**Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования**

**«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА АНАЛИЗА ДАННЫХ И ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Направление: 09.03.03 – Прикладная информатика

КУРСОВАЯ РАБОТА

**Веб-приложение психолого-педагогического сопровождения лиц с ограниченными возможностями здоровья**

Студент 3 курса

группы 09-254

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хамитова А.Р.

Научный руководитель

ученая степень, ученое звание,

должность

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахметшина Д.И.

Казань-2025

[**Введение 4**](#_k8i9q61mjfl7)

[**1. Анализ предметной области и теоретико-методологические вопросы психолого-педагогического сопровождения лиц с ОВЗ 6**](#_n7kemd4cupvu)

[1.1. Сущность, цели и задачи психолого-педагогического сопровождения лиц с ограниченными возможностями здоровья 6](#_7cxomsgypq6t)

[1.2. Применение информационно-коммуникационных технологий и веб-приложений в системе поддержки лиц с ОВЗ 9](#_c7ejt326i74b)

[1.3. Обзор существующих цифровых решений для психолого-педагогического сопровождения 12](#_x07jfa8vpbrx)

[**2. Проектирование веб-приложения для психолого-педагогического сопровождения лиц с ОВЗ 14**](#_1ez058s4r5qe)

[2.1. Определение требований к программному продукту 14](#_eflwtd28s0ea)

[2.2. Обоснование выбора средств разработки и технологий 18](#_whdsf16bjnek)

[2.3. Проектирование архитектуры веб-приложения 20](#_p79hzob1zfa9)

[2.4. Проектирование функциональных модулей 22](#_5xb5mfnfluhi)

[2.4.1. Проектирование главной страницы 22](#_co4h3tnieaod)

[2.4.2. Проектирование модуля развивающих мини-игр 23](#_wygnaurq1i3m)

[2.4.3. Проектирование модуля курсов 23](#_bduu0esp9i75)

[2.4.4. Проектирование модуля реабилитационных центров 23](#_c60zmkck8vrc)

[2.4.5. Проектирование модуля тестирования 24](#_5342qd9e70yx)

[2.4.6. Проектирование личных кабинетов 24](#_1yqi4k89i2fq)

[2.4.7. Проектирование интеграции с Telegram-ботом 24](#_kytfuj8q09uz)

[**3. Разработка веб-приложения психолого-педагогического сопровождения лиц с ОВЗ 25**](#_op5ia9jmg05t)

[3.1. Описание процесса и этапов разработки 25](#_vy45qvyiif0k)

[3.2. Реализация серверной части (Backend) 26](#_2f23o5s6ozfl)

[3.3. Реализация клиентской части (Frontend) 30](#_jtzwo6jzt68w)

[3.4. Реализация функциональных модулей 32](#_pcdz8s7jwv9f)

[3.4.1. Разработка главной страницы 32](#_vh0dreo5zkao)

[3.4.2. Разработка модуля развивающих мини-игр 33](#_300urjt94yhj)

[3.4.3. Разработка модуля списка курсов 36](#_23zem94n9f8y)

[3.4.4. Разработка модуля списка реабилитационных центров 36](#_ndje22bwq8uc)

[3.4.5. Разработка модуля тестирования 37](#_affpszevz19m)

[3.4.6. Разработка личных кабинетов родителя и специалиста 39](#_2c9erwbb6d6r)

[3.4.7. Реализация интеграции с Telegram-ботом и Telegram-каналом 41](#_ca4yykua464p)

[**4. Тестирование программного продукта (веб-приложения) 42**](#_d4qxm7ikelbx)

[4.1. Планирование и организация тестирования 42](#_dwsotufdgjpy)

[4.2. Функциональное тестирование 42](#_vth9thb4fyhp)

[4.4. Тестирование пользовательского интерфейса 44](#_dfg1nm7o1qsb)

[**Заключение 45**](#_uirgrkq5n6pq)

[**Список литературы 47**](#_827plq5wgxlp)

# Введение

В современном обществе проблема психолого-педагогического сопровождения лиц с ограниченными возможностями (ОВЗ) приобретает особую актуальность. Обеспечение полноценной интеграции данной категории граждан в социальную, образовательную и профессиональную среду является одной из приоритетных задач и предметом пристального внимания научного сообщества. Эффективное психолого-педагогическое сопровождение призвано способствовать раскрытию личностного потенциала лиц с ОВЗ, их адаптации к условиям жизни, формированию необходимых компетенций и улучшению качества жизни в целом. Однако, несмотря на значительные усилия, предпринимаемые в этой области, сохраняется ряд вызовов, связанных с доступностью, современностью и индивидуализацией такой поддержки.

Традиционные формы оказания психолого-педагогической помощи, несмотря на их несомненную ценность, зачастую сталкиваются с ограничениями, обусловленными географической удаленностью, нехваткой квалифицированных специалистов в отдельных регионах, а также физическими барьерами, затрудняющими доступ лиц с ОВЗ к необходимым услугам. В этом контексте стремительное развитие информационно-коммуникационных технологий (ИКТ) открывает принципиально новые возможности для модернизации системы сопровождения, предлагая инновационные инструменты и подходы. Среди таких инструментов особое место занимают веб-приложения, способные обеспечить дистанционное взаимодействие, предоставить доступ к широкому спектру информационных и методических ресурсов, а также создать персонализированную и адаптивную среду для развития и обучения.

Использование веб-приложений с сфере психолого-педагогического сопровождения лиц с ОВЗ позволяет преодолеть многие из существующих барьеров, предлагая гибкие и масштабные решения. Такие приложения могут включать диагностические модули, коррекционно-развивающие программы, консультационные платформы для взаимодействия со специалистами, а также образовательный контент. Таким образом, разработка и внедрение специализированных веб-приложений становится важным направлением, способствующим повышению эффективности и доступности психолого-педагогического сопровождения.

Актуальность выбранной темы обусловлена, с одной стороны, насущной потребностью в совершенствовании системы поддержки лиц с ОВЗ, а с другой – значительным потенциалом разработанного веб-приложения для решения этой задачи.

Объектом исследования выступает процесс психолого-педагогического сопровождения лиц с ОВЗ.

Предметом исследования является разработанное веб-приложение.

Целью курсовой работы является разработка и описание веб-приложения для психолого-педагогического сопровождения лиц с ОВЗ.

Для достижения поставленной цели были определены следующие задачи:

1. Изучить теоретические основы психолого-педагогического сопровождения лиц с ОВЗ и роль ИКТ в этой области.
2. Сформулировать требования к веб-приложению для сопровождения лиц с ОВЗ.
3. Спроектировать архитектуру и интерфейсы веб-приложения.
4. Разработать и реализовать веб-приложение в соответствии с проектом.
5. Описать функционал и особенности разработанного веб-приложения.
6. Определить практическую значимость и перспективы применения веб-приложения.

Структура курсовой работы состоит из введения, трех глав, заключения, списка использованных источников.

# Анализ предметной области и теоретико-методологические вопросы психолого-педагогического сопровождения лиц с ОВЗ

## 1.1. Сущность, цели и задачи психолого-педагогического сопровождения лиц с ограниченными возможностями здоровья

Психолого-педагогическое сопровождение представляет собой целенаправленную, системно организованную деятельность, ориентированную на создание благоприятных условий для развития, обучения, воспитания и успешной социализации личности. Это понятие шире, чем просто оказание помощи: оно подразумевает активное взаимодействие всех участников образовательного процесса и направлено на мобилизацию внутренних ресурсов самого человека для преодоления трудностей и достижения личностно значимых целей. Сопровождение – это, прежде всего, содействие индивиду в принятии оптимальных решений на его жизненном пути и в процессе развития.

В рамках психологического сопровождения можно выделить следующие виды: психологическая помощь, психологическая поддержка и психологическое обеспечение.

Ключевыми сущностными характеристиками психолого-педагогического сопровождения являются динамичность и наличие последовательных этапов, начиная от диагностического изучения и заканчивая анализом достигнутых результатов. Системность сопровождения проявляется в комплексном подходе к решению возникающих проблем. Важнейшей характеристикой является личностно-ориентированная направленность, при которой основное внимание уделяется индивидуальным особенностям, актуальным потребностям и потенциальным возможностям каждого человека. Неотъемлемой частью успешного сопровождения является активное вовлечение самого субъекта сопровождения и его ближайшего окружения в этот процесс.

В специальном образовании оформлены модели психологической помощи детям с проблемами в развитии. Каждый из подходов имеет свою научную специфику и вносит свой вклад в разработку организационно-методических основ психологического сопровождения лиц с ОВЗ: концепция комплекса параллельных служб в системе образования (О.Н. Усанова); интеграционные идеи о «психологизации», психологической составляющей в педагогической, социальной, диагностической, реабилитационной, медицинской и логопедической работе (И.И. Мамайчук); модели специальной психологической помощи детям с ЗПР на ранних этапах онтогенеза.

Цели психолого-педагогического сопровождения можно рассматривать на нескольких уровнях. Стратегической, или общей, целью является создание условий, способствующих полноценному развитию личности, ее успешной адаптации и интеграции в общество. К числу более конкретных целей можно отнести:

– оказание содействия в преодолении возникающих образовательных, коммуникативных и социальных барьеров;

– формирование жизненно важных компетенций, включая навыки социального взаимодействия;

– стимулирование развития когнитивных процессов, эмоционально-волевой сферы и коммуникативных умений;

– осуществление коррекции и компенсации имеющихся нарушений и дефицитов в развитии;

– обеспечение психологической поддержки, способствующей формированию адекватной самооценки и устойчивой учебной и социальной мотивации.

Для достижения этих целей можно выделить следующие задачи:

1. Проведение комплексной психолого-педагогической диагностики для выявления индивидуальных особенностей развития, определения актуальных потребностей и потенциальных трудностей.
2. Разработка и последующая реализация индивидуализированных программ сопровождения, включающих образовательные маршруты, коррекционно-развивающие занятия и другие мероприятия.
3. Оказание консультативной помощи всем субъектам образовательного процесса: обучающимся, их родителям или законным представителям, а также педагогическим работникам.
4. Создание и поддержание благоприятного, безопасного и развивающего психологического климата в образовательной организации и семейной среде.
5. Организация информационно-просветительской деятельности, направленной на повышение уровня психолого-педагогической культуры родителей и педагогов.
6. Осуществление систематического мониторинга динамики развития личности и оценка эффективности реализуемых мер сопровождения.

Чтобы сопровождение было действительно эффективным, необходимо тесное и хорошо скоординированное взаимодействие между специалистами из разных областей. Это означает, что психологи, педагоги, дефектологи, логопеды, медицинские работники и сотрудники социальных служб должны работать сообща. Также важно, чтобы у лиц с ОВЗ и их близких была необходимая информация о том, к каким специалистам обратиться и как получить помощь. Конечная цель такого комплексного сопровождения – это не просто помочь человеку освоить учебную программу. Гораздо важнее максимально развить его жизненные навыки, подготовить к самостоятельной и как можно более полной жизни, а также обеспечить его успешное включение в общество наравне с другими.

На современном этапе развития психолого-педагогического сопровождения и обслуживания лиц с ОВЗ работа в основном сосредоточена в центрах различного типа (ПМПК, лекотеки и т.п.). Разрушение старой системы организации профессиональной помощи лицам с ОВЗ и внедрение идеи интеграции требуют переосмысления существующих моделей психологического сопровождения, специального образования и построения на этой основе системы психологической помощи и сопровождения лиц с ОВЗ с использованием информационно-коммуникационных технологий.

## 1.2. Применение информационно-коммуникационных технологий и веб-приложений в системе поддержки лиц с ОВЗ

Современный этап развития общества характеризуется глубоким проникновением информационно-коммуникационных технологий (ИКТ) во все сферы человеческой деятельности, включая образование, социальную поддержку и здравоохранение. Для лиц с ограниченными возможностями (ОВЗ) ИКТ открывают принципиально новые горизонты, выступая мощным инструментом компенсации нарушенных функций, средством преодоления барьеров и фактором, способствующим их полноценной интеграции в общество. Использование цифровых инструментов в психолого-педагогическом сопровождении данной категории лиц становится не просто актуальной тенденцией, а насущной необходимостью, позволяющей существенно повысить эффективность и доступность оказываемой помощи.

Роль ИКТ в поддержке лиц с ОВЗ многогранна. Во-первых, они способствуют преодолению информационных и коммуникационных барьеров. Например, для людей с нарушениями слуха, зрения или речи существуют специализированные программные и аппаратные средства, которые становятся незаменимыми помощниками в общении и получении информации. Во-вторых, ИКТ расширяют возможности для обучения и развития, предоставляют доступ к разнообразным образовательным ресурсам, адаптивным обучающим программам и дистанционным формам обучения. В-третьих, они способствуют социальной адаптации и интеграции, позволяя лицам с ОВЗ поддерживать социальные контракты, участвовать в общественной жизни и самореализовываться.

