

Муниципальное автономное общеобразовательное учреждение
средняя общеобразовательная школа №9

Автоматическая система
психологического тестирования
исследовательский проект

Исполнитель:
обучающаяся 11А класса
Клещева Елена Владимировна
Руководитель:
учитель информатики
Коротеев Антон Геннадьевич

Содержание

Введение	3
1 Психологические тесты	4
1.1 Тесты, их классификация	4
1.2 Основные характеристики психологических тестов	4
1.3 Достоинства тестов	5
1.4 Недостатки тестов	6
1.5 Процесс проведения психологического тестирования в школе	6
2 Разработка автоматической системы психологического тестирования	7
2.1 Разработка дизайна сайта	7
2.2 Основы HTML-вёрстки	13
2.3 Основы CSS	15
2.4 Специальные возможности – JavaScript	17
2.5 HTML-вёрстка сайта по разработанному макету	19
2.6 Домен и хостинг	23
2.7 Основы PHP-программирования	24
2.8 База Данных	25
2.9 Программирование сайта с использованием PHP и БД	26
2.10 Программирование автоматической системы психологического тестирования ..	29
3 Проведение психологического тестирования с использованием разработанной автоматической системы	37
Заключение	40
Список источников	41

Введение

Для усовершенствования процесса обучения и образования, повышения успеваемости, рационального распределения учебной нагрузки необходимо проводить мониторинг психического состояния учащихся образовательных учреждений с целью выявления причин низкого усвоения знаний, оценки интеллектуального развития, выявления проблем адаптации в коллективе, а также решения проблем профориентации. Наиболее распространенная форма такого мониторинга – психологическое тестирование, представляющее собой метод измерения и оценки психологических характеристик человека с помощью специальных техник.

В школах психологическое тестирование обычно проводится массово на классных часах, и из-за таких условий нередко возникает вопрос о достоверности выявленных индивидуально-психологических отличий отдельного респондента. Классные руководители вынуждены вручную обрабатывать результаты тестирований. На данное мероприятие педагог тратит много ценного времени. Большой объем рабочего материала сказывается на загруженности учителя, а усталость от монотонной работы с большим количеством чисел приводит к вычислительным ошибкам и недостоверности оценки способностей тестируемого учащегося.

Альтернативой традиционным ручным тестированиям является развитие и использование онлайн-тестирования. Учащиеся смогут сами выбирать время и обстановку для своего психологического тестирования, что положительно отразится на истинности результатов. Онлайн-тестирование сможет облегчить жизнь и классным руководителям, оно существенно сократит траты времени на сбор и обработку результатов. А автоматизация процесса при помощи компьютерных вычислений позволит минимизировать ошибки в процессе обработки результатов.

Цель.

Автоматизировать процесс проведения психологического тестирования в образовательных учреждениях.

Задачи

1. Разработать систему автоматического психологического тестирования.
2. Провести психологическое тестирование с использованием разработанной автоматической системы.
3. Проанализировать результаты тестирований.

1 Психологические тесты

1.1 Тесты, их классификация

Психологическое тестирование представляет собой исследование определённых психологических качеств и свойств личности путем использования психологических тестов.

Что такое психологические тесты?

Тесты – это специализированные методы психологического диагностического исследования, применяя которые можно получить точную количественную или качественную характеристику изучаемого явления.

От других методов исследования тесты отличаются тем, что:

- 1) предполагают стандартизованную, выверенную процедуру сбора и обработки данных, а также их интерпретации;
- 2) с помощью тестов можно изучать и сравнивать между собой людей, давать оценки их психологии и поведению.

Для выявления психофизиологических и личностных характеристик, знаний, умений и навыков испытуемого, различают следующие типы тестов:

- Тест-опросник основан на системе заранее отобранных и проверенных с точки зрения их валидности и надежности вопросов, по ответам испытуемых, на которые определенно можно судить об их психологических качествах
- Тест-задание предполагает оценку психологии и поведения человека не на основе того, что он говорит, а на базе того, что он делает. В тестах этого типа человеку дается серия специальных заданий, по итогам, выполнения которых судят об изучаемом качестве.
- Проективные тесты обычно предназначены как раз для изучения тех психологических и поведенческих особенностей человека, которые им слабо осознаются или вызывают к себе с его стороны крайне отрицательное отношение.

1.2 Основные характеристики психологических тестов

Основные характеристики психологических тестов - это валидность, надежность, репрезентативность и достоверность.

- *Валидность* – это соответствие результатов теста той характеристике, для измерения которой он предназначен.

- *Надёжность* – это свойство теста давать при повторном измерении близкие результаты. Надёжность как внутренняя согласованность - направленность всех элементов тестовой шкалы на измерение одного качества.
- *Репрезентативность* – это соответствие между нормами (интервалами на тестовой шкале), полученными на выборке, и нормами, которые могут быть получены на популяции.
- *Достоверность* – это свойство теста противодействовать фальсификации - намеренному или бессознательному искажению результатов испытуемыми.

1.3 Достоинства тестов

Популярность метода тестов объясняется следующими главными его достоинствами.

