**Пример Технического Задания на Разработку Веб-приложения «Фриланс маркетплейс»**

**1. Введение**

1.1 **Цель документа**

Этот документ описывает требования к разработке веб-приложения «Фриланс маркетплейс», предназначенного для автоматизации взаимодействия между заказчиками и фрилансерами, управления проектами, задачами и сроками выполнения.

1.2 **Актуальность и назначение проекта**

Этот документ описывает требования к разработке веб-приложения «Фриланс маркетплейс», предназначенного для автоматизации взаимодействия между заказчиками и фрилансерами, управления проектами, задачами и сроками выполнения.

1.3 **Основные пользователи системы**

Система предназначена для использования сотрудниками компании:

* Заказчики — создание заказов, управление проектами и контроль выполнения.
* Фрилансеры — поиск заказов, выполнение задач, управление портфолио, предоставление услуг.
* Администраторы — модерация контента и пользователей, управление платформой.

**2. Цели и задачи проекта**

* 1. **Основные цели**
* Создание удобной платформы для взаимодействия заказчиков и фрилансеров.
* Автоматизация процессов управления заказами и проектами.
* Обеспечение прозрачности и удобства для всех участников

.2.2 **Конкретные задачи**

* Разработка функционала для регистрации, создания и выполнения заказов.
* Реализация системы уведомлений о статусах заказов.
* Организация рейтинговой системы для пользователей.
* Разработка интерфейса для фильтрации заказов по параметрам.

*Пример:* Система должна позволять заказчику размещать заказы с описанием, стоимостью и сроками выполнения, а фрилансеры могут откликаться на эти заказы. Также дизайнер может выставлять свои услуги чтобы заказчики могли найти его по определенным услугам и заказать на прямую у данного дизайнера.

**3. Требования к системе**

3.1 **Функциональные требования**

* Регистрация и авторизация заказчиков/фрилансеров с валидацией.
* Размещение заказов: создание, редактирование, удаление заказов заказчиками.
* Поиск заказов: возможность фильтрации и сортировки для фрилансеров.
* Поиск дизайнеров: возможность фильтрации и сортировки по категориям для нахождения нужной услуги и определенного дизайнера.
* Система откликов: отправка предложений от фрилансеров заказчикам.
* Система уведомлений: отправка уведомлений о статусе заказов, сроках выполнения и нарушении дедлайнов.
* Рейтинг и отзывы: возможность оставлять отзывы о сотрудничестве.

*Пример*: После выполнения задачи заказчик оставляет отзыв о фрилансере, который отображается в его профиле. Также и фрилансер может оставить отзыв на заказчика.

3.2 **Нефункциональные требования**

* Производительность: время отклика интерфейса не более 1 секунды.
* Масштабируемость: поддержка 10000+ пользователей без снижения производительности.
* Безопасность: шифрование данных через SSL, поддержка двухфакторной аутентификации.

**4. Требования к пользовательскому интерфейсу**

4.1 **Основные экраны**

* Главная страница: навигация, общий поиск, отображение популярных фрилансеров, отображение популярных услуг, фильтрация.
* Личный кабинет: информация о профиле, статистика, управление заказами, отзывы, портфолио(фрилансер).
* Детали заказа: описание проекта, срок выполнения, бюджет, отзывы.
* Панель администратора: модерация пользователей, просмотр статистики.

4.2 **Навигация**

Система должна включать меню навигации с разделами: "Главная", "Мои заказы", "Биржа", "Профиль", "Дропсы".

4.3 **Юзабилити**

* Простой и понятный интерфейс для быстрого доступа к задачам и их статусу.
* Использование визуальных индикаторов: красный — просроченные, желтый — в процессе, зеленый — завершенные.

Пример: Например на проектах будут стоять иконки или.

**5. Требования к технической реализации**

5.1 **Языки и технологии**

* **Backend**: PHP Laravel.
* **Frontend**: React.js, Vue.js .
* **База** **данных**: MySQL.

5.2 **Архитектура системы**

* Микросервисная архитектура с REST API для взаимодействия между модулями.
* Хранение данных в реляционной базе данных.

5.3 **Интеграции**

* Внешние уведомления: Email и Push-уведомления
* Интеграция с системами авторизации, такими как Google OAuth.

**6. Требования к безопасности**

* **Шифрование** **данных**: HTTPS для всех передаваемых данных.
* **Аутентификация**: OAuth для входа через Google и Facebook.
* **Авторизация**: Разделение ролей (заказчики, фрилансеры, администраторы).

**7. Ограничения и допущения**

**7.1 Технические ограничения**

* Серверная часть должна размещаться на облачных платформах (например, AWS).

**7.2 Финансовые ограничения**

* Бюджет проекта:до 500 000 ₽.

**7.3 Сроки выполнения**

* Полная реализация проекта: 8 месяцев.

**8. Требования к тестированию и приемке**

**8.1 Типы тестирования**

* Функциональное тестирование: проверка основных функций.
* Нагрузочное тестирование: проверка работы при 10 000+ пользователях.
* Тестирование безопасности: анализ на уязвимости.

**8.2 Критерии приемки**

* Все основные функции реализованы и протестированы.

Пример: если выявляется более 3 критических ошибок в ходе тестирования, система не принимается.

**9. Требования к документации**

* Пользовательская документация: инструкции для работы с системой.
* Техническая документация: описание API, структура базы данных, руководство по развертыванию.

**10. План реализации**

**10.1 Этапы разработки**

1. Сбор требований и проектирование — 1 месяц.
2. Разработка прототипа — 1 месяц.
3. Полноценная разработка и тестирование — 2 месяца.
4. Внедрение и поддержка — 1 месяц.

**10.2 Ответственные лица**

* **Менеджер проекта:** Иванов И.И.
* **Разработчик:** Петров П.П.

**11. Приложения**

* **Приложение A:** Примеры экранов.
* **Приложение B:** Технические спецификации API.
* **Приложение C:** Схема базы данных.