Особое место среди цифровых инструментов занимают веб-приложени, обладающие доступностью и интерактивностью. Доступность проявляется в нескольких аспектах. Физическая доступность: пользователь может получить доступ к ресурсам из любого места, что особенно важно для лиц с двигательными нарушениями или проживающих в отдаленных районах. Кроме того, современные веб-технологии позволяют создавать адаптивные интерфейсы, учитывающие потребности пользователей с различными нарушениями. Важным аспектом является и временная доступность: многие веб-приложения функционируют круглосуточно, предоставляя информацию и поддержку в удобное для пользователя время.

В отличие от традиционных пассивных форм получения знаний, интерактивные веб-инструменты вовлекают пользователя в активную деятельность через:

– игровые механики (геймификация);

– мультимедийный контент, интеграции текста, графики, аудио и видео;

– симуляции и виртуальные тренажеры;

– системы обратной связи;

– возможности персонализации.

В особенности использования веб-технологий в системе поддержки лиц с ОВЗ сходит платформенная независимость: доступ к веб-приложениям осуществляется через стандартный браузер, что делает их совместимыми с различными операционными системами и расширяет охват аудитории.

Во-вторых, централизованное обновление и управление контентом является преимуществом для разработчиков и администраторов системы. Все изменения и обновления вносятся на сервере, и пользователи автоматически получают доступ к актуальной версии приложения без необходимости скачивать и устанавливать обновления на свои устройства. Это обеспечивает единообразие и своевременность предоставления информации и функционала.

В-третьих, веб-технологии предоставляют широкие возможности для сбора и анализа данных о процессе использования приложения. Это позволяет отслеживать прогресс пользователей, выявлять трудности, с которыми они сталкиваются, и на основе этих данных адаптировать содержание и методики сопровождения, а также оценивать общую эффективность системы.

В-четвертых, веб-приложения могут служить платформой для организации дистанционного взаимодействия между лицами с ОВЗ, их родителями и специалистами. Это могут быть форумы для общения и обмена опытом, системы для проведения онлайн-консультаций, вебинары и групповые занятия. Такие формы работы особенно важны для обеспечения непрерывности сопровождения и поддержки семей, проживающих в удаленных регионах.

Несмотря на значительные преимущества, внедрение ИКТ и веб-приложений в систему поддержки лиц с ОВЗ сопряжено и с определенными вызовами. К ним относятся проблема «цифрового неравенства» (неравный доступ к технике и интернету), необходимость обучения пользователей и специалистов навыкам работы с цифровыми инструментами, обеспечение информационной безопасности и конфиденциальности данных, а также важность сохранения баланса между виртуальным и реальным взаимодействием, чтобы технологии дополняли, а не заменяли живое человеческое общение и поддержку.

Таким образом, информационно-коммуникационные технологии, и в частности веб-приложения, обладают огромным потенциалом для трансформации системы психолого-педагогического сопровождения лиц с ОВЗ. Их грамотное и целенаправленное использование, основанное на принципах доступности, интерактивности и персонализации, способно значительно расширить возможности для развития, обучения и социальной интеграции данной категории граждан, делая поддержку более гибкой, эффективной и ориентированной на индивидуальные потребности каждого человека.

## 1.3. Обзор существующих цифровых решений для психолого-педагогического сопровождения

В настоящее время цифровое пространство предлагает много решений, направленных на поддержку лиц с ограниченными возможностями здоровья (ОВЗ) и их семей а также специалистов, работающих в данной области. Анализ существующих веб-сайтов, платформ и мобильных приложений позволяет выявить достижения и ниши, требующие дальнейшего развития и интеграции.

Существующие цифровые решения можно условно классифицировать по их основной направленности и функционалу:

1. Информационные порталы и ресурсные центры. Данная категория включает веб-сайты, агрегирующие информацию о различных аспектах жизни с ОВЗ, законодательстве, доступных льготах, методиках реабилитации и абилитации, а также каталоги специализированных учреждений (например, порталы «овз.рф», «инклюзия-овз.рф»).
2. Платформы с развивающими играми и упражнениями. Это веб-сайты и мобильные приложения, предлагающие интерактивные игры и упражнения, направленные на развитие когнитивных функций (внимание, память, мышление), речи, мелкой моторики и других навыков у детей, в том числе с ОВЗ (например, «Мерсибо», «Маам.ру»).
3. Диагностические онлайн-платформы и приложения. Ресурсы, предлагающие прохождение различных психологических и педагогических тестов и опросников онлайн для детей, родителей или специалистов (например, «mirr.pro»).
4. Специализированные мобильные приложения узкой направленности. Существует множество мобильных приложений, разработанных для помощи при конкретных видах ОВЗ (например, приложения для альтернативной и дополнительной коммуникации (AAC), приложения для тренировки слухового восприятия, приложения-дневники для отслеживания состояния).

Анализ существующих цифровых решений показывает, что, несмотря на их многообразие, большинство из них решают лишь отдельные, узкоспециализированные задачи в области психолого-педагогического сопровождения лиц с ОВЗ. Наблюдается фрагментация сервисов: информация находится на одних ресурсах, развивающие игры – на других, диагностические инструменты – на третьих, а консультации специалистов – на четвертых. Это создает неудобства для пользователей (родителей, специалистов), которым приходится перемещаться между множеством платформ, самостоятельно собирая и анализируя разрозненную информацию.

Разрабатываемое веб-приложение призвано преодолеть указанные ограничения, предложив комплексный, интегрированный подход к психолого-педагогическому сопровождению лиц с ОВЗ. Его необходимость и новизна обосновываются следующими аспектами:

– интеграция разнопланового функционала;

– система персонализированных личных кабинетов;

– инновационное использование AI-технологий;

– формирование единой экосистемы поддержки.

# 2. Проектирование веб-приложения для психолого-педагогического сопровождения лиц с ОВЗ

## 2.1. Определение требований к программному продукту

Определение точных и полных требований является фундаментальным этапом в процессе проектирования и разработки любого программного продукта, в особенности такого социально значимого, как веб-приложение для психолого-педагогического сопровождения лиц с ОВЗ. Требования служат основой для всех последующих проектных решений, процесса разработки и тестирования, а также определяют критерии успешности и соответствия продукта ожиданиям пользователей. Требования к программному продукту принято разделять на функциональные и нефункциональные.

Функциональные требования описывают, что система должна делать, какие задачи выполнять и какие функции предоставлять пользователям. Для разрабатываемого веб-приложения определены следующие ключевые функциональные требования, сгруппированные по основным модулям и компонентам:

1. Общие системные функции и аутентификация:

1.1. Система должна обеспечивать возможность регистрации новых пользователей.

1.2. Зарегистрированные пользователи должны иметь возможность безопасного входа в систему (аутентификации) с использованием логина и пароля.

1.3. Система должна обеспечивать безопасный выход пользователя из сеанса работы.

1. Главная страница

2.1. Главная страница должна отображать общую информацию о приложении, его целях и возможностях.

2.2. Должны быть представлены интуитивно понятные элементы навигации для доступа ко всем основным разделам приложения (мини-игры, курсы, реабилитационные центры, тесты, профиль, ссылка на Telegram-бот/канал).

2.3. Может содержать новостной блок и анонсы актуальных событий, материалов.

1. Модуль развивающих мини-игр

3.1. Система должна отображать каталог доступных развивающих мини-игр.

3.2. Пользователь (ребенок через интерфейс родителя) должен иметь возможность выбрать и запустить любую игру из каталога.

1. Модуль «Список доступных курсов»:

4.1. Система должна отображать перечень доступных курсов (для родителей, специалистов, возможно, и для детей).

4.2. Для каждого курса должна отображаться подробная информация (описание, программа, ссылка на прохождение).

1. Модуль «Список реабилитационных центров»:

5.1. Система должна предоставлять доступ к базе данных реабилитационных центров.

5.2. Для каждого центра должна отображаться информация (название, ссылка на контакты).

1. Модуль тестирования:

6.1. Система должна предоставлять возможность прохождения тестов для детей: диагностические и развивающие тесты, адаптированные под возраст и возможные особенности.

6.2. Система должна предоставлять возможность прохождения тестов для родителей: тесты на выявление уровня родительского стресса, информированности об особенностях ребенка.

6.3. Система должна предоставлять возможность прохождения тестов для специалистов: профессиональные тесты.

6.4. Результаты пройденных тестов должны автоматически обрабатываться и сохраняться в системе.

1. Личный кабинет родителя:

7.1. Родитель должен иметь доступ к просмотру результатов тестов своего ребенка и собственных тестов.

7.2. В кабинете могут отображаться рекомендации по курсам, играм или центрам, сформированные на основе результатов тестов или интересов.

1. Личный кабинет специалиста:

8.1. Специалист должен иметь доступ к просмотру результатов тестов детей и родителей.

8.2. Должна быть возможность просмотра результатов собственных профессиональных тестов.

1. Интеграция с Telegram-ботом и Telegram-каналом:

9.1. Веб-приложение должно содержать хорошо заметную ссылку или иной способ перехода к Telegram-боту.

9.2. Telegram-бот (внешняя система, но связанная с приложением) должен быть способен отвечать на типовые вопросы родителей, используя базу знаний и ИИ-модель, и при необходимости направлять пользователя к соответствующим разделам веб-приложения или к специалистам.

9.3. Должна быть предусмотрена ссылка на информационный Telegram-канал, который ведут специалисты.

Нефункциональные требования описывают, как система должна работать, то есть ее качественные характеристики, атрибуты и ограничения. Они не менее важны, чем функциональные, так как напрямую влияют на удовлетворенность пользователей и общую жизнеспособность продукта. Для разрабатываемого приложения выделены следующие нефункциональные требования:

1. Производительность:

1.1. Время загрузки страниц веб-приложения не должно превышать 3-5 секунд при стабильном интернет-соединении.

1.2. Система должна обеспечивать быстрый отклик на действия пользователя (например, запуск игры, отправка формы теста) – не более 1-2 секунд.

1.3. Приложение должно быть способно обрабатывать запросы от ожидаемого количества одновременных пользователей без существенного снижения производительности.

1. Надежность и доступность:

2.1. Система должна быть доступна для пользователей 99.5% времени (высокий уровень аптайма).

2.2. Должны быть предусмотрены механизмы обработки ошибок и исключительных ситуаций, чтобы предотвратить сбои в работе и потерю данных.

1. Безопасность:

3.1. Должна быть обеспечена защита от несанкционированного доступа к данным пользователей и административным функциям системы.

3.2. Должны соблюдаться требования законодательства о защите персональных данных.

1. Удобство использования (Usability) и пользовательский опыт (UX):

4.1. Интерфейс приложения должен быть интуитивно понятным, логичным и простым в освоении для всех категорий пользователей, включая тех, кто имеет ограниченный опыт работы с компьютером.

4.2. Дизайн должен быть привлекательным, не перегруженным, с четкой визуальной иерархией.

4.3. Навигация по сайту должна быть последовательной и предсказуемой.

4.4. Система должна предоставлять пользователю адекватную обратную связь на его действия.

1. Масштабирование:

5.1. Архитектура приложения должна позволять наращивание функционала (добавление новых модулей, игр, тестов) и увеличение количества пользователей в будущем без кардинальной переработки системы.

1. Совместимость:

6.1. Веб-приложение должно корректно отображаться и функционировать в последних версиях основных современных веб-браузеров (например, Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).

6.2. Дизайн должен быть адаптивным, обеспечивая корректное отображение на различных устройствах (десктопы, планшеты, смартфоны).

1. Поддерживаемость (Maintainability):

7.1. Код приложения должен быть структурированным, хорошо прокомментированным и следовать общепринятым стандартам кодирования для облегчения его понимания, модификации и исправления ошибок.

7.2. Должна быть предусмотрена возможность легкого обновления контента (например, добавление новых курсов, центров, новостей).

Тщательная проработка и документирование этих функциональных и нефункциональных требований позволяет сформировать четкое видение конечного продукта и служит надежным ориентиром на всех этапах его жизненного цикла.

## 2.2. Обоснование выбора средств разработки и технологий

Выбор стека технологий и инструментов разработки является критически важным этапом проектирования, поскольку он напрямую влияет на скорость разработки, производительность, масштабируемость, безопасность и удобство поддержки будущего программного продукта. Для веб-приложения психолого-педагогического сопровождения лиц с ОВЗ, обладающего разнообразным функционалом – от интерактивных мини-игр до личных кабинетов и интеграции с Telegram-ботом – был сделан осознанный выбор в пользу современных, проверенных и хорошо зарекомендовавших себя технологий, соответствующих задачам проекта (рис. 1).