1. Стандартизация условий и результатов

Тестовые методики относительно независимы от квалификации пользователя (исполнителя).

2. Оперативность и экономичность

Типичный тест состоит из серии кратких заданий, на выполнение каждого из которых требуется, как правило, не более полуминуты, а весь тест занимает не более часа (в школьной практике это один урок); тестированию одновременно подвергается сразу группа испытуемых, таким образом, происходит значительная экономия времени (человеко-часов) на сбор данных.

3. Количественный дифференцированный характер оценки

Дробность шкалы и стандартизованность теста позволяют рассматривать его как «измерительный инструмент», дающий количественную оценку измеряемым свойствам (знаниям, умениям в данной области). Кроме того, количественный характер тестовых результатов дает возможность применить в случае тестов хорошо разработанный аппарат психометрии, позволяющий оценить, насколько хорошо работает данный тест на данной выборке испытуемых в данных условиях.

4. Оптимальная трудность

5. Надежность

6. Справедливость

7. Возможность компьютеризации

В результате компьютеризации повышаются все параметры тестирования (например, при адаптивном компьютерном тестировании резко сокращается время тестирования). Также компьютеризация – это мощный инструмент обеспечения информационной безопасности (достоверности диагностики).

8. Психологическая адекватность

Это важнейшее психологическое следствие оптимальной сложности. Наличие в тесте (по сравнению с традиционными экзаменационными вариантами) большого количества кратких заданий средней трудности дает многим испытуемым (особенно тревожным, не уверенным в себе) шанс «зацепиться», поверить в себя, активизировать психологически оптимальную установку «на преодоление».

1.4 Недостатки тестов

Как всегда, определенные недостатки метода тестов являются продолжением его достоинств.

1. Опасность «слепых» (автоматических) ошибок
2. Потеря индивидуального подхода
3. Отсутствие доверительной обстановки
4. Необоснованная сложность

1.5 Процесс проведения психологического тестирования в школе

Рассмотрим стандартное тестирование школьников на классном часе.

Есть два варианта развития событий. Первый, и на мой взгляд, самый благоприятный вариант – классный руководитель раздаёт распечатанный материал каждому ученику для самостоятельного прочтения и выполнения тестовых заданий. Второй – классный руководитель пишет на доске варианты ответов и начинает зачитывать вопросы из методики, что естественно не нравится респондентам, так как не каждый успевает «переварить» вопрос и выбрать оптимальный для него вариант ответа. Просьба повторить вопрос вызывает раздражение как у проводящего тестирование, так и у тех, кто уже ответил на данный вопрос. В это время создается стрессовая атмосфера, которая непосредственно негативно сказывается на результаты тестирования. Несовершенство сложившейся системы приводит к предвзятому отношению к психологическому тестированию со стороны учащихся.

Классным руководителям же приходится не только проводить тестирование, но и обрабатывать результаты и составлять отчёты. Данная работа очень трудозатратна и занимает непозволительно много времени. А нерациональное расходование временного

ресурса, следовательно, сказывается на производительности труда педагога. К тому же ручная обработка тестов чревата, к сожалению, неизбежными вычислительными ошибками, допускаемыми из-за усталости или невнимательности человека.

Таким образом, можно сделать вывод, что реализация традиционного ручного тестирования далека от совершенства, она вызывает недоверие со стороны школьников и отнимает много трудового времени у классных руководителей. И из-за ручного труда возможны ошибки и неточности в формировании отчета по личным качествам учащихся. Возникает необходимость в автоматизации процесса проведения подобных тестирований.

2 Разработка автоматической системы психологического тестирования

2.1 Разработка дизайна сайта

С чего начинается разработка дизайна сайта?

