

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ВЕБ-ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ ДЛЯ
ВЫЯВЛЕНИЯ НАРУШЕНИЙ КООРДИНАЦИИ ПРИ ПОМОЩИ
ПРОБЫ РОМБЕРГА**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Абу Саиф Марии Иссамовны

Научный руководитель

зав.каф.техн.пр., к. ф.-м. н., доцент _____

И. А. Батраева

Заведующий кафедрой

к. ф.-м. н., доцент _____

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

Современный мир тесно связан с цифровыми технологиями. Они встречаются в различных сферах: финансовой, образовательной, транспортной, производственной, медицинской и так далее. После прошедшей пандемии потребность в цифровизации медицинских процессов стала очевидна. Внедрение современных технологий в медицинскую сферу позволяет значительно ускорить процесс изучения анализов и повышает точность результатов различных тестов. Также автоматизация в медицине помогает уменьшить рабочую нагрузку и ускорить проведение приёма у врачей.

Внедрение цифровых технологий также упрощает жизнь обычных людей. Не у всех есть возможность постоянно проверяться в медицинских центрах — иногда из-за расположения населённого пункта, в котором проживает пациент, иногда из-за финансового состояния, а иногда просто нет возможности выделить несколько часов для посещения медицинского учреждения.

Наличие приложений для прохождения различных медицинских тестов в домашних условиях поможет пациенту узнать своё текущее состояние в течение нескольких минут, не покидая своего текущего местонахождения, не прерываясь надолго от своих дел. На данный момент невозможно создать такие приложения для всевозможных существующих тестов, но для каких-то отдельных сфер медицины это вполне реально. Например, в сфере неврологии для проверки равновесия человека используется проба Ромберга. Этот тест позволяет выявить нарушения координации, что особенно важно для пациентов с неврологическими заболеваниями, с травмами головы или возрастными изменениями.

При обычных обстоятельствах для прохождения пробы Ромберга нужно посетить врача, который вживую оценит координацию пациента. Однако, разработка мобильного или веб-приложения с автоматизированным анализом данных даёт возможность значительно ускорить процесс обследования, а также делает посещение врача не обязательным условием, для прохождения данного теста, что является достаточно важным фактором в современном мире. Это помогает пациенту заранее пройти самодиагностику, для дальнейшего понимания: есть ли необходимость посетить врача в ближайшее время или координация не была нарушена. Тест не является полной заменой приема у специалиста, но является дополнительным инструментом для диагностики собственного состояния.

Целью данной работы является разработка веб-интерфейса для приложения оценки равновесия с помощью пробы Ромберга. Для этого нужно:

- разработать легкий для понимания дизайн сайта;
- создать изображения для сайта;
- реализовать запись и загрузку видео, и его обработку в существующем ядре;
- организовать вывод данных по обработанному видео в виде графика.

Структура и объём работы. Выпускная квалификационная работа состоит из введения, 2 основных глав, заключения, списка использованных источников из 21 наименования и 3 приложений. Общий объём работы составляет 57 страниц, из них 34 страницы — основное содержание, включая 20 рисунков, USB-носитель в качестве одного приложения.

Первый раздел «Анализ предметной области» содержит основную теоретическую информацию о стеке технологий, которые использовались при разработке веб-приложения.

Второй раздел «Разработка сайта «Ромберговеденье» содержит подробное описание процесса разработки веб-приложения.

Анализ предметной области

Постановка задачи и описание приложения

Целью проекта является создание веб-приложения, позволяющего проходить тест Ромберга в домашних условиях. Данный метод является важным инструментом для диагностики отклонений в координации и часто применяется в следующих случаях:

- неврологические обследования;
- ортопедия и травматология;
- спортивная медицина;
- гериатрическая практика.

Проба Ромберга особенно важна при жалобах на головокружение, нарушениях походки, после черепно-мозговых травм и при подозрении на интоксикации (алкоголь, лекарственные средства).

Данный тест используется очень часто и разными людьми разных возрастов, и поэтому иметь возможность пройти его в любое время — очень полезно и даже важно.

Основные вычислительные процессы выполняются в ядре программы, написанной другими людьми, в данном случае нужно просто визуализировать результаты в веб-интерфейсе, адаптировав под данную платформу.

Постановка задачи:

- разработать минималистичный интерфейс;
- разработать рекомендации по использованию сайта;
- отрисовать изображения для сайта;
- реализовать три страницы: главная, инструкция и сканирование;
- реализовать возможность как записи видео, так и его загрузки;
- вывести данные по проанализированному видео;
- вывести визуализацию полученных данных в виде графика.