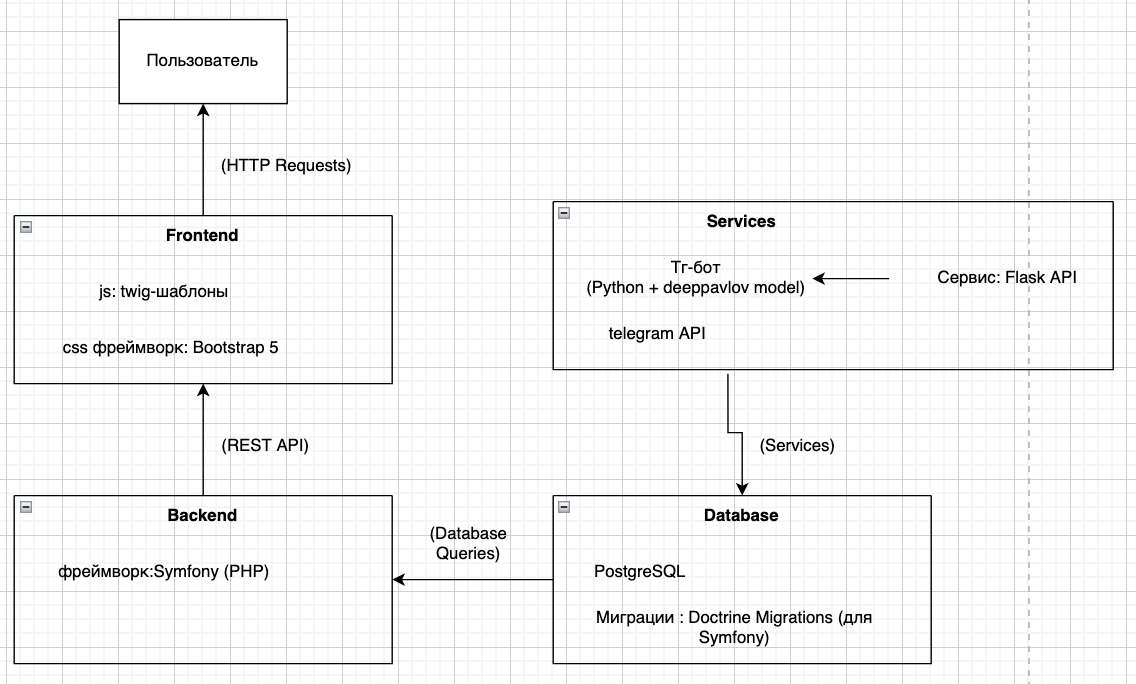


Рисунок 1. Функциональная схема веб-приложения

Для разработки пользовательского интерфейса были выбраны стандартные веб-технологии: HTML для структуры, CSS для стилизации и JavaScript для интерактивности. Чтобы ускорить создание адаптивного дизайна и обеспечить единообразие интерфейса, использовался CSS-фреймворк Bootstrap.

Серверная логика приложения реализована на языке PHP в связке с мощным фреймворком Symfony. PHP был выбран из-за его широкого распространения в веб-разработке и большого числа готовых решений. Symfony предоставляет структурированный подход к разработке, интегрированную ORM Doctrine для удобной работы с базой данных и инструменты для управления изменениями схемы базы данных через Doctrine Migrations, что упрощает командную работу и обеспечивает контроль версий структуры БД.

В качестве системы управления базами данных используется PostgreSQL. Эта реляционная СУБД выбрана за ее высокую надежность, стабильность, соответствие стандартам SQL, поддержку сложных запросов и транзакций, а также хорошую интеграцию с Doctrine ORM, что важно для обеспечения целостности и эффективного управления данными приложения.

Telegram-бот, служащий для первичной обработки запросов пользователей, разработан на языке Python благодаря наличию удобных библиотек для работы с Telegram Bot API и быстроте разработки на этом языке. Для связи между Python-ботом и основным PHP-бэкендом используется легковесное API на базе микрофреймворка Flask (Python), который выступает простым и эффективным мостом для передачи данных и команд, не перегружая основное приложение.

В процессе разработки используются стандартные для индустрии инструменты: система контроля версий Git для управления кодовой базой; подходящие среды разработки (IDE); а также менеджеры зависимостей Composer (для PHP) и npm/Yarn (для JavaScript).

## 2.3. Проектирование архитектуры веб-приложения

Приложение реализует классическую клиент-серверную архитектуру. Клиентская часть (Frontend): работает в браузере пользователя. Она отвечает за отображение пользовательского интерфейса, обработку пользовательского ввода и взаимодействие с серверной частью через HTTP-запросы (API). Разработана с использованием HTML, CSS, JavaScript. Серверная часть (Backend): является ядром приложения. Она обрабатывает запросы от клиента, реализует бизнес-логику, взаимодействует с базой данных и внешними сервисами. Разработана на PHP с использованием фреймворка Symfony. База данных (Database): хранит все данные приложения. Используется реляционная СУБД PostgreSQL.

В дополнение к основной клиент-серверной модели, система включает отдельный сервис для Telegram-бота, который взаимодействует с основным бэкендом через специально разработанное API. Этот Telegram-бот написан на Python и использует Flask для создания API-прослойки для общения с основным PHP-бэкендом.

База данных спроектирована с учетом необходимости хранения информации о пользователях, их ролях, детях (для родителей), тестах, результатах тестирования, курсах, реабилитационных центрах и развивающих играх. Основные сущности и их предполагаемые связи представлены на ER-диаграмме (Entity-Relationship Diagram) базы данных (рис. 2).

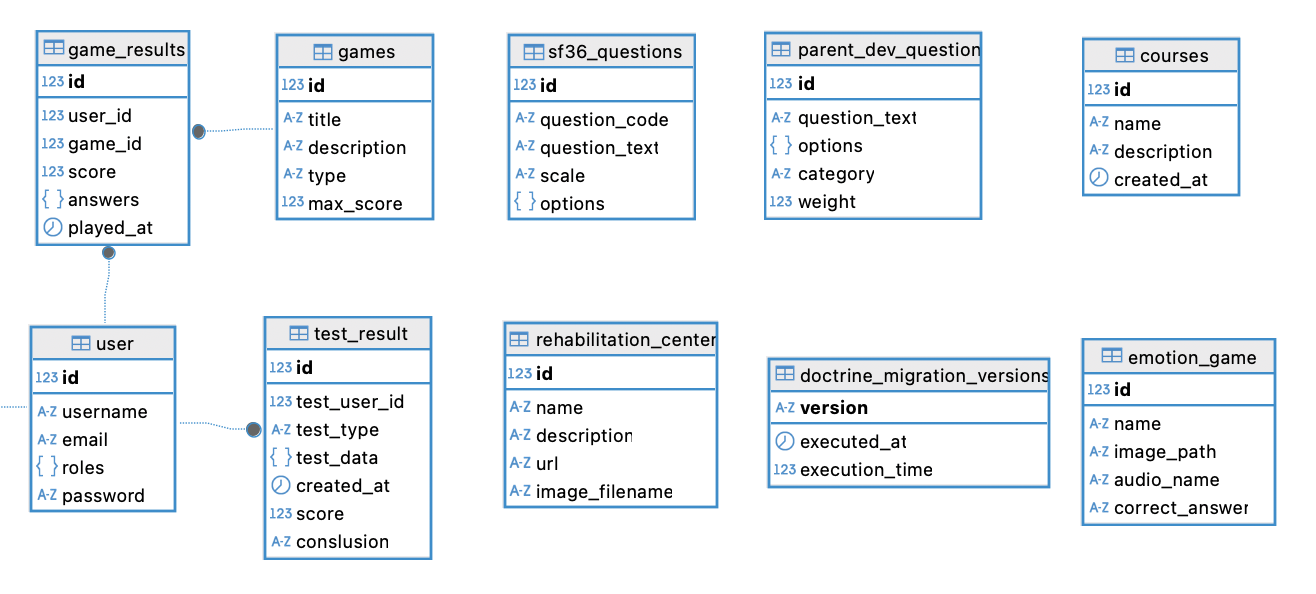


Рисунок 2. ER-диаграмма базы данных

Взаимодействие между компонентами архитектуры происходит по четко определенным сценариям.

Взаимодействие пользователя с веб-приложением (Frontend ↔ Backend):

1. Пользователь открывает веб-приложение в браузере. Клиентская часть (Vue.js) загружается и отображает интерфейс.
2. При действиях пользователя (например, переход по ссылке, отправка формы, запуск игры) Frontend формирует HTTP-запрос (GET, POST, PUT, DELETE) к соответствующему API-эндпоинту на серверной части (PHP/Symfony).
3. Backend (Symfony) принимает запрос, выполняет аутентификацию и авторизацию пользователя (если требуется).
4. Контроллер Symfony обрабатывает запрос, взаимодействуя с сервисами и моделями (Doctrine), которые реализуют бизнес-логику.
5. При необходимости доступа к данным, модели Doctrine формируют SQL-запросы к базе данных PostgreSQL.
6. База данных выполняет запросы и возвращает данные.
7. Backend формирует HTTP-ответ (часто в формате JSON) и отправляет его обратно на Frontend.
8. Frontend (Vue.js) получает ответ, обновляет пользовательский интерфейс и отображает данные пользователю.

Взаимодействие пользователя с Telegram-ботом:

1. Пользователь отправляет сообщение Telegram-боту (Python).
2. Логика Telegram-бота обрабатывает сообщение.
3. Если для ответа требуется информация из основной системы или выполнение сложной логики, Python-бот формирует HTTP-запрос к API-прослойке на Flask.
4. Flask API принимает запрос и обрабатывает его.
5. Flask API передает ответ Python-боту.
6. Python-бот формирует ответное сообщение и отправляет его пользователю в Telegram.

## 2.4. Проектирование функциональных модулей

### 2.4.1. Проектирование главной страницы

Главная страница – это «лицо» приложения. Ее цель – предоставить пользователю общее представление о возможностях ресурса, его миссии и целевой аудитории, а также обеспечить удобную навигацию к основным разделам. Структура и контент:

– шапка (Header): логотип, название приложения, основное навигационное меню (ссылки на ключевые модули: «Игры», «Курсы», «Центры», «Тесты», «Профиль», «О нас»);

– приветственный блок: краткое, емкое сообщение о том, чем полезно приложение, для кого оно предназначено;

– блок «Наши возможности»: краткое описание преимуществ для родителей, детей, специалистов;

– блок с ссылками на Telegram-бот и Telegram-канал;

– блок с мини-играми;

– блок «О нас»;

– всплывающее окно с ссылкой на Telegram-бот;

– подвал (Footer).

### 2.4.2. Проектирование модуля развивающих мини-игр

Назначение: предоставить детям с ОВЗ доступ к интерактивным играм, направленным на развитие различных когнитивных, речевых и моторных навыков в увлекательной форме. Структура и контент:

– карточки игр: каждая игра представлена карточкой, содержащей название, иконку/миниатюру, краткое описание;

– страница конкретной игры содержит непосредственно интерфейс игры, краткие правила или инструкция перед началом при необходимости.

### 2.4.3. Проектирование модуля курсов

Назначение: предоставить родителям и специалистам доступ к информации об обучающих курсах, семинарах, вебинарах, направленных на повышение их компетенций в вопросах воспитания, обучения и реабилитации лиц с ОВЗ. Структура и контент:

– карточки курсов: название, краткое описание, ссылка на прохождение;

– ссылка на все курсы.

### 2.4.4. Проектирование модуля реабилитационных центров

Назначение: предоставить пользователям удобный доступ к базе данных реабилитационных центров и специалистов, работающих с лицами с ОВЗ. Структура и контент:

– карточки центров: название, картинка центра, краткое описание, ссылка на центр.

### 2.4.5. Проектирование модуля тестирования

Назначение: предоставить пользователям (детям, родителям, специалистам) возможность пройти различные психолого-педагогические тесты и опросники для самодиагностики, оценки уровня развития, выявления проблемных зон. Структура и контент:

– разделение тестов по категориям: «Тесты для детей», «Тесты для родителей», «Тесты для специалистов»;

– страница прохождения теста;

– страница результатов теста;

– типы тестов: для детей: «Тест на внимательность»; для родителей: «Оценка качества жизни», «Оценка развития»; для специалистов: «Оценка качества жизни».

### 2.4.6. Проектирование личных кабинетов

Назначение: предоставить зарегистрированным пользователям персонализированное пространство для управления своими данными, просмотра результатов, отслеживания прогресса. Структура и контент:

– личный кабинет родителя: просмотр результатов прохождения тестов ребенка и своих результатов;

– личный кабинет специалистов: просмотр результатов прохождения своих тестов, просмотр результатов прохождения тестов детей и родителей, а также их контактных данных.

### 2.4.7. Проектирование интеграции с Telegram-ботом

Назначение: обеспечить дополнительный, быстрый и удобный канал получения информации и первичной поддержки для родителей через популярный мессенджер. Бот не заменяет приложение, а дополняет его и направляет пользователя к нужным ресурсам. Структура и контент:

– точка входа из веб-приложения: на сайте должна быть четкая ссылка или для перехода в Telegram-бот;

– направление к специалистам: в сложных случаях или при прямом запросе бот может предлагать информацию о том, как связаться со специалистами (например, через форму на сайте, или предоставляя контакты, если это предусмотрено).

