Создание сайта системы учета баллов — это многоэтапный процесс, который включает в себя проектирование, разработку, тестирование и запуск. Для этого потребуется выполнить несколько ключевых задач и действий, разделённых на несколько этапов.

**1. Подготовительный этап**

**1.1 Анализ требований**

- Определение целей и задач: что именно система будет отслеживать (баллы за определённые действия, достижения, бонусы, штрафы и т.д.).

- Целевая аудитория: кто будет пользоваться системой (пользователи, педагоги, администратор).

- Функциональные требования:

- Регистрация и авторизация пользователей (авторизация пользователей будет происходить следующим образом: педагогам в их профиль будут приходить заявки на подключение к их кванту. Например: ученик зарегистрировался и выбрал обучение в кванте VR/AR. Алесе Андреевне придёт подтверждение заявки).

- Личный кабинет пользователя с отображением текущих баллов.

- История начислений и списания баллов.

- Механизмы начисления и списания баллов.

- Административный интерфейс для управления пользователями и системой.

- \*\*Нефункциональные требования\*\*:

- Безопасность (шифрование паролей, защита от SQL-инъекций).

- Масштабируемость и производительность.

- Адаптивность интерфейса.

**1.2 Проектирование**

- \*\*Разработка структуры данных\*\*:

- Базы данных для хранения информации о пользователях; начислениях и списаниях баллов.

- Структуры для хранения логов событий.

- Проектирование таблиц и их связей.

- \*\*Создание прототипа интерфейса\*\*:

- Разработка макетов для интерфейса пользователя, педагога, администратора.

- Создание макетов всех страниц: главная страница, личный кабинет, страница количества баллов и их начисления (когда, сколько, за что), история, расписание занятий, все кванты (направления).

**2. Этап разработки**

**2.1 Разработка бэкенда (серверной части)**

- \*\*Выбор стека технологий\*\*:

- Язык программирования для серверной логики (например, Python, Node.js, Ruby, PHP).

- Выбор фреймворка для разработки (Django, Flask, Express.js и т.д.).

- Настройка базы данных (например, MySQL, PostgreSQL, MongoDB).

*- \*\*Разработка REST API или GraphQL для взаимодействия с фронтендом\*\*:*

*- Эндпоинты для регистрации пользователей, входа в систему.*

*- Эндпоинты для отображения текущих баллов, истории начислений.*

*- Эндпоинты для администрирования (добавление/удаление пользователей, управление баллами).*

- \*\*Реализация логики начисления и списания баллов\*\*:

- Создание базы для хранения данных о начислениях и списаниях.

*- Логика автоматического начисления баллов за действия пользователей.*

- Логика списания баллов.

***2.2 Разработка фронтенда (клиентской части)***

*- \*\*Выбор фреймворка для фронтенда\*\* (например, React, Angular, Vue.js).*

*- \*\*Дизайн интерфейса\*\*:*

*- Разработка страниц: домашняя страница, регистрация, личный кабинет, история начислений.*

*- Адаптивный дизайн для работы на мобильных и десктопных устройствах.*

*- \*\*Интеграция с бэкендом\*\*:*

*- Подключение к API для отображения и управления данными пользователей.*

*- Реализация пользовательских форм (регистрация, вход, редактирование данных).*

**2.3 Разработка административной панели**

- Интерфейс для администраторов для:

- Управления пользователями (создание, редактирование, удаление).

- Управления баллами (ручное начисление/списание).

- Просмотра статистики по пользователям (баллы, действия).

**3. Этап тестирования**

**3.1 Модульное тестирование**

- Тестирование отдельных функций, таких как регистрация, начисление и списание баллов, авторизация.

**3.2 Интеграционное тестирование**

- Проверка взаимодействия различных частей системы (бэкенда и фронтенда, базы данных и серверной логики).

**3.3 Тестирование безопасности**

- Проверка на уязвимости:

- Защита от SQL-инъекций.

- Шифрование паролей.

- Защита от CSRF и XSS атак.

**3.4 Юзабилити-тестирование**

- Оценка удобства интерфейса для конечных пользователей.

- Проводится с фокус-группой или отдельными пользователями для получения отзывов и улучшения интерфейса.

**4. Этап запуска**

**4.1 Размещение сайта на сервере**

- Настройка хостинга или облачного решения (например, AWS, Google Cloud, DigitalOcean).

- Развертывание базы данных и бэкенда.

- Размещение фронтенда (например, с использованием CDN).

**4.2 Настройка домена и SSL-сертификата**

- Настройка домена (например, через Cloudflare или другой провайдер).

- Установка SSL-сертификата для обеспечения безопасности.

**4.3 Запуск сайта в продакшн**

- Миграция базы данных на сервер.

- Тестирование всех функций на продакшн-сервере.

**5. Этап пост-запуска**

**5.1 Мониторинг и аналитика**

- Настройка инструментов мониторинга (например, Google Analytics, Sentry).

- Анализ посещаемости и поведения пользователей.

**5.2 Обратная связь и улучшения**

- Сбор отзывов от пользователей.

- Регулярные обновления и улучшения функционала.

**5.3 Обслуживание и поддержка**

- Регулярное обновление безопасности.

- Поддержка пользователей (например, через службу поддержки).

**Дополнительные задачи**

*- Разработка мобильного приложения (если это необходимо).*

*- Интеграция с внешними сервисами (например, для начисления баллов за действия, через API партнёров).*

*- Оптимизация SEO (если сайт должен быть видимым в поисковых системах).*

Этот список задач и действий можно адаптировать в зависимости от сложности системы и её требований.