Графический редактор Figma

Figma — это облачный графический редактор для дизайна интерфейсов и прототипирования, в котором предусмотрен режим совместной работы над проектом. Этот редактор используется дизайнерами, разработчиками и продукт-менеджерами для создания цифровых продуктов: веб-сайтов, мобильных приложений, презентаций и бренд-ассетов.

Язык разметки HTML и язык стилей CSS

HTML — стандартизированный язык гипертекстовой разметки для создания веб-страниц. Он определяет структуру контента с помощью тегов, которые браузер преобразует в визуальные или функциональные элементы.

CSS — язык стилей, описывающий внешний вид HTML-документов. С его помощью можно задавать шрифты, цвета различных элементов, а также сложные анимации и адаптивный дизайн.

Библиотека React.js

React.js — это декларативная JavaScript-библиотека с открытым исходным кодом, предназначенная исключительно для построения пользовательских интерфейсов. Основная функциональная задача React сводится к отображению UI-компонентов на веб-странице и поддержанию их синхронизации с текущим состоянием приложения, поэтому для создания полноценного приложения только React.js обычно недостаточно — нужна интеграция с дополнительными инструментами и библиотеками.

Ключевое технологическое преимущество React заключается в его хорошей производительности, которая обеспечивается инновационной реализацией виртуального DOM. Проблема низкой производительности многих веб-приложений обычно связана с необходимостью частого выполнения ресурсоемких операций по модификации реального DOM-дерева. React решает эту проблему, используя легковесную виртуальную копию DOM (VDOM).

Библиотека React Router DOM

React Router DOM — стандартная библиотека маршрутизации между разными частями веб-приложения, созданными на React.

React Router предоставляет интуитивно понятный способ описания маршрутов и обеспечивает плавную навигацию между компонентами-страницами. В отличие от классических веб-сайтов с перезагрузкой страниц, эта библиотека предназначена для одностраничных приложений (SPA) и использует браузерное History API для синхронизации URL с интерфейсом, что позволяет мгновенно переключать контент без обновления страницы.

Язык программирования JavaScript

JavaScript — это язык программирования, который обеспечивает динамическое поведение веб-страниц, например, отображение периодически обновляемого контента или воспроизведение каких-либо анимаций. В отличие от статичного отображения контента, обеспечиваемого HTML и CSS, которые образуют первые два уровня веб-разработки, JavaScript добавляет третий, интерактивный слой, отвечающий за сложную функциональность.

Фактически, любая интерактивность на современной веб-странице — от простых всплывающих уведомлений до сложных веб-приложений — в основном реализуется средствами JavaScript. Это делает его незаменимым инструментом для современного веб-разработчика, дополняющим базовые технологии разметки и стилизации.

Библиотека Recharts

Recharts — это React-ориентированная библиотека для визуализации данных, основанная на возможностях D3.js. Recharts является удобным инструментарием для создания графиков и диаграмм с использованием современных веб-технологий (HTML, SVG и CSS). Для рендеринга используется SVG, что обеспечивает четкое отображение на любых устройствах.

Фреймворк FastAPI

FastAPI — это современный инструмент для разработки веб-API, сочетающий высокую скорость работы с интуитивно понятным синтаксисом. Его архитектура изначально создана под асинхронную обработку запросов, что обеспечивает эффективное использование серверных ресурсов при работе с высокой нагрузкой.

Язык программирования Python

Python — это высокоуровневый язык программирования общего назначения, который относится к интерпретируемым языкам, то есть написанный на Python код интерпретируется в момент обращения программой-интерпретатором без предварительной компиляции. Python отличается эффективностью, простотой и универсальностью использования. За счет простого и интуитивно понятного синтаксиса является одним из распространенных языков для обучения программированию.

Разработка сайта «Ромберговедение»

Цветовая палитра сайта

Прежде всего нужно было определиться с цветовой палитрой сайта. Так как сайт тематически связан с медициной, нужно было выбрать соответствующую палитру.

Основные цветовые палитры для медицинских сайтов: классическая бело-синяя гамма, зеленые оттенки, светло-голубая палитра.

Акцентные сочетания: красный акцент, фиолетовые тона, теплые бежевые оттенки.

Запрещенные цветовые приемы: ядовито-зеленые и кислотные оттенки, полностью белые схемы без смягчающих оттенков, темно-коричневая гамма.