# 3. Разработка веб-приложения психолого-педагогического сопровождения лиц с ОВЗ

## 3.1. Описание процесса и этапов разработки

Разработка веб-приложения для психолого-педагогического сопровождения лиц с ОВЗ осуществлялась с применением итерационного подхода, что позволило гибко реагировать на возникающие в процессе работы уточнения и последовательно наращивать функционал. Данный подход предполагает разделение всего процесса на ряд коротких циклов (итераций), каждый из которых включает этапы планирования, проектирования, реализации и тестирования определенной части функционала.

Процесс разработки можно условно разделить на следующие ключевые этапы:

1. Предпроектный анализ и планирование (Итерация 0).
2. Проектирование архитектуры и интерфейсов.
3. Последовательная разработка функциональных модулей (Основные итерации): Разработка велась по модульному принципу. Каждая итерация была сфокусирована на реализации одного или нескольких взаимосвязанных модулей:
4. Итерация 1: Разработка базовой структуры проекта, системы аутентификации и регистрации пользователей, создание личных кабинетов (родителя и специалиста) с базовым функционалом.
5. Итерация 2: Реализация модуля развивающих мини-игр (разработка нескольких пилотных игр и каталога).
6. Итерация 3: Разработка модуля тестирования (создание интерфейсов для прохождения тестов и отображения результатов).
7. Итерация 4: Реализация модулей «Список курсов» и «Список реабилитационных центров» (включая функционал фильтрации и отображения).
8. Итерация 5: Разработка API для связи с Telegram-ботом на стороне основного бэкенда и реализация самого Telegram-бота на Python с использованием Flask API для взаимодействия.
9. Интеграция и тестирование.
10. Рефакторинг и оптимизация.
11. Подготовка к представлению.

Итерационный подход позволил своевременно вносить коррективы в проект, управлять сложностью разработки и постепенно двигаться к финальному продукту, регулярно оценивая промежуточные результаты.

## 3.2. Реализация серверной части (Backend)

Серверная часть (бэкенд) веб-приложения является его ядром, ответственным за обработку всех запросов от браузера пользователя (через отображение Twig-шаблонов). Бэкенд реализует всю бизнес-логику, управляет данными и взаимодействует с базой данных. Разработка велась на языке PHP с использованием фреймворка Symfony и ORM Doctrine для работы с СУБД PostgreSQL.

Маршруты (URL-адреса, доступные пользователям) определялись в Symfony с помощью аннотаций непосредственно в классах контроллеров. Каждый маршрут связывался с конкретным методом контроллера. Контроллеры отвечают за прием HTTP-запросов от браузера, извлечение данных из них (параметры GET-запроса, данные POST-форм), вызов соответствующих сервисов для обработки бизнес-логики и передачу данных в Twig-шаблоны для последующего рендеринга HTML-страницы. Пример контроллера с аннотацией для маршрута для страницы отображения курсов:

<?php

namespace App\Controller;

use App\Repository\CourseRepository;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

class CourseController extends AbstractController

{

#[Route('/courses', name: 'app\_courses')]

public function index(CourseRepository $courseRepository): Response

{

$courses = $courseRepository->findAllOrderedByDate();

return $this->render('course/index.html.twig', [

'courses' => $courses,

]);

}

}

Взаимодействие с пользователем и реализация бизнес-логики приложения происходят преимущественно в контроллерах Symfony. Контроллеры отвечают за прием HTTP-запросов от браузера, извлечение данных из запроса, реализацию основной бизнес-логики, прямое взаимодействие с Doctrine EntityManager и репозиториями для извлечения, сохранения или изменения данных в базе. Например, метод из контроллера TestController.php, который проверяет тип теста и при несовпадении выдает исключение:

#[Route('/test/sf36/results/{id}', name: 'sf36\_results')]

public function sf36Results(TestResult $result): Response

{

if ($result->getTestType() !== 'sf36') {

throw $this->createNotFoundException();

}

return $this->render('test/sf36\_results.html.twig', [

'result' => $result,

'testData' => $result->getTestData()

]);

}

Также контроллеры отвечают за вызов соответствующих сервисов для обработки бизнес-логики. Код сервиса Service/Sf36Calculator.php, который подсчитывает результаты теста «SF-36» приведен в Приложении Б.

Работа с базой данных PostgreSQL осуществлялась с использованием Doctrine ORM непосредственно из методов контроллеров. Были созданы PHP-классы (сущности) для каждой таблицы базы данных с описанием структуры и связей через аннотации. В серверной части реализованы следующие ключевые сущности: Пользователи (User), Курсы (Course), Мини-игры (EmotionGame), Реабилитационные центры (RehabilitationCenter), Тесты (Sf36Question, ParentDevQuestion, TestResult). Пример сущности Game:

<?php

namespace App\Entity;

use App\Repository\GameRepository;

use Doctrine\DBAL\Types\Types;

use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: GameRepository::class)]

#[ORM\Table(name: 'games')]

class Game

{

#[ORM\Id]

#[ORM\GeneratedValue]

#[ORM\Column]

private ?int $id = null;

#[ORM\Column(length: 100)]

private ?string $title = null;

#[ORM\Column(type: Types::TEXT)]

private ?string $description = null;

#[ORM\Column(length: 20)]

private ?string $type = null;

#[ORM\Column]

private ?int $maxScore = 10;

EntityManager внедрялся в контроллеры (например, через конструктор или сервис-контейнер) и использовался для операций сохранения (persist), обновления и удаления (remove) сущностей, а также для применения изменений к базе данных (flush). Фрагмент кода с примером использования репозитория и EntityManager в контроллере для выборки и изменения данных приведен ниже:

#[Route('/save-result', name: 'save\_emotion\_result', methods: ['POST'])]

public function saveResult(Request $request, EntityManagerInterface $em): JsonResponse

{

$data = json\_decode($request->getContent(), true);

$result = new GameResult();

$result->setGame($em->getReference(Game::class, 1)); // ID игры "Угадай эмоцию"

$result->setUser($this->getUser());

$result->setScore($data['score']);

$result->setAnswers($data['answers']);

$em->persist($result);

$em->flush();

return $this->json(['status' => 'success']);

}

При добавлении или изменении сущностей в коде используется команда генерации миграций, которая создает SQL-скрипты для обновления схемы базы данных:

php bin/console make:migration

После этого миграции применяются командой:

php bin/console doctrine:migrations:migrate

Для обеспечения стабильного тестового окружения и удобства разработки в проекте применяются Doctrine Fixtures — специальные классы, которые позволяют автоматически загружать наборы тестовых данных в базу. Это особенно важно для модульного и функционального тестирования, где требуется иметь однородные и предсказуемые данные при каждом запуске тестов. В процессе разработки используются команды Symfony для загрузки фикстур:

php bin/console doctrine:fixtures:load --env=test

Эта команда очищает тестовую базу данных и загружает в неё необходимые данные, что позволяет проводить тесты в изолированном и контролируемом окружении.

## 3.3. Реализация клиентской части (Frontend)

Клиентская часть (фронтенд) веб-приложения отвечает за визуальное представление информации пользователю и обеспечение интерактивного взаимодействия с системой. В данном проекте фронтенд реализован с использованием классического подхода, где HTML-структура страниц генерируется на стороне сервера с помощью шаблонизатора Twig (в рамках фреймворка Symfony), а затем стилизуется с помощью CSS (включая фреймворк Bootstrap) и дополняется интерактивностью с помощью JavaScript.

Twig-шаблоны использовались для создания базового макета (base.html.twig), который включал общие элементы, такие как шапка (header), подвал (footer) и основную область для контента. Другие страницы наследовались от этого базового макета, переопределяя или дополняя его блоки. Код базового макета приведен ниже:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>РостОк</title>

<link rel="stylesheet" href="{{ asset('assets/styles/app.css') }}">

<link rel="stylesheet" href="{{ asset('assets/styles/profile.css') }}">

{% block stylesheets %}

{{ encore\_entry\_link\_tags('app') }}

{% endblock %}

<!-- Bootstrap (для ускорения) -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

{% include 'parts/header.html.twig' %}

<main class="container mt-4">

{% block content %}{% endblock %}

</main>

{% include 'parts/footer.html.twig' %}

<!-- Скрипты -->

{% block javascripts %}

<!-- Bootstrap JS -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

{% endblock %}

</body>

</html>

Активно применялись классы Bootstrap для создания сеток, стилизации форм (.form-control, .form-label), кнопок (.btn, .btn-primary), навигационных панелей (.navbar), модальных окон и других компонентов.

Для придания приложению индивидуального стиля и переопределения некоторых стандартных стилей Bootstrap были написаны кастомные CSS-правила. Они размещались в отдельных файлах и подключались в Twig-шаблонах или непосредственно в самих шаблонах.

JavaScript использовался для добавления функциональности, которая не требует полной перезагрузки страницы или выполняется непосредственно в браузере. Перед отправкой данных на сервер, JavaScript использовался для предварительной проверки заполнения обязательных полей, корректности форматов (например, email), что снижало нагрузку на сервер и улучшало пользовательский опыт, предоставляя мгновенную обратную связь.

Для развивающих мини-игр основная логика взаимодействия (проверка правильности ответа в простых играх, анимации) реализовывалась на JavaScript.

Фрагмент JS-кода логики мини-игры «Найди пару»:

function initGame(level) {

// Очищаем контейнер

cardsContainer.innerHTML = '';

flippedCards = [];

matchedPairs = 0;

canFlip = true;

// Устанавливаем текущий уровень

currentLevelEl.textContent = level + 1;

// Создаем пары карточек

const images = cardImages[level];

totalPairs = images.length;

const cards = [];

// Создаем пары

images.forEach(image => {

cards.push(image);

cards.push(image);

});

// Перемешиваем карточки

shuffleArray(cards);

## 3.4. Реализация функциональных модулей

### 3.4.1. Разработка главной страницы

Главная страница является визитной карточкой приложения. Ее реализация включала создание соответствующего контроллера и Twig-шаблона.

Контроллер (MainController.php) отвечал за обработку запроса к корневому URL и передавал данные в Twig-шаблон для рендеринга:

<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

class MainController extends AbstractController

{

#[Route('/', name: 'app\_home')]

public function index(): Response

{

return $this->render('main/index.html.twig');

}

}

Twig-шаблон (templates/main/index.html.twig) наследовался от базового шаблона (base.html.twig), определял структуру главной страницы: шапку, приветственный блок, блоки с ссылками на Telegram-канал и бот, блок с мини-играми, всплывающее окно с ссылкой на Telegram-бот, информационный блок, подвал (рис. 3).