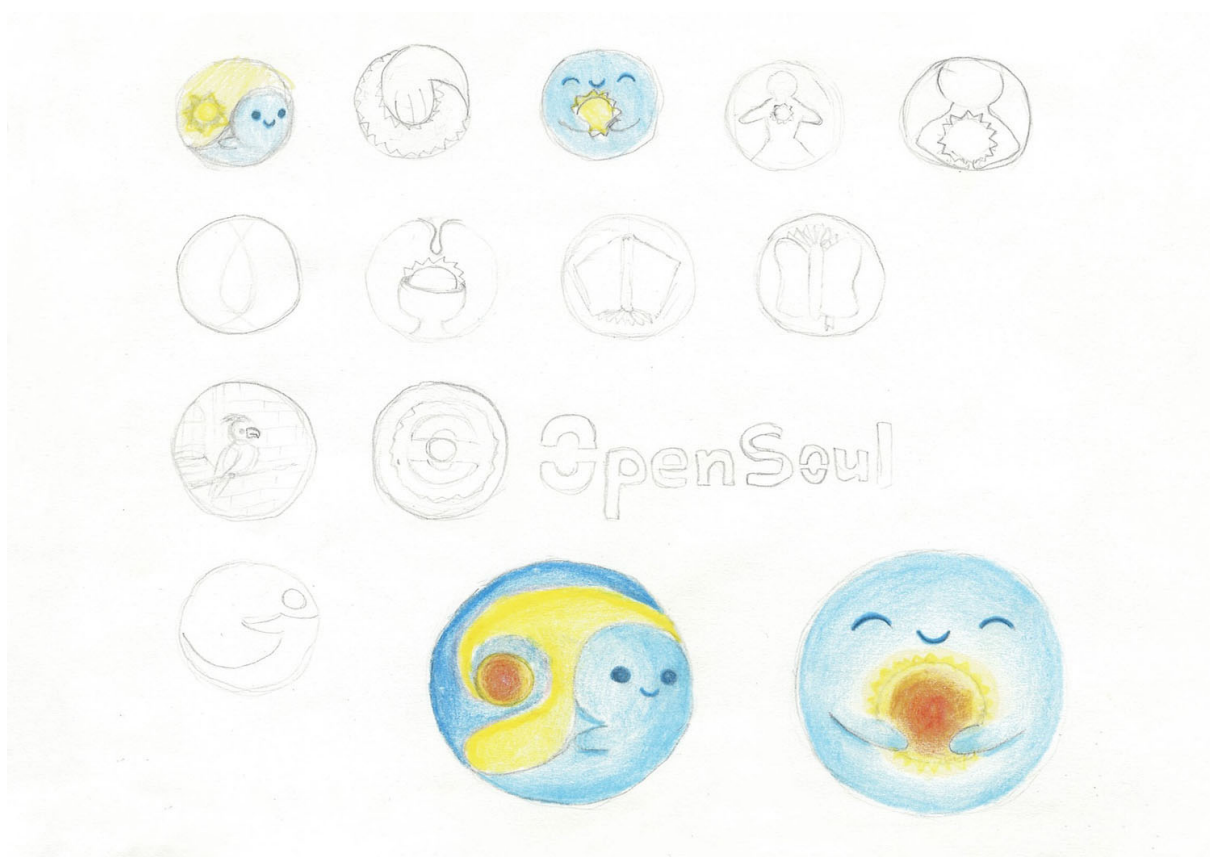
Конечно с его отличительных и хорошо запоминающихся деталей, а именно с названия сайта и его логотипа.

Начнем с имени проекта. *(Важное примечание: имя проекта и доменное имя сайта разные понятия; в данном пункте будет рассмотрен только выбор имени для проекта)* Разумеется, оно должно быть говорящим и передавать суть проекта, хотя бы частично, но на этом этапе главное не перемудрить, чтобы случайно не ввести пользователей в заблуждение. Чем проще звучит название проекта, чем меньше в нем сложных для понимания слов и оборотов, тем лучше оно воспринимается. Аббревиатуры, сокращения, длинные слова и словосочетания в названиях творческих или исследовательских проектов не приветствуются.

«OpenSoul» (с англ. «Открытая душа») отлично справляется в роли названия проекта. Простота и лаконичность отражаются в легкости запоминания. Ассоциативность настраивает доверительные отношения с пользователем сайта.

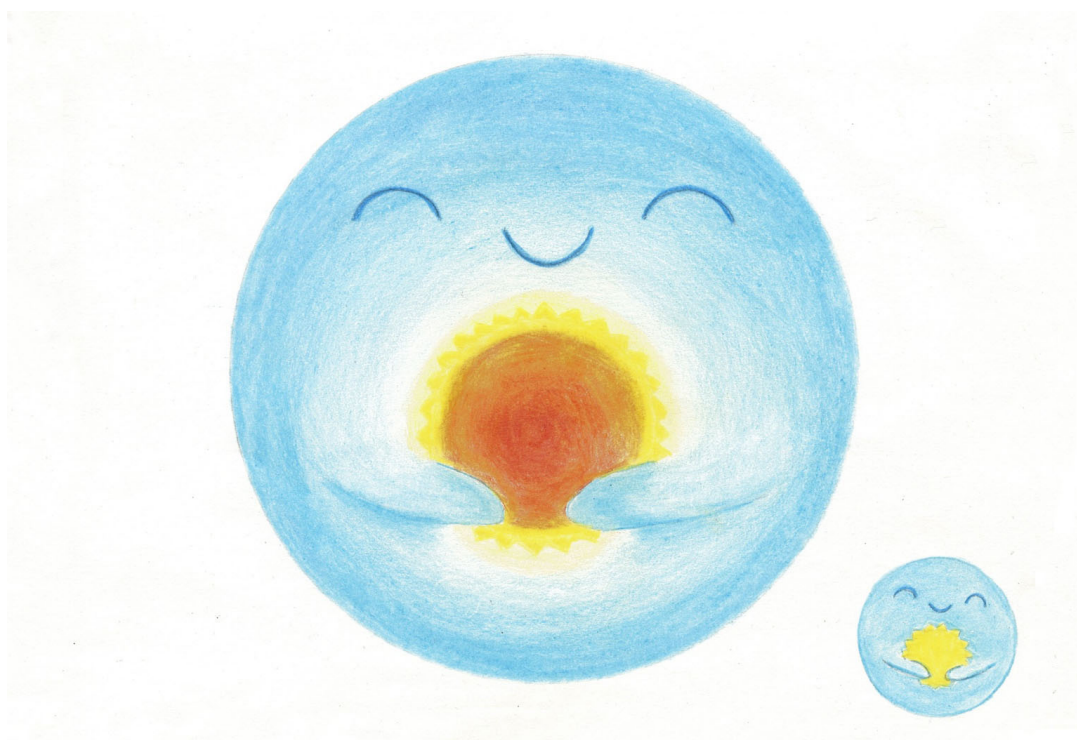
Когда имя проекта подобрано, самое время начать думать над дизайном логотипа. Этот этап поистине требует творческого мозгового штурма.

