайтом и изменить настройки, что может привести к серьезным последствиям.

Векторы атаки:

- Уязвимости в системах управления контентом (CMS)
- Необходимость обновления программного обеспечения
- Использование слабых паролей и учётных записей администраторов

Изменения функциональности:

- Регулярное обновление программного обеспечения
- Подключение системы мониторинга и безопасности
- Используются сильные пароли и аутентификация с двухфакторной аутентификации
- Установка лимитов на отправку запросов

4. Разрушение данных и вандализм: злоумышленники могут случайно или намеренно уничтожить данные на сайте, что может привести к неисправности и неполадкам.

Векторы атаки:

- SQL-инъекции
- Межсайтовое подделывание запросов (CSRF)
- Атаки на сервер, выполнение команд и другие вредоносные программы

Изменения функциональности:

- Создание регулярных бэкапов данных
- Использование антивирусного ПО и систем мониторинга безопасности
- Использование механизмов защиты от вредоносных программ
- Создание тестовых сред для отладки

5. Временное нарушение работы системы без разрушения данных (Denial-of-Service): злоумышленники могут перегрузить сайт, нанося урон его работе и бизнесу.

Векторы атаки:

- DDos-атаки
- Некорректная работа сервера
- Перегрузка запросами и данных

Изменения функциональности:

- Использование CDNs
- Балансировка нагрузки
- Кэширование контента и оптимизация сайта, уменьшение количества запросов файерволом
- Ограничение сетевых соединений

Комбинация этих мер поможет защитить веб-сайт от основных угроз безопасности и избежать серьезных проблем. Вывести сайт из строя может любая из угроз, и поэтому защита должна быть комплексной. Она должна включать в себя предотвращение угроз, защиту от атак, мониторинг и реагирование на происходящее, управление безопасностью и принятие согласованных мер для рисков и угроз безопасности.

7.3 Требования к производительности

Определим диапазоны количественных параметров размеров задач, для решения которых предназначена система, в том числе объём базы данных, количество одновременно работающих пользователей и скорость поступления внешних сигналов. Система предназначена для решения задач в следующих диапазонах количественных параметров:

1. Объём базы данных: Минимальный объём: 1 ГБ Максимальный объём: 1 ТБ
2. Количество одновременно работающих пользователей: Минимальное количество: 10 Максимальное количество: 10000
3. Скорость поступления внешних сигналов для веб сайта: Минимальная скорость: 10 запросов в секунду Максимальная скорость: 100000 запросов в секунду

Минимальные требования к количественным показателям эффективности системы:

1. Время отклика: Предельно допустимое значение: менее 1 секунды Ожидаемое среднее значение: менее 500 миллисекунд
2. Скорость выполнения транзакций: Предельно допустимое значение: более 100 транзакций в секунду Ожидаемое среднее значение: более 500 транзакций в секунду
3. Время обработки запроса: Предельно допустимое значение: менее 1 секунды Ожидаемое среднее значение: менее 500 миллисекунд
4. Объём используемой оперативной памяти и дискового пространства: Предельно допустимое значение: менее 80% от максимального объёма Ожидаемое среднее значение: менее 50% от максимального объёма

Если часть показателей зависит от внешних факторов, необходимо оптимизировать их комбинацию для достижения оптимальных показателей эффективности системы. Например, для качественного TCP/IP соединения необходимо наличие высокоскоростного интернет-соединения с широкой шириной канала.

Ожидаемые реальные показатели в различных условиях:

- При минимальной нагрузке:
 - время отклика - менее 100 миллисекунд,
 - скорость выполнения транзакций - более 1000 транзакций в секунду,
 - время обработки запроса - менее 50 миллисекунд,

- объём используемой оперативной памяти и дискового пространства - менее 20% от максимального объёма.
- При максимальной нагрузке:
 - время отклика - менее 1 секунды,
 - скорость выполнения транзакций - более 500 транзакций в секунду,
 - время обработки запроса - менее 500 миллисекунд,
 - объём используемой оперативной памяти и дискового пространства - менее 80% от максимального объёма.