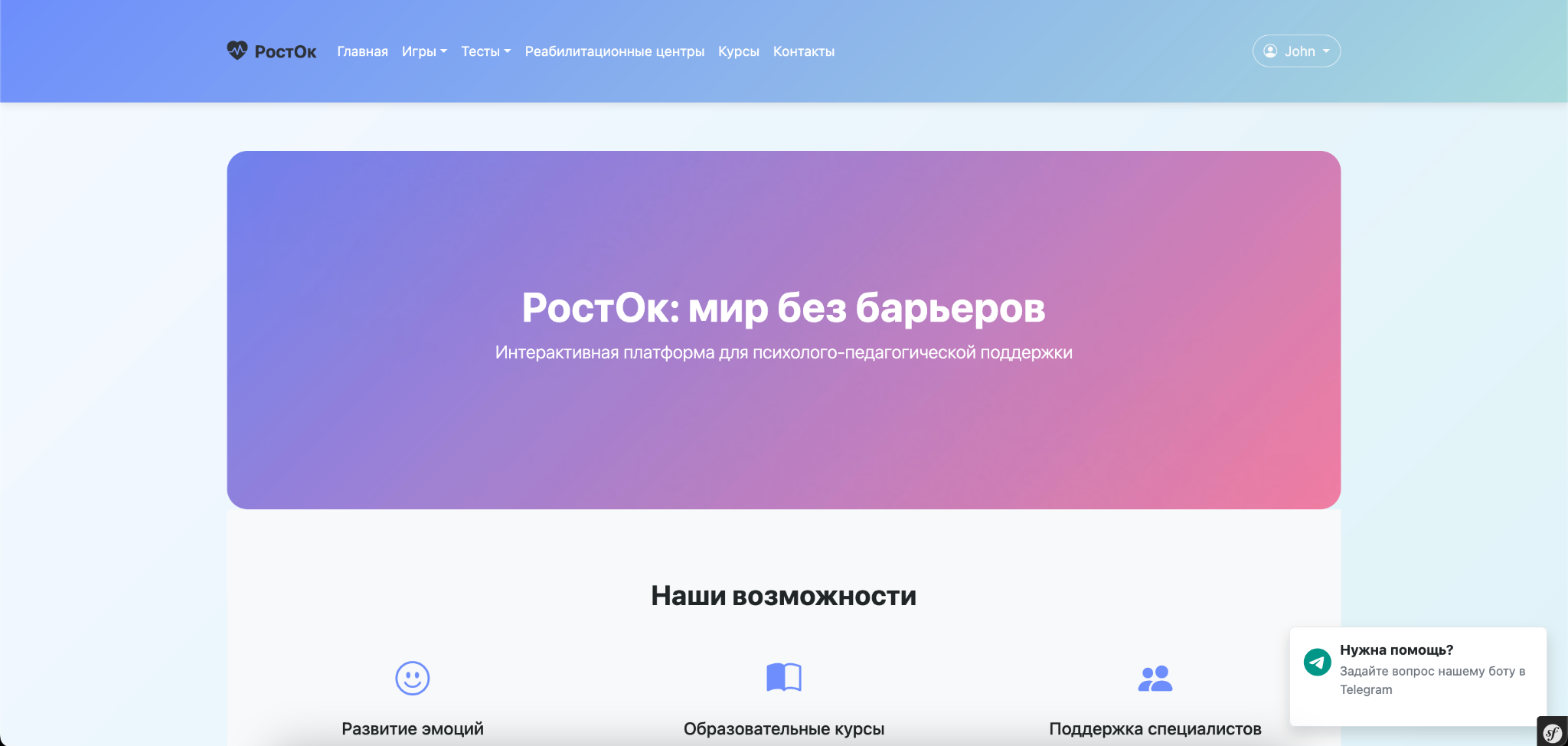


Рисунок 3. Интерфейс главной страницы

### 3.4.2. Разработка модуля развивающих мини-игр

Этот модуль требовал как серверной части для отображения каталога и сохранения результатов, так и значительной клиентской части на JavaScript для реализации логики самих игр.

Для каждой из мини-игр существует свой контроллер.

Контроллер для игры «Угадай эмоцию» (рис. 4) (EmotionGameController.php) получает список эмоций из базы данных и передает их в шаблон, который содержит контейнер для игры и скрипты.

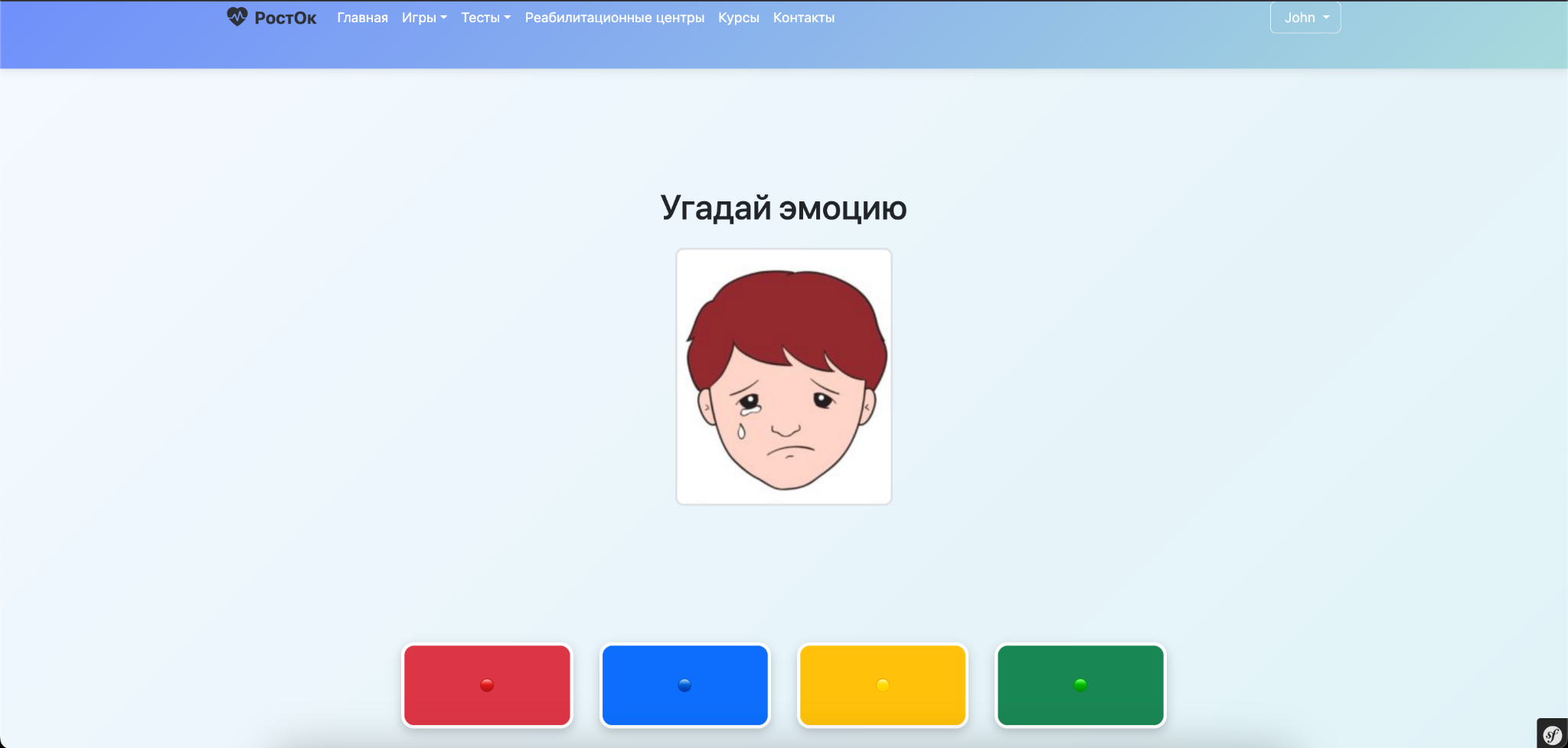


Рисунок 4. Интерфейс мини-игры «Угадай эмоцию»

Twig-шаблон (templates/emotion\_game/index.html.twig) отображает игровое поле. Задача ребенка – выбрать подходящий вариант на основе услышанных аудио-подсказок. Все аудио файлы, такие как описание эмоций и обратная связь при выборе, были внедрены в шаблон через тег <audio>, что позволяет обеспечить дополнительную поддержку детям с особенностями восприятия (рис. 5).



Рисунок 5. Аудио-файлы

Основная логика игры прописана в JavaScript. При загрузке страницы DOM полностью строится, после чего происходит инициализация событий и состояния игры. Кнопки выбора цвета содержат обработчики кликов, которые вызывают функцию проверки ответа пользователя. При начале игры случайным образом выбирается эмоция, и система озвучивает все варианты ответов, используя разные аудиозаписи для каждого цвета и соответствующей эмоции. Пользователь должен выбрать тот цвет, который, по его мнению, лучше всего соответствует показанной эмоции.

После выбора цвета система проверяет правильность ответа и подсвечивает кнопки: зелёным – правильный ответ, красным – неправильный. Также проигрывается соответствующее звуковое сопровождение, а затем озвучивается правильный ответ.

Контроллер для игры «Найди пару» (рис. 6) (FlipGameController.php) предоставляет два маршрута: первый – отображает саму игру по адресу /flip-game, проверяя, авторизован ли пользователь через роль ROLE\_USER; второй маршрут /flip-game/save-result принимает результаты игры в формате JSON. JS-код генерировал сетку карточек, обрабатывал клики, сравнивал выбранные карточки, реализовывал логику "переворачивания" и исчезновения пар, а также подключал звуковое сопровождение успеха/ неудачи.

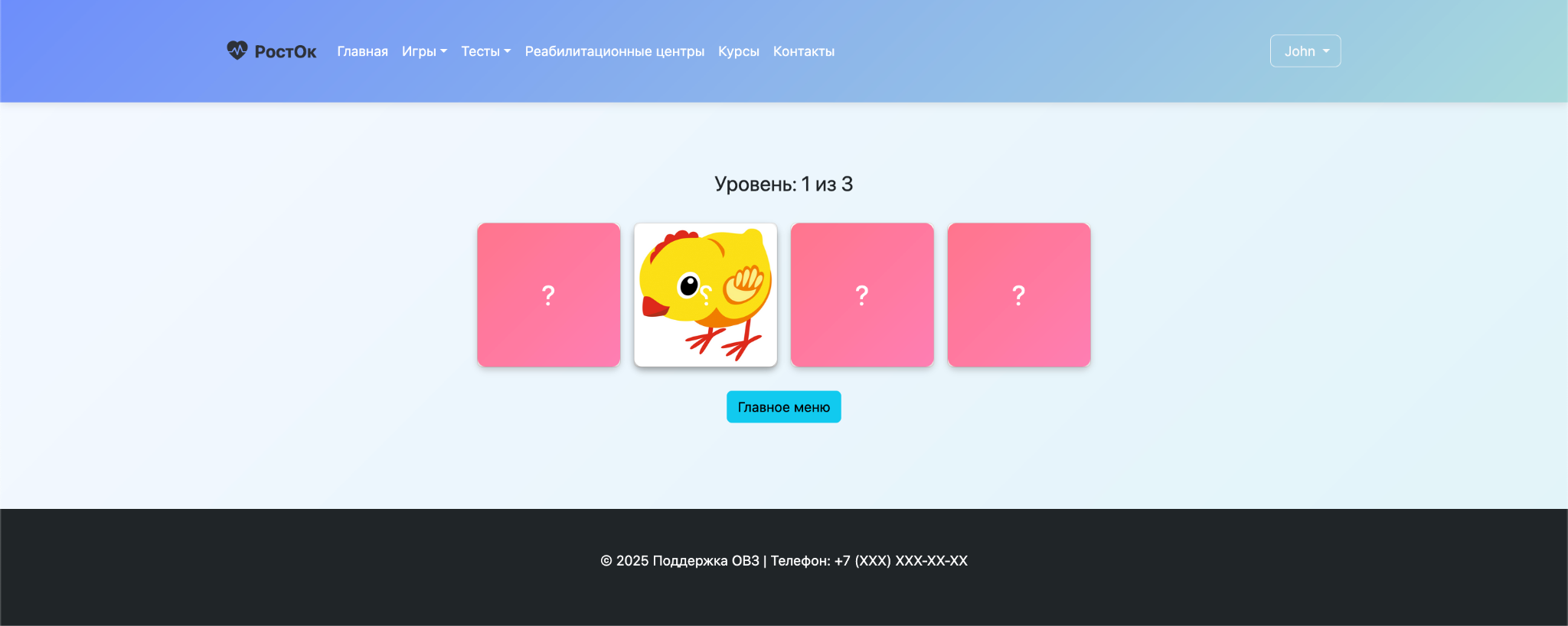


Рисунок 6. Интерфес мини-игры «Найди пару»

### 3.4.3. Разработка модуля списка курсов

За обработку запросов к странице курсов отвечает контроллер CourseController. В нем определен маршрут /courses, который отображает список всех курсов, отсортированных по дате добавления. Для получения данных используется репозиторий CourseRepository с методом findAllOrderedByDate().

Визуальная часть (рис. 7) реализована в шаблоне course/index.html.twig. Шаблон расширяет базовый макет и определяет стили, структуру и интерактивное поведение раздела. Вывод курсов: для каждого курса формируется карточка, содержащая: название курса и дату добавления, краткое или полное описание с возможностью раскрытия, кнопку «Подробнее», ведущую на внешний сайт с дополнительной информацией. Кнопка для перехода к дополнительным курсам размещена в конце списка и ведет на внешний образовательный ресурс.