Первые варианты логотипа:



В качестве логотипа было решено создать безымянного маскота, являющего олицетворением внутреннего мира человека, то есть души.

Эскиз в цвете:



Когда эскиз логотипа готов, его необходимо реализовать в векторное изображение.

Но почему именно в векторное изображение? Что это такое? В чем его преимущество?

Векторная графика – это метод создания компьютерного изображения, которое строится посредством обработки математического описания в специальном программном обеспечении. В отличие от растрового рисунка, который представляет набор пикселей разных цветов, векторное изображение – это набор примитивов (линий, кривых, квадратов, кругов), описанных математическими формулами. Рисунок формируется в виде файла, который содержит данные о координатах точек и о линии, пересекающей опорные точки. Кроме этого, файл включает информацию о примененных эффектах.

Такой принцип построения цифрового изображения обеспечивает возможность его масштабирования, изменения размера без снижения качества, изменения формы и перегруппировки примитивов для создания новых рисунков на базе исходных объектов. Кроме этого, такой файл занимает намного меньше места по сравнению с растровым, в котором содержится значительный объем сведений, включая данные о каждом пикселе, его координатах, цвете и других параметрах.

Для создания векторного изображения я использовала Adobe Illustrator.

Adobe Illustrator – программа, которая используется художниками-иллюстраторами для создания веб-графики. В отличие от знаменитого Adobe Photoshop, Illustrator работает с векторными изображениями, а не растровыми.

Векторное изображение логотипа:



Когда имя и логотип готовы, самое время начинать думать о том, как будет выглядеть сайт в браузере.

Начнем с цветовой палитры сайта.

Цвет — один из фундаментальных признаков, на которых основан дизайн. В руках профессионала он может стать мощным инструментом. Он влияет на многие факторы, которые играют большую роль в визуальном восприятии. Цвет оказывает огромное воздействие на наше сознание, он меняет наше отношение к любому предмету буквально за секунды, а также заставляет людей реагировать на него и даже предпринимать определенные действия.

Рассмотрим самые распространенные эмоциональные отзывы на цвета.

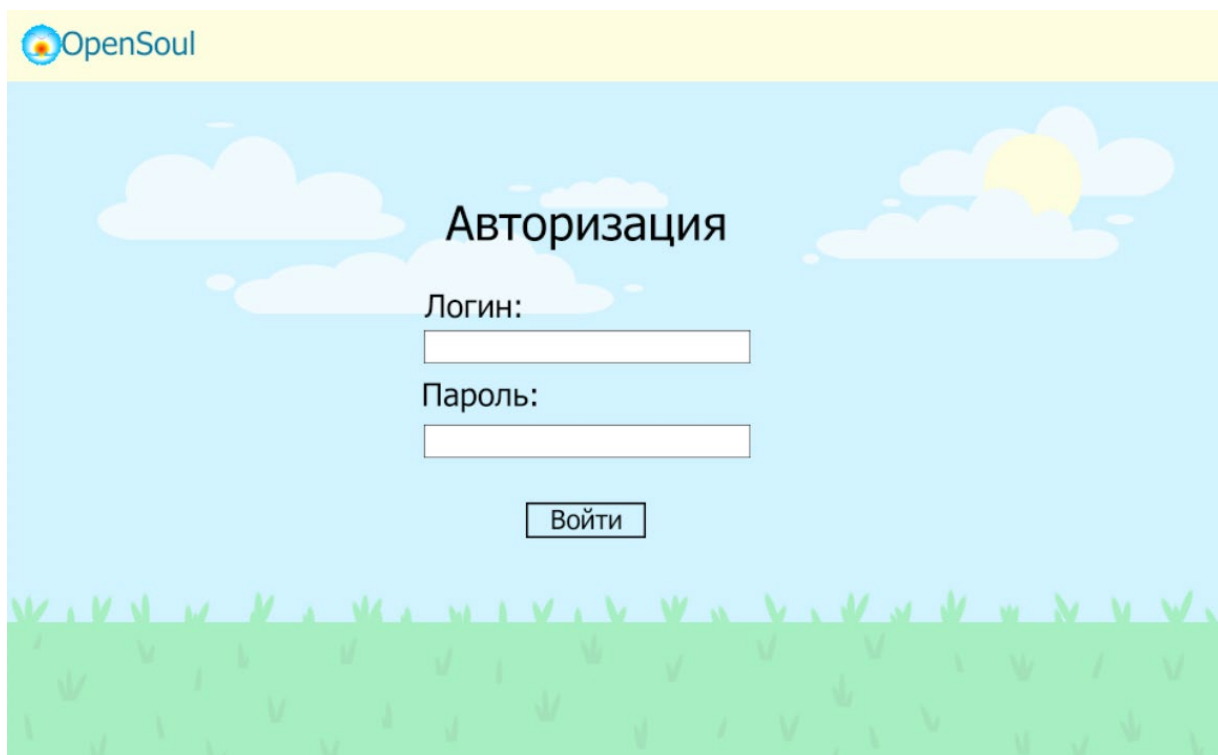
- Красный вызывает чувства азарта, смелости, желания, пробуждает лидерское начало. Но не злоупотребляйте им, оформляя сайт, ведь цвет так силен, что может вызвать и чувство тревоги, опасности.
- Голубой настраивает на спокойствие, формирует чувство мира, надежности, любви, стабильности. Если хотите вызвать доверие к профессионализму специалистов или качеству товаров, представленных веб-ресурсом, обязательно используйте возможности голубого цвета. Особенно это касается сайтов, основная масса посетителей которого – мужчины.
- Желтый пробуждает энергичность, любопытство, ассоциируется с радостью, развлечениями, интеллектом. Но! Старайтесь не перенасытить ресурс этим цветом – пробудить чувство осторожности.
- Оранжевый вызывает ощущение бодрости, стимулирует к творчеству, дружелюбию, игривости, дает чувство уверенности, даже мужества и стойкости.
- Фиолетовый будит честолюбивые мысли, ассоциируется с властью, мудростью, роскошью, магией и тайной.
- Зеленый – не только цвет жизненной энергии, но и вызывает мысли о деньгах, при этом ассоциируется с гармонией, природой, исцелением.
- Коричневый сопутствует чувству отдыха, уверенности, стабильности. Он означает долговечность, приземленность, природу, комфорт и надежность.
- Серый – цвет консерваторов. Он способствует возникновению чувства доверия, серьезности, честности. Вызывает чувство, консерватизма и традиционализма. Пробуждает ощущение чистоты и невинности.