8. Проект

8.1 Средства реализации

1. Backend

- Python

Python - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Он поддерживает несколько парадигм программирования, структурированное, процедурное, объектно ориентированное и функциональное программирование. Python имеет обширный набор стандартных библиотек, а также имеет систему управления пакетами `pip` для загрузки библиотек, которые были созданы сообществом.

- Django

Django - это свободный фреймворк для веб-приложений на языке программирования Python. Фреймворк был создан с целью избавить разработчиков от повторения основных этапов разработки приложений, таких как управление базой данных, управление сессиями и URL-маршрутизация. Django имеет гибкую архитектуру, которая позволяет быстро создавать веб-приложения различной сложности, используя принципы DRY (Do not Repeat Yourself) и MVC (Model-View-Controller).

Он также имеет множество встроенных компонентов, таких как ORM (Object-Relational Mapping), шаблонизатор и систему аутентификации. Django имеет большое сообщество разработчиков, которые создают и поддерживают множество сторонних библиотек и расширений для фреймворка, таких как пакеты для работы с API, обработки изображений и административных панелей, что делает его одним из наиболее популярных и универсальных фреймворков в мире веб-разработки

- SQLite

SQLite — компактная встраиваемая СУБД. Среди преимуществ данной СУБД:

1. Простота использования - SQLite не требует от пользователя настройки сервера, установки и запуска какого-либо ПО, что делает его крайне простым в использовании.
2. Легковесный и быстрый - SQLite обычно работает быстрее, чем другие СУБД, такие как MySQL или PostgreSQL, так как не требует отправки запросов на удаленный сервер.
3. Надежный - SQLite является очень стабильной базой данных, с фиксированной структурой и проверками целостности данных - гарантией сохранности ваших данных даже в случае неожиданного сбоя программного обеспечения.
4. Безопасный - благодаря проекту SQLCipher, SQLite может быть зашифрован и обеспечивать безопасное хранение данных.

2. Frontend

- HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки»)

HTML – стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

- CSS (Cascading Style Sheets «каскадные таблицы стилей»)

CSS – формальный язык декорирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML).

- JavaScript

JavaScript – современный мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

3. Общие стредства: - Visual Studio Code

8.2 Структуры данных

8.3 Модули и алгоритмы

8.4 Стандарт кодирования

Стандарт кодирования — набор правил и соглашений, которые описывают базовые принципы оформления программного кода, используемого совместно группой разработчиков. Цель использования стандарта — упрощение восприятия программного кода человеком, сокращение нагрузки на память и зрение при чтении программы. Для разработки и серверной, и клиентской части использовался один язык программирования, поэтому разницы в стандарте кодирования не было.

8.5 Проект интерфейса

- Цвета: Основными цветами выбраны белый (# FFFFFF) — цвет фона панелей и кнопок, чёрный (#000000) — цвет шрифта и контуров, оранжевый (#057d9f) — цвет активных элементов.
- Шрифт: Шрифт overpass используется для обычного текста, а шрифт roboto для заголовков

Определим страницы нашего веб приложения, среди них:

- Главная страница.**

Главная страница имеет ознакомительный характер, содержит информацию о данном проекте. Нажав на кнопку "Заказы" вы переместитесь на соответсвующую вкладку с каталогом имеющихся услуг, нажав на кнопку "Лента" вы переместитесь на страницу, содержащую ленту постов других пользователей. Нажатие на кнопку "Присоединиться" переместит вас на страницу регистрации, а кнопка "Вход" на страницу с аунтификацией.

- Страница регистрации.**

Страница предназначена для регистрации пользователя. Содержит следующие поля обязательные для заполнения: Имя пользователя, логин, почта, дата рождения, пол, пароль, повторение пароля. Если у вас уже есть созданный аккаунт, то вы можете перейте на страницу авторизации нажав Sign in или же кнопку "Вход".

После регистрации вы должны будете подтвердить свою почту. Нажав на кнопку "Change email" вы сможете выбрать другую почту для верификации, кнопка "Resend verifying email" отправляет ещё одно сообщение на выбранную почту.

- Страница входа.**

Страница предназначена для авторизации пользователя. Содержит следующие поля обязательные для заполнения: Почта или логин и пароль.