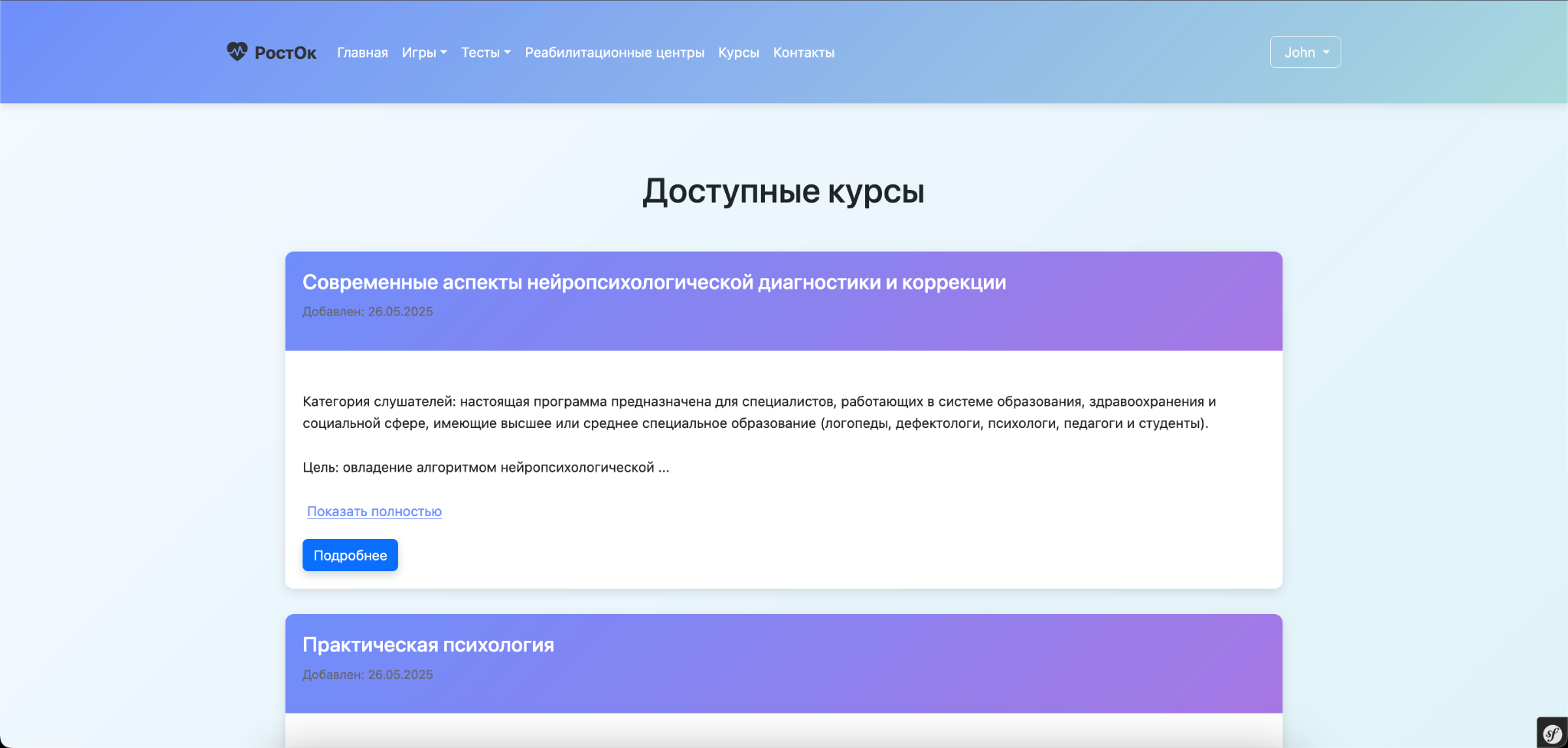


Рисунок 7. Интерфейс страницы с курсами

### 3.4.4. Разработка модуля списка реабилитационных центров

За обработку запросов, связанных с реабилитационными центрами, отвечает контроллер RehabilitationCenterController. Он реализует маршрут /centers, который отображает список всех доступных центров. В контроллере используется внедрение зависимости – репозитория RehabilitationCenterRepository, что позволяет получать данные о центрах из базы данных с помощью Doctrine ORM.

Визуальное представление (рис. 8) реализовано с помощью Twig-шаблона rehabilitation\_center/index.html.twig. Шаблон расширяет базовый layout base.html.twig и определяет блоки для заголовка, стилей и основного контента. В блоке stylesheets добавлены кастомные стили для карточек центров. В блоке content реализована адаптивная сетка карточек с помощью Bootstrap.

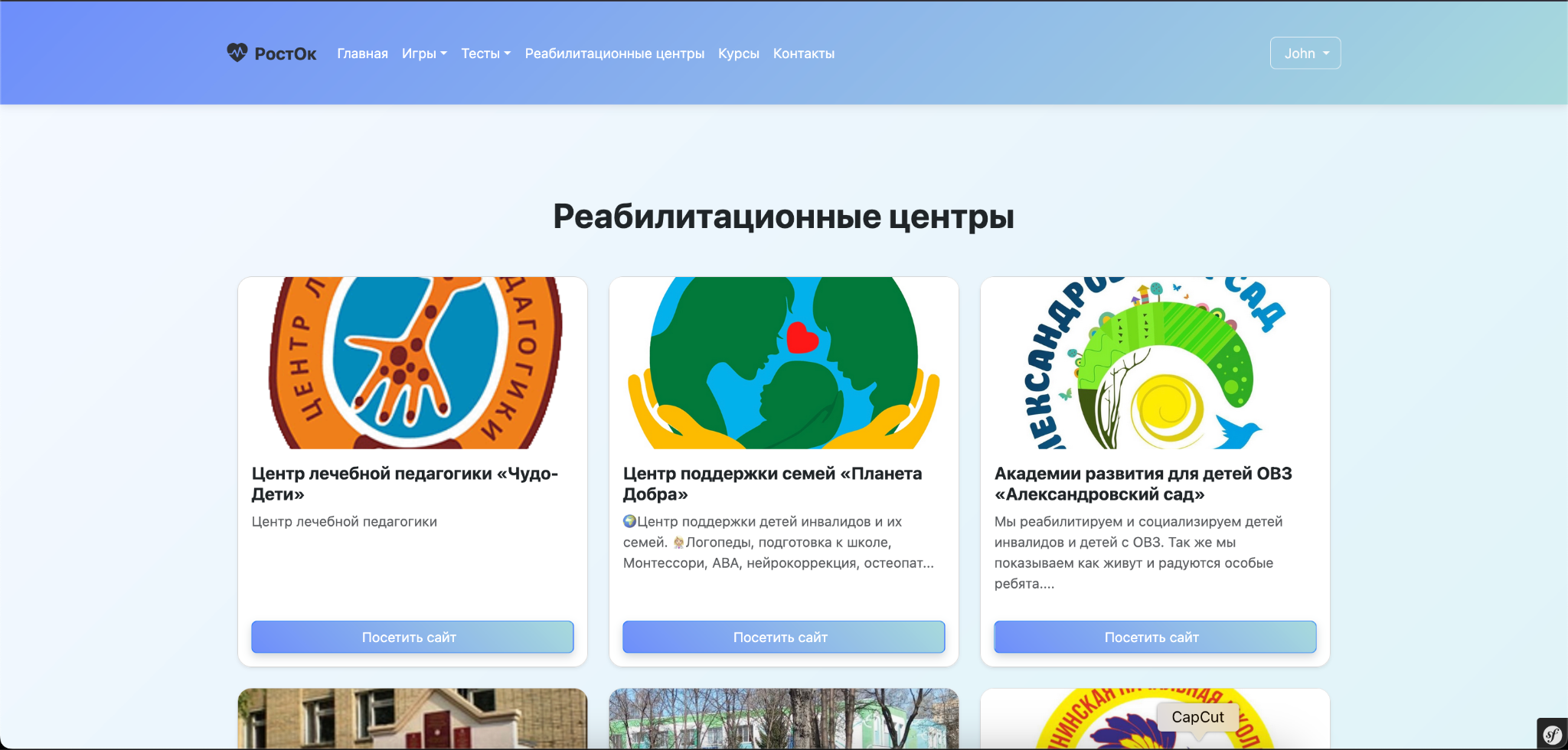


Рисунок 8. Интерфейс страницы с реабилитационными центрами

### 3.4.5. Разработка модуля тестирования

Модуль тестирования реализован в двух контроллере TestController и SchulteController.

TestController обеспечивает прохождение и обработку различных тестов, таких как SF-36 (рис. 9) и тест оценки развития ребенка (parent-dev) (рис. 10). Для каждого теста предусмотрены маршруты отображения вопросов, обработки ответов и вывода результатов. Вопросы загружаются из базы через соответствующие репозитории, а результаты рассчитываются с помощью сервисов и сохраняются в сущности TestResult вместе с персональными данными пользователя и заключениями. На основе набранных баллов формируются рекомендации и выводятся подробные отчёты.

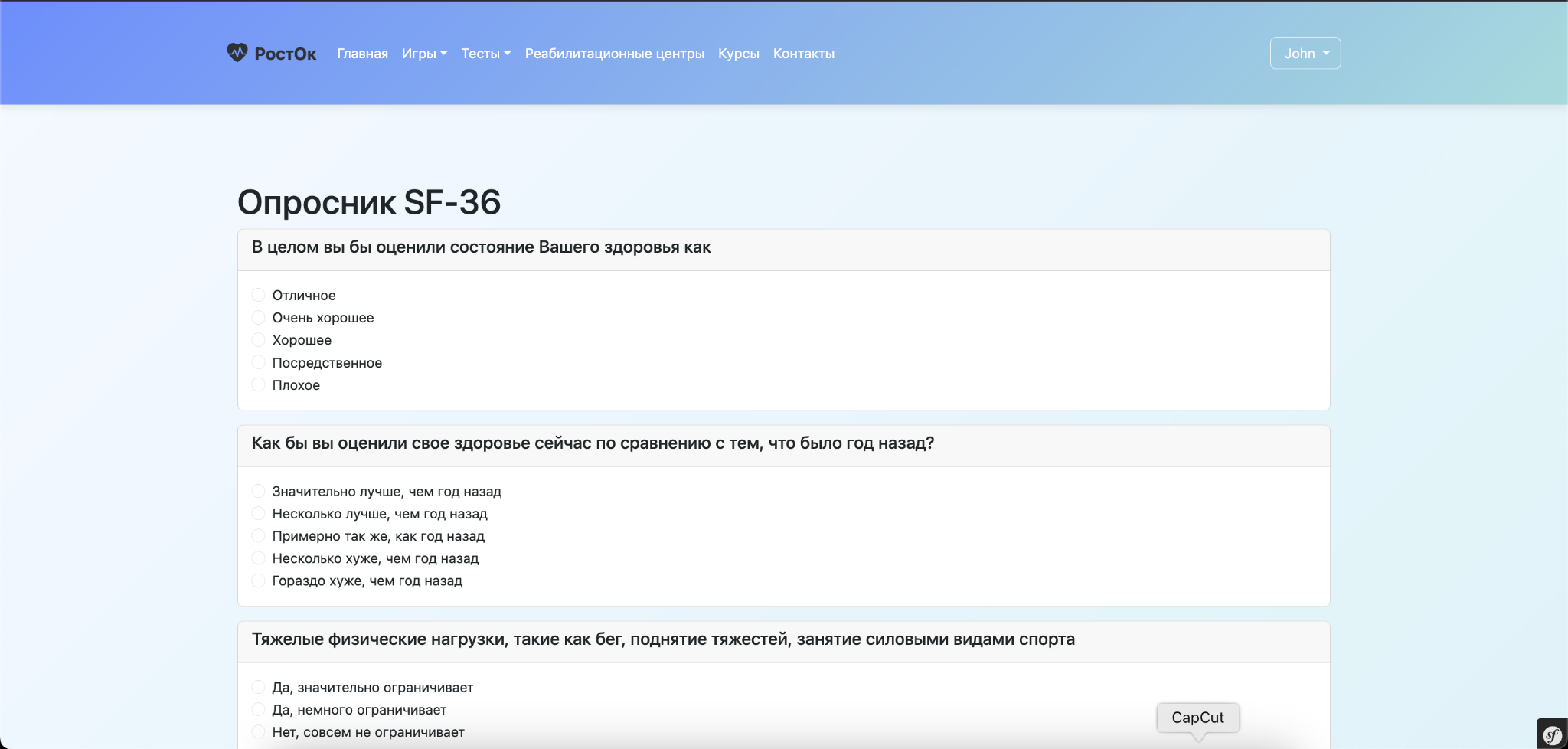


Рисунок 9. Интерфейс теста «Оценка качества жизни»