- Розовый хорош для формирования таких женственных эмоций, как романтичность, воспитание, слабость, пассивность, привязанность. На веб-сайте для мужчин будет неуместен и не способствует притоку пользователей.
- Белый – цвет чистоты, простоты и доброты.
- Черный цвет связан с властью, элегантностью и стилем. Отличный цвет для придания дизайну перспективы и глубины. Используйте умеренно, чтобы не вызвать у посетителя тяжелых чувств.

Проанализировав психологические особенности восприятия цвета и соотнеся их с тематикой сайта, я вывела основную палитру:

- Голубой
- Желтый
- Зеленый

Теперь можно приступить к созданию макета сайта. Для этого я воспользовалась программой Adobe Photoshop и спроектировала внешний вид трех разделов: форма входа, «Тестирование», «Результаты», «О проекте»:





Тест «Самоанализ (анализ) личности»

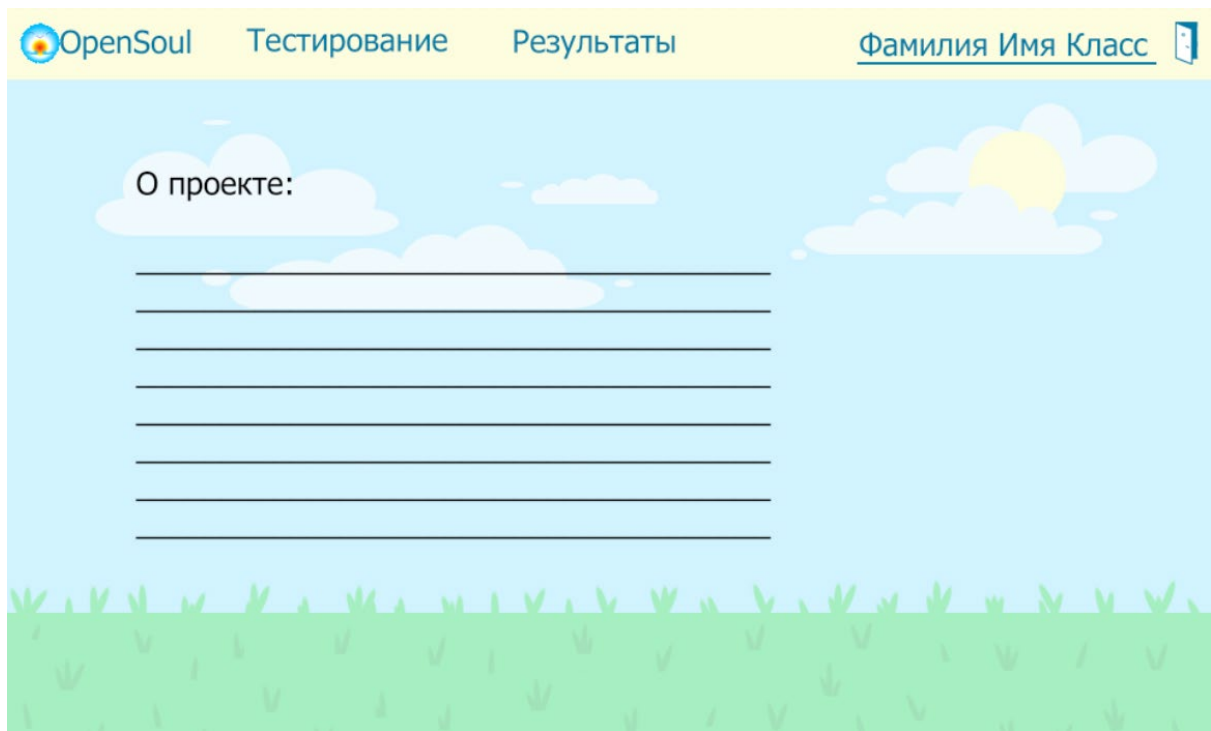
Тест на самооценку

Тест «Мотивация учения и отношения учащихся к процессу познания»

Тест «Оценка коммуникативно-организаторских способностей»



Тест «Самоанализ (анализ) личности» (Выполнен)

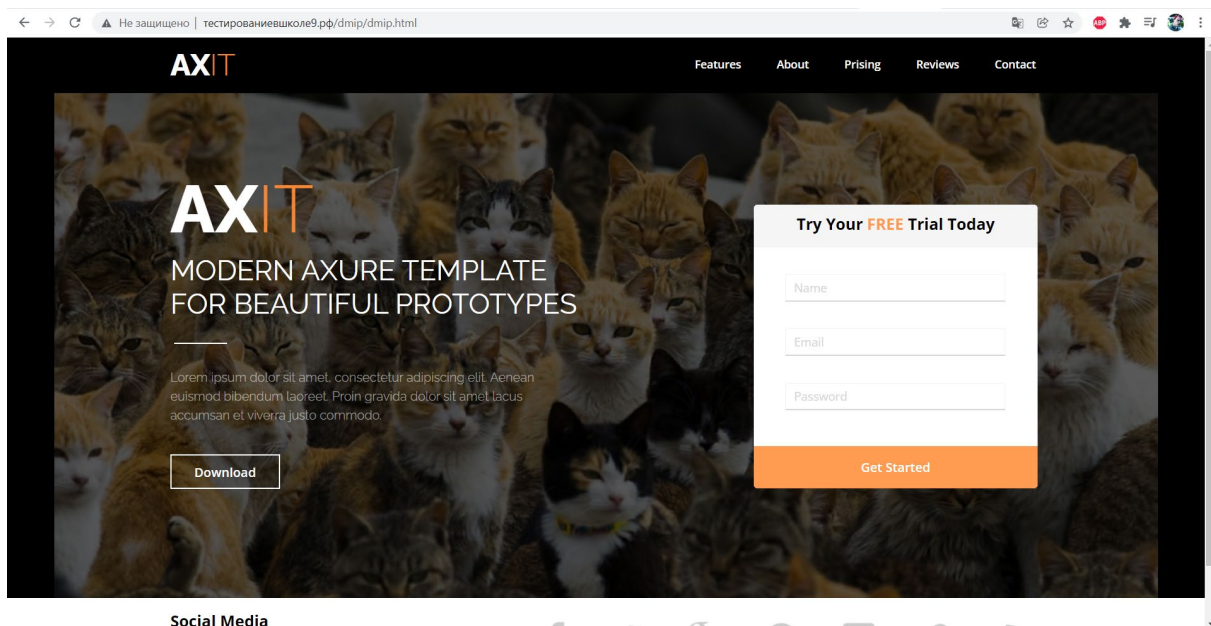


Макет успешно создан, но как его воплотить в код?

2.2 Основы HTML-вёрстки

HTML — это первое, с чем сталкиваются начинающие веб-разработчики.

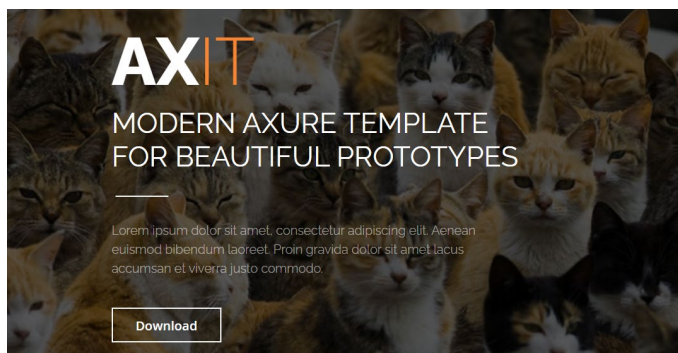
Когда мы открываем страницы сайтов, перед нами, обычно, красивые картинки, текст, иногда реклама и другие элементы. Например, рассмотрим мою последнюю конкурсную HTML-вёрстку:



