**Техническое задание на разработку программы**

**«SharpRise» – интерактивный задачник по языку программирования C#**

# 1. Введение

## 1.1. Наименование программы

**Официальное наименование программного продукта: «SharpRise».**

**Полное наименование: "Веб-приложение для интерактивного изучения языка программирования C# с автоматической проверкой решений и системой группового обучения".**

## 1.2. Краткая характеристика области применения

**Программа предназначена для использования в образовательных учреждениях (вузах, колледжах, IT-курсах), а также для самостоятельного обучения программированию. Может применяться:**

**- В учебном процессе как основной или дополнительный инструмент обучения**

**- Для подготовки к техническим собеседованиям**

**- В корпоративных тренинг-центрах IT-компаний**

# 2. Назначение и область применения

## 2.1. Основные цели и задачи разработки

**Цель: Создание универсальной образовательной платформы для эффективного изучения C# через практику.**

**Задачи:**

**1. Автоматизация процесса проверки задач**

**2. Организация системы дистанционного обучения**

**3. Создание инструментов для преподавателей**

# 4. Разработка аналитического модуля успеваемости

## 2.2. Описание целевой аудитории

**Основные группы пользователей:**

**| Группа | Характеристика | Требования к системе |**

**|--------|---------------|----------------------|**

**| Ученики | Студенты, начинающие программисты | Простой интерфейс, мгновенная проверка решений |**

**| Преподаватели | Преподаватели вузов, тренеры | Гибкие настройки курсов, инструменты анализа |**

**| Администраторы | Технические специалисты | Панель управления, системы мониторинга |**

**2.3. Аналоги и конкуренты**

**Анализ существующих решений:**

**1. Stepik**

**- Преимущества: популярная платформа**

**- Недостатки: слабая поддержка C#, нет специализированного функционала**

**2. LeetCode**

**- Преимущества: хорошая система задач**

**- Недостатки: отсутствие группового обучения**

**3. Codeforces**

**- Преимущества: система соревнований**

**- Недостатки: не адаптирован для обучения**

**Конкурентные преимущества SharpRise:**

**- Специализация на C#**

**- Интеграция с образовательным процессом**

**- Уникальная система автоматической проверки**

# 3. Требования к программе

## 3.1. Требования к функциональным характеристикам

### 3.1.1. Функционал для ученика

**1. Система решения задач**

**- Редактор кода с подсветкой синтаксиса**

**- Встроенный компилятор**

**- Мгновенная проверка решений**

**- Подробный разбор ошибок**

**2. Теоретические материалы**

**- Структурированная база знаний**

**- Примеры кода**

**- Видеолекции**

**3. Личный кабинет**

**- Статистика прогресса**

**- Дневник достижений**

**- Настройки профиля**

### 3.1.2. Функционал для преподавателя

**1. Управление курсами**

**- Создание учебных программ**

**- Формирование групп**

**- Назначение заданий**

**2. Проверка работ**

**- Система комментариев**

**- Ручная проверка**

**- Выставление оценок**

**3. Аналитика**

**- Отчеты по успеваемости**

**- Выявление проблемных тем**

**- Сравнительные графики**

### 3.1.3. Функционал администратора

**1. Управление пользователями**

**- Регистрация/блокировка**

**- Назначение ролей**

**- Модерация контента**

**2. Техническое администрирование**

**- Мониторинг сервера**

**- Резервное копирование**

**- Обновление системы**

## 3.2. Требования к надежности

### 3.2.1. Обеспечение отказоустойчивости

**1. Резервирование данных**

**- Ежедневное бэкапирование**

**- Хранение резервных копий на отдельном сервере**

**2. Защита от сбоев**

**- Автоматическое восстановление после падения**

**- Мониторинг работоспособности**

**3. Безопасность**

**- Защита от SQL-инъекций**

**- Шифрование передаваемых данных**

**- Двухфакторная аутентификация**

### 3.2.2. Восстановление после сбоев

**- Программные сбои: не более 30 минут**

**- Аппаратные сбои: не более 4 часов**

**- Критические аварии: не более 24 часов**

## 3.3. Технические требования

### 3.3.1. Серверная часть

**- Процессор: 8 ядер, 3.0 ГГц**

**- Оперативная память: 32 ГБ**

**- Хранилище: 1 ТБ SSD**

**- ОС: Ubuntu Server 22.04 LTS**

### 3.3.2. Клиентская часть

**- Минимальные требования:**

**- Процессор: 2 ядра, 2.0 ГГц**

**- Память: 4 ГБ**

**- Браузер: Chrome 90+**

# 4. Требования к программной документации

## 4.1. Состав документации

**1. Техническая документация**

**- Техническое задание**

**- Руководство системного администратора**

**- API-документация**

**2. Пользовательская документация**

**- Руководство преподавателя**

**- Руководство ученика**

**- FAQ и справочная система**

**3. Проектная документация**

**- Диаграммы UML**

**- ER-диаграммы базы данных**

**- Технические спецификации**

# 5. Технико-экономические показатели

## 5.1. Ожидаемая эффективность

**1. Для образовательных учреждений**

**- Сокращение времени проверки работ на 70%**

**- Увеличение успеваемости на 25%**

**2. Для коммерческих организаций**

**- Снижение затрат на обучение новых сотрудников**

**- Ускорение процесса адаптации**

**3. Для самостоятельных пользователей**

**- Экономия на репетиторах и курсах**

**- Гибкий график обучения**

# 6. Стадии и этапы разработки

## 6.1. Детальный план разработки

# 7. Порядок контроля и приемки

## 7.1. Критерии приемки

**1. Полное соответствие ТЗ**

**2. Отсутствие критических ошибок**

**3. Выполнение всех обязательных функций**

## 7.2. Процедура приемки

**1. Подписание акта выполненных работ**

**2. Передача исходного кода**

**3. Обучение администраторов**

## 8. Приложения

**1. ГОСТ 19.201-78**

**2. Диаграмма классов**

**3. Макеты интерфейсов**

**Исполнитель:**

**Симанович М.А., группа Т-295**

**Заказчик:**

**ЧУО "Колледж бизнеса и права"**

**Дата составления:**

**«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024 г.**