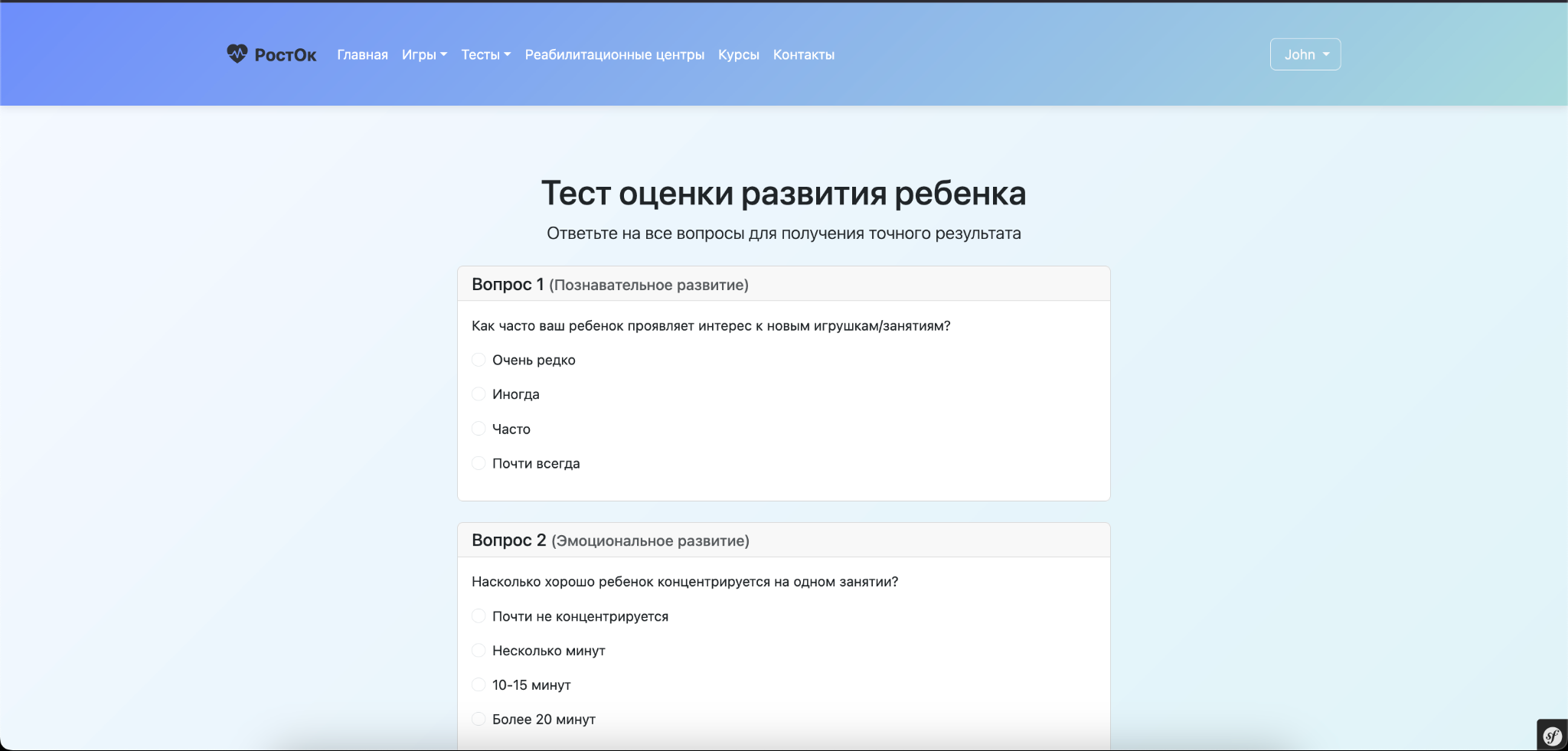


Рисунок 10. Интерфейс теста «Оценка развития»

Важной частью разработанного приложения является интерактивный модуль тестирования внимательности с использованием таблиц Шульте (рис. 11). Контроллер SchulteController беспечивает отображение интерфейса теста и обработку результатов. Контроллер содержит два основных маршрута: первый ('/test/schulte') отвечает за отображение страницы с тестом, а второй ('/test/schulte/save-results') обрабатывает и сохраняет результаты прохождения теста.

Интерфейс теста представляет собой адаптивную сетку с числами, которые пользователь должен находить в порядке возрастания. Пользователю предоставляется возможность выбрать размер таблицы (3×3, 4×4, 5×5 или 6×6), что позволяет регулировать сложность теста в зависимости от индивидуальных возможностей. Для расчета итогового балла используется метод calculateScore, который учитывает размер таблицы, затраченное время и количество ошибок. Формула расчета построена таким образом, что больший размер таблицы дает больше базовых баллов, а время и ошибки снижают итоговый результат. Контроллер генерирует персонализированное заключение с помощью метода generateConclusion, который анализирует полученный балл и формирует текстовые рекомендации в зависимости от уровня результата. Для высоких баллов (80 и выше) предлагается поддерживать текущий уровень и пробовать таблицы большего размера, для средних результатов (50-79) рекомендуются регулярные тренировки, а для низких баллов даются советы по постепенному улучшению навыков внимания. Вся информация о тесте, включая детали расчета и формулу, сохраняется в поле testData, что позволяет в дальнейшем анализировать результаты и отслеживать прогресс пользователя.

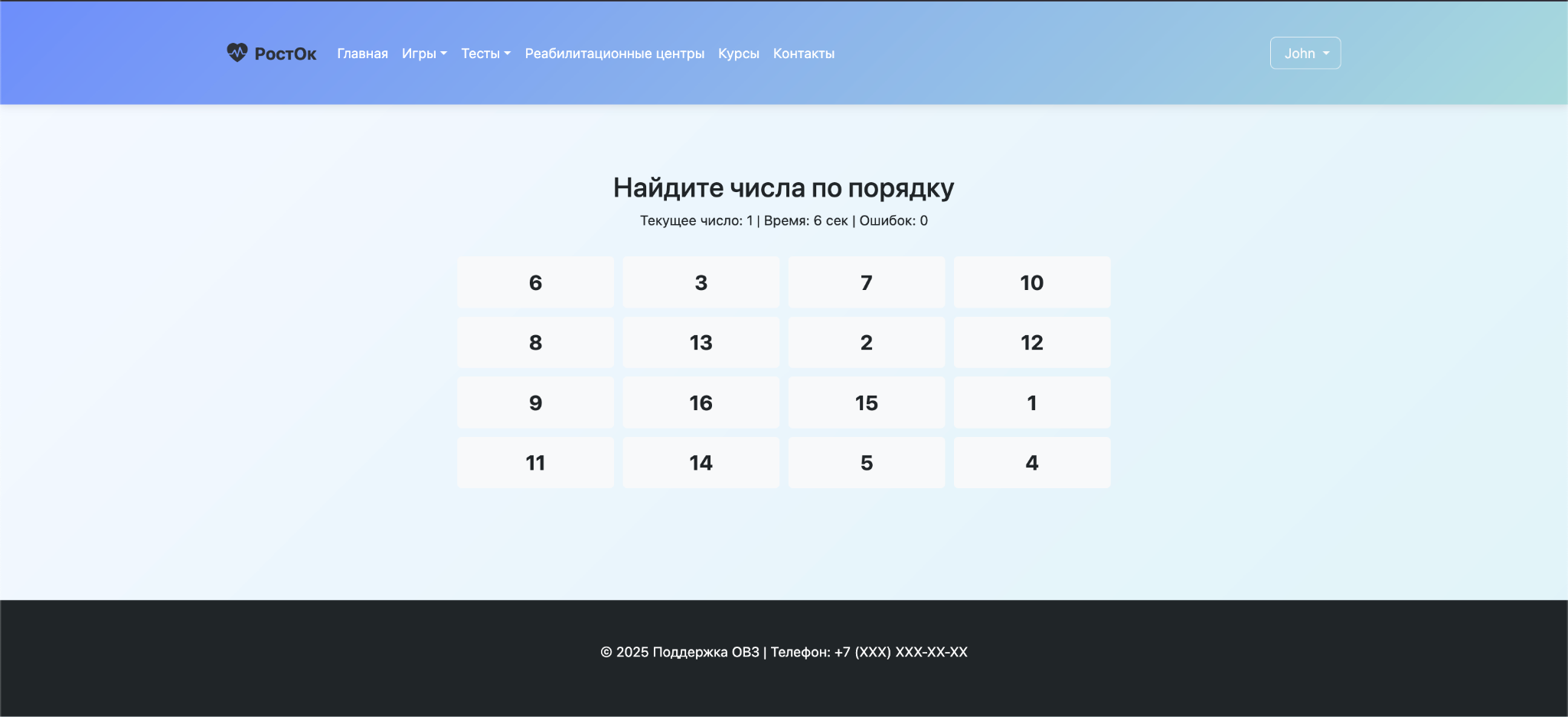


Рисунок 11. Интерфейс теста «Тест на внимательность»

### 3.4.6. Разработка личных кабинетов родителя и специалиста

За обработку логики профиля отвечает контроллер ProfileController. В его основе лежит механизм авторизации: при обращении к маршруту /profile система проверяет, вошел ли пользователь в систему, и если нет — возвращает ошибку доступа. После успешной аутентификации определяется роль пользователя. Если у пользователя роль специалиста, вызывается метод renderSpecialistProfile, если родитель — renderParentProfile.

Для родителя в профиле отображаются две основные вкладки: «Результаты ребенка» и «Мои тесты» (рис. 12). В первой вкладке выводятся результаты тестов, которые относятся к ребенку (например, тесты внимательности или развития), а во второй — личные тесты родителя, такие как SF-36 или тесты оценки эмоционального состояния.

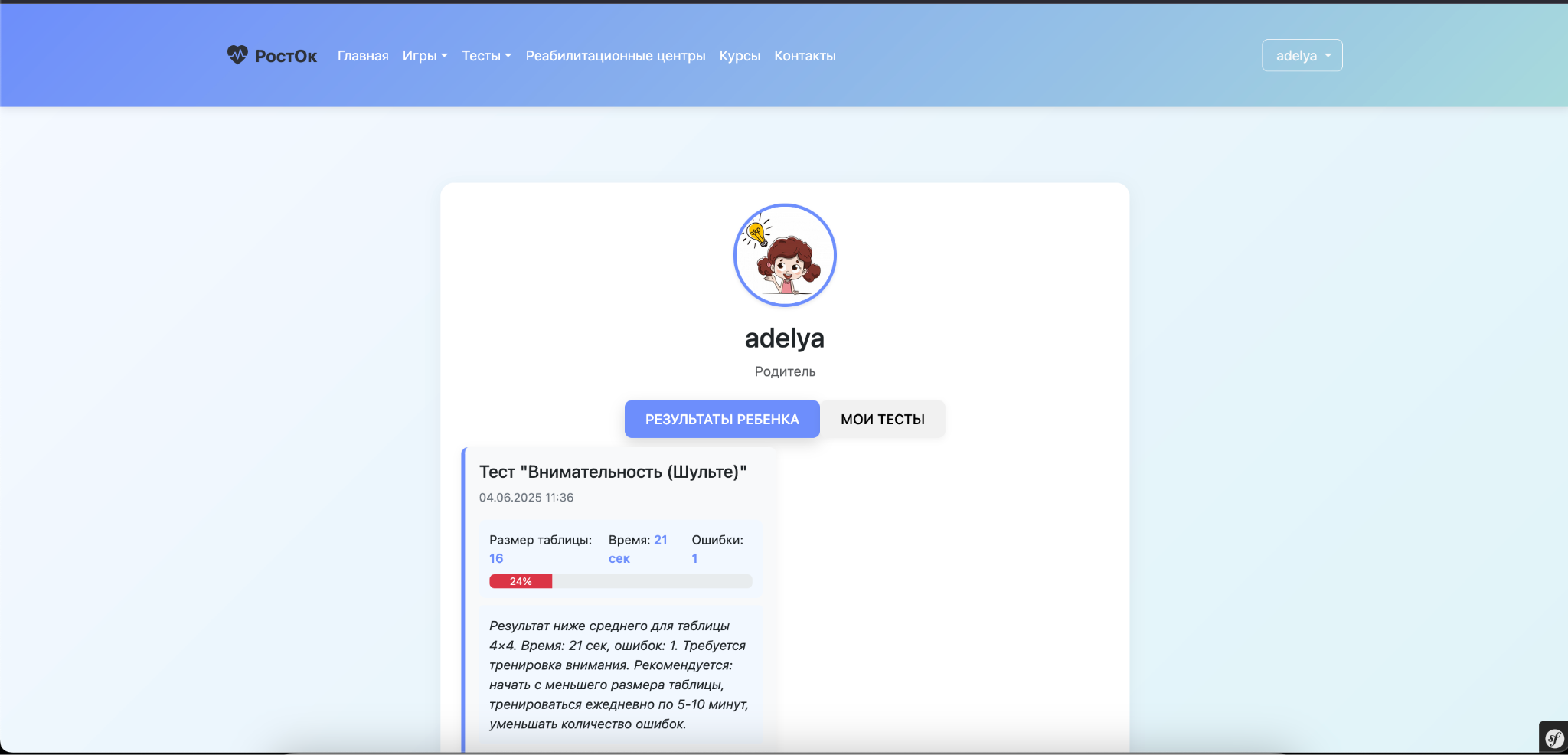


Рисунок 12. Интерфейс личного кабинета родителя

Профиль специалиста содержит расширенную информацию. Помимо собственных результатов тестирования (например, SF-36), специалисту доступен просмотр тестов других пользователей, что реализовано через отдельный метод репозитория (рис. 13).