Если вызвать контекстное меню и нажать на пункт «Исходный код страницы» (в зависимости от браузера и языка системы называется по-разному), то мы увидим примерно следующее:

```
<div class="header_information">
  <div class="logo2"><span style="color: #ffffff; font-family: RalewayExtraBold;">AX</span><span style="color: #f68636; font-family:
  RalewayLight">IT</span></div>
  <div class="tagline">MODERN AXURE TEMPLATE FOR BEAUTIFUL PROTOTYPES</div>
  <div class="divider"></div>
  <div class="your_text">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida
  dolor sit amet lacus accumsan et viverra justo commodo.</div>
  <div class="download_button"><a href="">Download</a></div>
</div>
```

Отображение в браузере:



Именно такой код и называется HTML. Hypertext Markup Language (HTML) — это язык разметки для создания веб-страниц или веб-приложений.

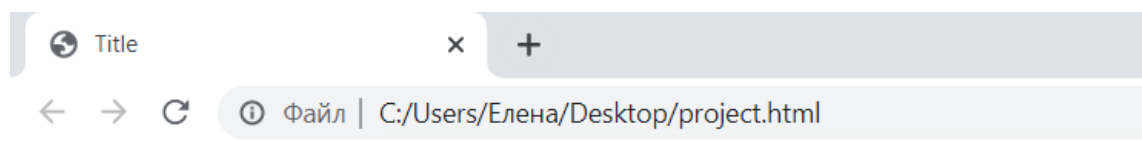
Проще всего понять идею HTML на примере редактирования текста в редакторе. Когда нам необходимо выделить текст жирным, создать список или вставить таблицу, мы «размечаем» текст. В обычных текстовых редакторах это делается визуально. Мы выделяем текст, затем нажимаем кнопку "Жирный", и текст становится жирным.

Так мы видим текст, когда открываем наш файл через специализированный редактор, понимающий его формат. Но внутри самого файла, конечно же, нет никакого жирного текста. Просто тот блок, который мы видим, как жирный, помечен специальным образом. При выводе разметка превращается в визуальные выделения.

В HTML такие выделения делаются с помощью специальных элементов HTML, которые являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты такие как интерактивная веб-формы могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Такие теги, как `<img />` и `<input />`, напрямую вводят контент на страницу. Другие теги, такие как `<p>`, окружают и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

Перед тем, как двигаться дальше, давайте попробуем создать нашу первую страничку. Для этого необходимо создадим файл `index.html` в любом удобном месте компьютера, добавить туда `html` код, указанный ниже, а затем открыть файл в браузере:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Title</title>
5    <meta name="description" content="My first page">
6  </head>
7  <body>
8    <h1>Hello, World!</h1>
9    <ul>
10     <li>One</li>
11     <li>Two</li>
12   </ul>
13 </body>
14 </html>
```



Hello, World!

- One
- Two

Подобным образом возможно создать сколько угодно страниц. Добавив на эти страницы ссылки, которые сплетают их в единое целое, на выходе получим статический сайт.

2.3 Основы CSS

В `HTML`-верстке существует еще одно понятие, такое как `CSS`.

`CSS` — `Cascading Style Sheets` — это каскадные таблицы стилей. По сути — язык, который отвечает за описание внешнего вида `HTML`-документа. Подавляющее большинство современных веб-сайтов работают на основе связки `HTML+CSS`.

Зачем же нужен этот `CSS`?

Всё просто: если HTML структурирует контент на странице, то CSS позволяет отформатировать его, сделать более привлекательным для читателя. Изначально веб-разработчики использовали исключительно HTML — так было на заре развития интернет-технологий. С помощью разметки можно было выделить параграф, заголовок, изменить начертание текста. А большего и не требовалось.

Со временем этих возможностей стало мало — и появилась технология форматирования без изменения самого содержимого и структуры документа. CSS решил проблему «зоопарка» тегов форматирования, когда разные браузеры поддерживали разные теги. Их унификация и единая база упростили работу с веб-документами и облегчили жизнь веб-мастерам.

CSS используется для определения стилей (правил) оформления документов — включая дизайн, вёрстку и вариации макета для различных устройств и размеров экрана. У такого способа форматирования несколько достоинств:

- теги не дублируются;
- документ проще обслуживать;
- внешний вид всего сайта можно изменить централизованно, а не корректировать форматирование каждой странички.

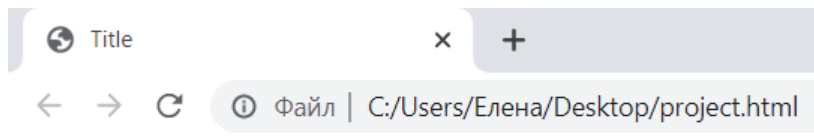
Стили можно разметить внутри тега `<HEAD>` или использовать отдельный CSS-файл.

Вот так можно прописать CSS в качестве атрибута непосредственно в HTML:

```
<p style="color: red;">Это важный текст.</p>
```

А так CSS прописывается при помощи тега `<style>` и в теге `<head>` документа HTML:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Title</title>
5  <style>
6      p {color: red;}
7  </style>
8  </head>
9  <body>
10     <p>Этот абзац будет красным.</p>
11 </body>
12 </html>
```



Этот абзац будет красным.

