

Техническое задание на создание турнира по футболу и волейболу

1. Введение

Целью данного проекта является разработка платформы для организации и проведения турниров по футболу и волейболу. Платформа будет состоять из серверной и клиентской частей, обеспечивающих функционал для регистрации участников, создания турниров, управления матчами, отображения результатов и статистики.

2. Требования к функционалу

2.1. Общие требования

- Поддержка двух видов спорта: футбол и волейбол.
- Возможность создания нескольких турниров одновременно.
- Ролевая модель: администратор, участник, зритель.
- Мультиплатформенность: веб-приложение (клиентская часть) и серверная часть (API).

2.2. Серверная часть (Backend)

- **Язык программирования**: Python (Django/Flask/FastAPI), Node.js, или другой подходящий язык.
- **База данных**: PostgreSQL/MySQL/MongoDB.
- **API**: RESTful API или GraphQL.
- **Функционал**:
 1. Регистрация и авторизация пользователей (JWT или OAuth).
 2. Управление турнирами:
 - Создание, редактирование, удаление турниров.
 - Настройка правил для каждого вида спорта (количество участников, формат турнира: групповой этап, плей-офф и т.д.).
 3. Управление командами и участниками:
 - Регистрация команд.
 - Добавление/удаление участников в команды.

4. Управление матчами:

- Создание расписания матчей.
- Внесение результатов матчей.
- Автоматический расчет турнирной таблицы.

5. Статистика:

- Отображение статистики по командам и игрокам.
- История матчей.

6. Уведомления:

- Уведомления о начале матчей, изменениях в расписании, результатах.

7. Администрирование:

- Панель администратора для управления пользователями, турнирами и матчами.

2.3. Клиентская часть (Frontend)

- **Технологии**: React.js, Vue.js или Angular.

- **Функционал**:

1. Главная страница:

- Список активных турниров.
- Краткая информация о предстоящих матчах.

2. Страница турнира:

- Информация о турнире (название, даты, правила).
- Турнирная таблица.
- Расписание матчей.
- Результаты матчей.

3. Страница команды:

- Состав команды.
- Статистика команды и игроков.

4. Личный кабинет:

- Для участников: управление своими командами, просмотр своих матчей.
- Для администраторов: управление турнирами и матчами.

5. Регистрация и авторизация.

6. Адаптивный дизайн для мобильных устройств.

3. Нефункциональные требования

- ****Производительность****: система должна поддерживать до 10 000 пользователей одновременно.
- ****Масштабируемость****: возможность добавления новых видов спорта в будущем.
- ****Безопасность****: защита данных пользователей, HTTPS, валидация входных данных.
- ****Кроссплатформенность****: поддержка современных браузеров (Chrome, Firefox, Safari, Edge).

4. Этапы разработки

4.1. Подготовительный этап

- Анализ требований.
- Проектирование архитектуры системы.
- Выбор технологий и инструментов.

4.2. Разработка серверной части

- Настройка базы данных.
- Реализация API.
- Тестирование API (unit-тесты, интеграционные тесты).

4.3. Разработка клиентской части

- Верстка интерфейса.
- Интеграция с API.
- Тестирование интерфейса.

4.4. Тестирование и отладка

- Функциональное тестирование.

- Нагрузочное тестирование.
- Исправление ошибок.

4.5. Запуск и поддержка

- Развертывание на production-сервере.
- Мониторинг и техническая поддержка.

5. Требования к дизайну

- Минималистичный и современный дизайн.
- Использование корпоративных цветов (если есть).
- Удобная навигация и интуитивно понятный интерфейс.

6. Сроки и бюджет

- Сроки: 3-6 месяцев (в зависимости от сложности и команды).
- Бюджет: оценивается после уточнения требований и выбора технологий.

7. Дополнительные требования

- Документация: написание технической документации для API и пользовательской документации.
- Локализация: поддержка нескольких языков (если необходимо).