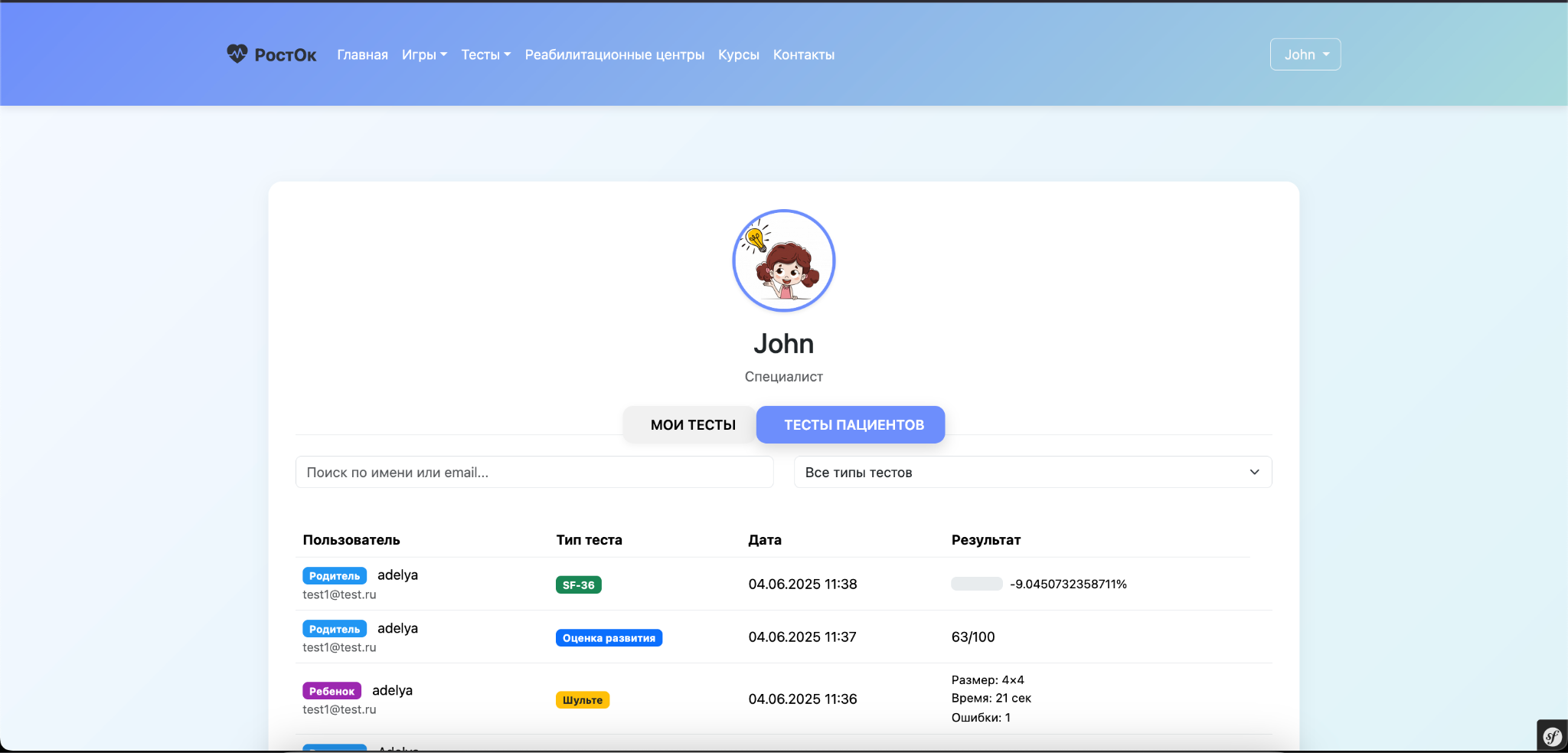


Рисунок 13. Интерфейс личного кабинета специалиста

### 3.4.7. Реализация интеграции с Telegram-ботом и Telegram-каналом

В рамках проекта реализован интеллектуальный Telegram-бот, который служит помощником для родителей детей с особыми образовательными потребностями. Основой для ответов на вопросы пользователей служит модель глубокого обучения, построенная на базе библиотеки DeepPavlov и модели squad\_ru\_bert, способной работать с русскоязычными текстами. Серверная часть приложения реализована с использованием Flask и предоставляет API, принимающее вопросы в формате JSON и возвращающее ответы с уровнем уверенности модели.

При запуске сервер загружает контекст из файла context.txt, который содержит базу знаний, необходимую для формирования ответов. По запросу пользователя модель обрабатывает вопрос в контексте заданного текста и возвращает наиболее релевантный ответ вместе с оценкой уверенности. В случае возникновения ошибок сервер возвращает информативное сообщение с соответствующим HTTP-кодом.

Telegram-бот построен с использованием библиотеки python-telegram-bot версии 20+, что обеспечивает асинхронную обработку сообщений и удобное управление хендлерами. При команде /start бот отправляет приветственное сообщение с описанием своих возможностей и примерами вопросов, которые пользователь может задать. Основной функционал реализован в обработчике текстовых сообщений: бот принимает вопрос пользователя, отправляет его на Flask-сервер через POST-запрос, получает ответ и в зависимости от уровня уверенности формирует ответное сообщение. Если уверенность модели очень высокая, бот выводит только ответ, в противном случае добавляет рекомендацию обратиться к специалистам, ссылаясь на Telegram-канал, информация о котором доступна на главной странице приложения.

# 

# 4. Тестирование программного продукта (веб-приложения)

## 4.1. Планирование и организация тестирования

Тестирование проводилось преимущественно ручным методом, что позволило гибко проверять разнообразные сценарии использования и оценить субъективные аспекты пользовательского опыта. Дополнительно применялись элементы исследовательского тестирования для выявления нетривиальных проблем. Объектом тестирования выступало веб-приложение в целом, включая все его страницы, функциональные модули и базовую проверку интеграции с Telegram-ресурсами. Тестирование проводилось на операционной системе macOS, в веб-браузерах Google Chrome, Mozilla Firefox. Адаптивность интерфейса проверялась с использованием инструментов разработчика в браузерах для эмуляции различных разрешений экрана (десктоп, планшет, смартфон). Критериями успешности считались отсутствие критических дефектов, соответствие функциональным требованиям и удовлетворительная оценка удобства интерфейса.

## 4.2. Функциональное тестирование

Функциональное тестирование было направлено на проверку работоспособности каждой функции приложения. Выполненные проверки описаны в таблице 1.

Таблица 1. Тестирование

| Функциональный модуль | Тест-кейсы | Результат |
| --- | --- | --- |
| Регистрация и аутентификация | Успешная регистрация нового родителя; попытка регистрации с уже существующим email; вход в систему с валидными данными; попытка входа с неверным паролем; корректный выход из системы | Все операции выполнялись корректно. Система предотвращала регистрацию дублирующихся email и корректно обрабатывала неверные учетные данные. |
| Развивающие мини-игры | Отображение каталога игр; успешный запуск игры «Найди пару»; корректная работа механики переворачивания карточек и исчезновения совпавших пар; запуск игры «Угадай эмоцию»; отображение сообщения об успешном завершении игры | Игры запускались и функционировали без сбоев, игровая логика соответствовала ожиданиям. |
| Список курсов и реабилитационных центров | Отображение полного списка курсов, реабилитационных центров; переход на страницу детального описания первого курса в списке; проверка отображения всей информации (описание, контакты) на странице центра. | Списки отображались корректно. |
| Тестирование | Выбор теста «Оценка развития», «Оценка качества жизни», «Тест на внимание»; корректное отображение всех вопросов теста и вариантов ответов; успешная отправка формы с выбранными ответами; проверка расчета итогового балла (на тестовом наборе ответов с заранее известным результатом); корректное отображение интерпретации результата; сохранение результата в личном кабинете родителя | Тесты успешно проходились, результаты рассчитывались корректно и сохранялись в профиле пользователя. |
| Личный кабинет (родителя и специалиста) | Вход в личный кабинет родителя; отображение данных профиля; просмотр результатов теста, пройденного ребенком; вход в личный кабинет специалиста; просмотр списка своих тестов и тестов пользователей | Пользователи могли просматривать информацию в соответствии со своей ролью. |
| Интеграция с Telegram-ботом и каналом | Проверка работоспособности ссылки на Telegram-бот на главной странице веб-приложения; отправка команды /start боту и получение приветственного сообщения; отправка сообщения «Нужна гимнастика при ДЦП для ног» и получение корректного ответа; проверка работоспособности ссылки на Telegram-канал | Ссылки на Telegram-ресурсы функционировали корректно, бот корректно отвечал на команды. |

## 4.4. Тестирование пользовательского интерфейса

Данный вид тестирования был направлен на оценку визуальной составляющей и удобства использования приложения.

Проверено корректное отображение всех страниц в Google Chrome, Mozilla Firefox на разрешении 1920x1080 (десктоп) и эмуляции разрешений 768x1024 (планшет) и 375x667 (смартфон). Значительных дефектов верстки (наложения элементов, некорректных переносов) не выявлено.

Структура меню и навигационные ссылки были протестированы на логичность и предсказуемость. Переход между разделами осуществлялся корректно. Кнопки «Назад» в браузере работали ожидаемо.

Проведена выборочная проверка текстового контента на страницах на предмет орфографических и грамматических ошибок. Формулировки инструкций и подписей к кнопкам были оценены как ясные.

# Заключение

В рамках курсовой работы было проведено комплексное исследование, посвященное разработке веб-приложения для психолого-педагогического сопровождения лиц с ограниченными возможностями здоровья (ОВЗ). Целью работы являлось создание функционального, удобного и доступного инструмента, способствующего повышению эффективности и качества оказываемой поддержки.

Разработанное веб-приложение представляет собой комплексное решение, объединяющее разнообразные инструменты и ресурсы для поддержки лиц с ОВЗ, их родителей и специалистов. Ключевыми преимуществами реализованного продукта являются:

1. Интегрированный подход, объединяющий информационную поддержку, развивающие игры, каталоги ресурсов и инструменты диагностики в единой системе.
2. Персонализированные личные кабинеты для различных категорий пользователей, обеспечивающие доступ к релевантной информации и результатам.
3. Адаптивный дизайн, обеспечивающий удобство использования на различных устройствах.
4. Потенциал для повышения доступности и эффективности психолого-педагогического сопровождения за счет предоставления удаленного доступа к квалифицированной помощи и ресурсам.

Вместе с тем, в ходе разработки и тестирования были выявлены некоторые ограничения и направления для дальнейшего совершенствования: необходимость расширения базы данных тестов и развивающих игр, потребность в улучшении алгоритмов обработки результатов тестов и предоставлении более развернутых рекомендаций, возможность интеграции с другими сервисами и платформами для расширения функциональности, повышение уровня доступности (accessibility) для пользователей с различными нарушениями, включая более глубокую проработку WCAG-совместимости.

Варианты развития разработанного программного продукта (приложения) включают: расширение функционала личных кабинетов (добавление инструментов для ведения дневника, реализация системы обмена сообщениями, создание календаря событий); ; развитие модуля развивающих игр (создание новых игр, внедрение системы достижений, адаптация сложности); улучшение интеграции с Telegram (обучение бота, реализация записи на консультации, активное использование канала); реализация мобильного приложения; и поддержка многоязычности. Дальнейшее развитие и совершенствование приложения, основанное на анализе потребностей пользователей и применении современных технологий, позволит существенно повысить качество и доступность оказываемой помощи, способствуя улучшению жизни людей с ограниченными возможностями здоровья.

# Список литературы

1. Педагогика дополнительного образования. Работа с детьми с особыми образовательными потребностями учеб. пособие для вузов / под ред. Л.В. Байбородовой. - 2-е изд., испр. и доп. - М. : Юрайт, 2020. - 241 с. - (Высшее образование)
2. Глухов В.П. Специальная педагогика и специальная психология : учеб. для вузов / В.П. Глухов. - 3-е изд., испр. и доп. - М. : Юрайт, 2020. - 323 с. - (Высшее образование))
3. Годовникова Л.В. Психолого-педагогическое сопровождение обучающихся с ОВЗ : учеб. Пособие для вузов / Л.В. Годовникова. - 2-е изд. - М. : Юрайт, 2020. - 218 с.
4. Основы комплексного сопровождения детей с ограниченными возможностями здоровья : метод. реком. / сост. М.А. Нугайбекова. – М., Берлин : Директ-Медиа, 2020. – 141 с.
5. (Подольская О.А. Теория и практика инклюзивного образования : учеб. пособие / О.А. Подольская, И.В. Яковлева. – М., Берлин : Директ-Медиа, 2018. - 202 с. URL: http://biblioclub.ru/index.php?page=book&id=494762) (дата обращения: 4.06.2025).
6. Как развернуть приложение на Symfony [Электронный ресурс] Режим доступа: URL: https://symfony.ru/doc/current/deployment.html (дата обращения: 04.06.2025)
7. Создание и использование шаблонов [Электронный ресурс] Режим доступа: URL: https://symfony.ru/doc/current/templates.html (дата обращения: 04.06.2025)
8. Context Question Answering [Электронный ресурс] Режим доступа: URL: https://docs.deeppavlov.ai/en/master/features/models/SQuAD.html (дата обращения: 04.06.2025)