Для фармацевтических компаний рекомендуются холодные сине-серебристые гаммы, подчеркивающие научный подход. Детские клиники часто добавляют мягкие желтые акценты для создания дружелюбной атмосферы.

Спустя большое количество просмотренных на разных сайтах цветовых палитр, выбор был сделан в пользу цветовой палитры состоящей из пастельных оттенков фиолетового и голубого. Использовалась именно такая палитра, потому что цвета на ней были визуальны приятны и с их можно было спокойно адаптировать под вид, нужный сайту.

Макет сайта

После выбора цветовой палитры сайта нужно было определиться со внешним видом сайта, то есть понять, сколько страниц будет на нём, и какие элементы будут расположены на страницах и в каком порядке. Для этого в Figma был разработан макет сайта — макет каждой страницы сайта с примерным расположением элементов.

Было решено разрабатывать именно макет сайта, а не его дизайн, потому что точно не было понятно, какие дизайнерские решения получится реализовать, а какие нет. В теории дизайн мог включать в себя множество интересных элементов и красивых элементов, но выбранные программные средства могли не содержать нужного функционала для их реализации.

При создании макета стало понятно, что достаточно трёхстраничного сайта для данного проекта. Первая страница — «Домашняя страница» нужна для приветствия зашедшего на сайт пользователя, поэтому на ней должна была

содержаться информация о том, что для чего нужен этот сайт и как использовать его функционал.

Содержимое сайта

После получения примерного макета сайта, составлялся файл с точным текстом для инструкции на сайт, а также искался текст для информации на домашней странице. После того, как вся информация была точно составлена, отрисовывались svg-изображения для размещения на сайте.

На изображениях нужно было изобразить человека, чтобы в дальнейшем отрисовать его для карточек с правилами. Персонаж также был отрисован в основных цветах сайта.

Структура проекта

Есть главная папка с названием проекта, в которой находятся две папки, отвечающие за фронтенд и за бэкенд сайта, с соответствующими названиями — «frontend» и «backend».

В папке «frontend» внутри папки «src» содержатся все компоненты и элементы, необходимых для их работоспособности. Все компоненты хранятся в папке «components», внутри которой для каждой компоненты создана собственная папка, в которой хранятся два файла — сама компонента в jsx-файле и её стили в css-файле.

В папке «backend» файл «main.py» содержит бэкенд сайта, а в папке «core» находится ядро — обрабатывающая видео программа, разработанная двумя студентами, обучающимися на направлении «Программная инженерия». Бэкенд вызывает функцию «process_video» из файла «rose.py», который находится в этой папке, для получения необходимых данных о видео.

Компонента App

Компонента App отвечает за сборку и отображение всего фронтенда. Именно в этой компоненте организована маршрутизация, а также в ней закреплены постоянные компоненты Header и Footer. Маршрутизация организована при помощи «react-router-dom».

Компоненты Header и Footer

Компонента «Header» представляет собой заголовок страницы, на котором расположены 4 кнопки, позволяющие перемещаться по сайту. Слева сбоку

Располагается логотип, при нажатии на который можно перейти на домашнюю страницу. Справа от логотипа, ближе к середине компоненты, находится группа из трёх кнопок: «Главная» — отправляет на домашнюю страницу, «Инструкция» — эта кнопка перенаправляет на страницу с правилами сайта, и «Сканировать» — отправляет на страницу с основными функционалом. Компонента «Header» прикреплена к верху видимой области, поэтому она не будет исчезать при пролистывании страницы вниз.

Компонента «Footer» представляет собой блок с информацией, которую пользователь видит при пролистывании страницы до конца. На данный момент на этой компоненте находится информация, никак не относящаяся к сайту, то есть просто данные для заполнения пустого пространства, не несущие в себе никакой конкретной информации.

Компоненты HomePage1 и HomePage2

Компоненты «HomePage1» и «HomePage2» — это два экрана, каждый размером в одну видимую область. Они вызываются в компоненте «HomePage» друг за другом, формируя одну длинную страницу.

«HomePage1» состоит фактически из трёх блоков: в левой стороне текстовый блок и блок с навигационными кнопками, с правой стороны расположен блок изображения.

«HomePage2» состоит из двух блоков: в левой стороне текстовый блок, состоящий из нумерованных списков, с правой стороны расположен блок изображения.