После нажатия на кнопку *Забыли пароль?* возникнет всплывающее окно которое запросит вашу почту.

Затем вы должны будете потвердить свою почту, на неё придет код восстановления.

Если код оказался верным, то вы сможете указать новый пароль.

- Профиль**

На странице профиля будет отображена ваша основная информация. Вы можете изменять своё изображение и описание, выкладывать новые изображения. На этой странице находится выход из аккаунта.

- Услуги**

Создание услуги осуществляется через отдельную вкладку. Указав все данные вы сможете опубликовать новую услугу.

При нажатии на кнопку заказы вас перенесет в каталог услуг.

При нажатии на услугу откроется отдельное окно с подробным описанием данной услуги.

При нажатии кнопки "Отозваться" вас перенесет на новую вкладку и вы сможете написать сообщение заказчику(при условии вашей авторизации).

- **Диалоги.**

Страница диалогов содержит все начатые диалоги пользователя. Нажав на выбранный диалог можно перейти на страницу коммуникации с собеседником.

9. Реализация и тестирование

Объем написанного кода на 27.05.2023: CSS – 7.29 Кб, 400 строк HTML – 34.1 Кб, 782 строки JS – 5,42 Кб, 186 строк Python – 12,34 Кб, 382 строки Объем автоматически сгенерированного кода: 7,89 Кб, 398 строк (генерация кода происходит под воздействием js файлов и Django фреймворка) Количество страниц 9 Количество модулей – 5 Количество форм – 7 Количество всплывающих окон - 7 Количество выпадающих списков - 2 Количество автоматических тестов – 1 (тестирование системы регистрации) Фактическое количество пользователей – 0 (выгрузка на сервер будет осуществлена 9.06)

Сервис прошел проверку на валидацию данных. Система рейтинга, количество отзывов о пользователе, комментариев под постами, подписчиков и подписок на аккаунте было успешно протестировано по методу "черного ящика".

Заключение

В результате проделанной работы над проектом "NotJob" был создан мультиплатформенный веб-сервис, который предоставляет удобный и простой способ поиска исполнителей и работодателей с запросом на выполнение работы с творческой направленностью. Данный сервис позволяет пользователям найти нужную работу или заказ, а также продвигать свои творческие проекты, хранить и демонстрировать их. Особое внимание было уделено удобству использования и интерфейсу, дизайну и функциональности. Были использованы передовые технологии и инструменты разработки веб-приложений и баз данных. В результате проделанной работы над проектом "NotJob" мы создали очень полезный и удобный инструмент для творческих людей, который сможет упростить их жизнь и помочь достичь успеха в своей работе.

Список литературы

1. Методические указания по подготовке и защите работ по программированию. Федеральное агентство по образованию РФ. Дальневосточный федеральный университет. Кафедра информатики, математического и компьютерного моделирования. Кленин А. С. Владивосток, 2002-2015.
2. Figma “Прототип интерфейса” URL: <https://www.figma.com/file/5tyN3lDsXRzrd7Ymo098AE/Untitled?type=design&node-id=0-1>
3. Официальная документация Django URL: <https://docs.djangoproject.com/en/4.2/>
4. Современный учебник JavaScript URL: <https://learn.javascript.ru>
5. Django 4 By Example: Build powerful and reliable Python web applications from scratch, 4th Edition 2022 Antonio Mele
6. Gritern сайт с вакансиями для студентов, выпускников и молодых специалистов URL: <https://grintern.ru>
7. artoffers сервис, где собраны предложения о творческой работе с сайтов и бирж трудоустройства. URL: <https://artoffers.ru>
8. illustrators.ru сайт URL: <https://illustrators.ru>
9. yesjob.ru URL: <http://yesjob.ru>
10. creately - визуальный инструмент с возможностями построения диаграмм URL: <https://app.creately.com/d/oGv3G3ME2ze/edit>
11. METANIT.COM Введение в django URL: <https://metanit.com/python/django/1.1.php>
12. Habr.com SQLite URL: <https://habr.com/ru/articles/149356/>
13. Помошник python URL: <https://pythonpip.ru/django/django-mvt>