Компонента InstructionPage

Компонента «InstructionPage» состоит из трёх частей: заголовка страницы, блока карточек, который состоит из двух рядов, и ещё одного блока текста. На карточках пошагово изображён план работы с приложением.

Карточка для компоненты «InstructionPage» также является отдельной компонентой, состоящей из изображения сверху и текстовой части снизу. Компонента «InstructionCard» получает в props путь к изображению, заголовок карточки и её текстовое содержимое, которое дальше помещается в конструктор карточки, наполняя её информацией.

Компонента ScanningPage

Компонента «ScanningPage» отвечает за страницу, на которой происходит запись/загрузка видео и отображение графов по обработанным видео. Сверху у страницы есть переключатель между камерой и графом, однако страница с отображением графов недоступна, пока не будут обработаны видео хотя бы один раз.

Переключение между компонентой камеры и компонентой графа происходит благодаря записываемым в `useState` состояниям. Если в `setActiveTab` содержится значение «camera», то страница с графом недоступна. После успешной обработки двух видео на бэкенде и возвращения по ним статуса «success» вкладка с графом становится доступной.

Компонента Camera

Компонента «Camera» состоит из текста, объясняющего пользователю, что от него ожидает приложение, экрана камеры, которая всегда изначально выключена, а также кнопки загрузки видео. После включения камеры пользователю становится доступна кнопка записи видео, с помощью которой можно в реальном времени записать видео, чтобы потом отправить его на проверку. Запись видео можно остановить до того, как закончится таймер, но для более точной работы приложения лучше следовать всем подсказкам, расположенным на сайте. На сайте также доступна загрузка видео из файлов, которые есть на устройстве. Для этого нужно нажать кнопку «Загрузить видео» и выбрать нужный файл в открывшемся окне проводника.

Записанные видео перед отправкой на сервер можно пересмотреть при помощи проигрывателя, который появляется под панелью управления записи видео. Если что-то пошло не так во время записи, или просто нужно заменить какое-то записанное или загруженное видео, то это можно сделать, выбрав нужную кнопку над проигрывателем видео и нажать кнопку «Начать запись» или «Загрузить видео».

Компонента Graph

Компонента «Graph» визуализирует данные о позе человека, через анализ углов и смещения различных частей тела.

Компонент анализирует и визуализирует:

- углы между ключевыми точками тела (суставами);

- смещения средней линии тела (плечи и бёдра);
- диагностирует возможные нарушения осанки и равновесия.

Компонента выводит данные, полученные после обработки видео. В левом блоке выводится вся текстовая информация, которая была получена из ядра, вся оставшаяся правая сторона отводится для вывода графа.

Файл `main.py`

Весь бэкенда данного сайта находится в файле `main.py`.

Внутри файла данного файла создаётся экземпляр `FastAPI`. Далее настраиваются директории, в которые будут сохраняться все данные по видео, а также сами видео. Основной эндпоинт для обработки видео, принимающий через `POST`-запрос два видео-файла — `upload_videos`.

Логика обработки данных:

- проверка `MIME`-типа;
- генерация уникальных `ID` для файлов;
- сохранение оригинальных видео в `UPLOAD_DIR`;
- обработка каждого видео через `process_single_video()`;
- возврат `JSON` с путями к результатам.

Функция обработки одного видео — `process_single_video()` генерирует пути для выходных файлов:

- `plot.png` — график;
- `elipsis.png` — изображение с эллипсами;
- `analysis.txt` — текстовый анализ;
- `result.json` — `JSON` с результатами;
- вызывает функцию `process_video` из модуля `pose`;
- возвращает имена созданных файлов

Эндпоинт для получения результатов:

```
1 @app.get("/results/{result_type}/{filename}")
```

- позволяет получить конкретный файл результата по типу и имени;
- поддерживаемые типы: `plot`, `elipsis`, `analysis`, `json`;
- использует `FileResponse` для возврата файла с правильным `Content-Type`

ЗАКЛЮЧЕНИЕ

В процессе работы над проектом были:

- проанализированы различные варианты дизайнов и цветовых палитр для сайта;
- разработан и реализован прототип для понимания и достаточный для хорошего функционирования дизайн;
- подобран удобный стек технологий;
- реализовано веб-приложение, позволяющее записывать или загружать видео;
- интегрирован модуль для анализа видео и были получены результаты, которые пользователь мог увидеть на сайте.

Таким образом, все задачи были выполнены. В результате работы было разработано веб-приложение «Ромберговедение».