Теперь разберёмся, что такое CSS-файл. Всё просто — это файл с расширением .css, где прописываются правила оформления документа. Чтобы привязать файл к оформлению страницы, нужно использовать тег `<link>`:

CSS:

```
p {color: red;}
```

HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <p>Этот абзац будет красным.</p>
  </body>
</html>
```

Современный способ оформления веб-документов вошёл в практику в 2011 году. Это свойство CSS grid — теперь оно поддерживается практически всеми браузерами. И если раньше приходилось верстать документы с использованием элементов вроде `<div class="container">`, то сейчас всё это выполняется средствами CSS.

После того как этот способ стал стандартом, проблема разделения содержания (HTML) и оформления (CSS) решилась раз и навсегда.

2.4 Специальные возможности – JavaScript

JavaScript — это полноценный динамический язык программирования, который обычно применяется в качестве встраиваемого инструмента для программного доступа к различным объектам приложений.

С точки зрения веб-разработки, без знаний этой технологии невозможно заниматься созданием современных интерактивных сайтов. Язык JS — это то, что «оживляет» разметку страниц (HTML) и пользовательский функционал сайтов. С помощью этого языка реализуется возможность реакции страницы или отдельных ее элементов на действия посетителя.

Сегодня JavaScript является базовым языком программирования для браузеров. Он полностью совместим с операционными системами Windows, Linux, Mac OS, а также всеми популярными мобильными платформами.

Теперь нужно разобраться с тем, что такое JavaScript с точки зрения функционирования. Все пользовательские действия в окне браузера создают события, а программирование на JS позволяет обрабатывать их определенным образом.

Стандартный алгоритм работы выглядит следующим образом:

1. Пользователь выполнил определенное действие.
2. Браузер определил событие.
3. Активируется JS-код.
4. На странице происходит заданное изменение.

Задача программиста заключается в том, чтобы создать обработчики для всех событий, на которые должен реагировать сайт при взаимодействии с пользователем. Для наглядности приведу два примера типичных сценариев.

1. Пользователь нажимает левую кнопку мыши.
2. Браузер фиксирует событие onclick.
3. Активируется функция changePhoto.
4. В окне просмотра фотографий изменяется изображение.

Если обработчик не внедрен в код, то алгоритм работы будет примерно следующим:

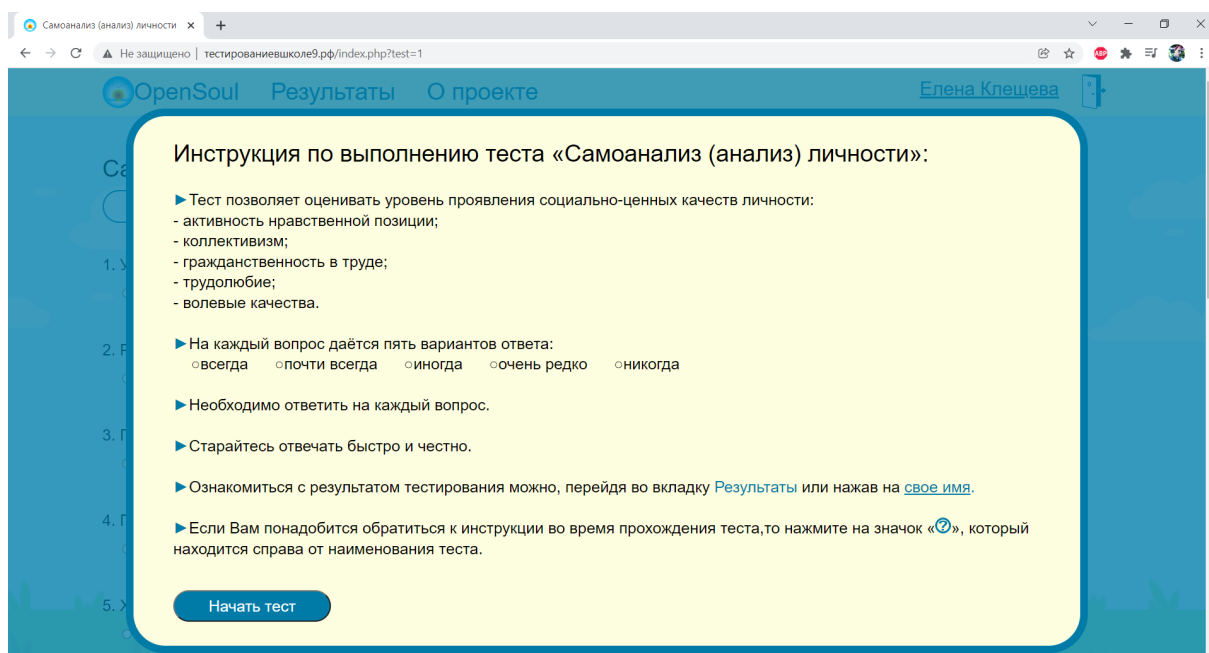
1. Пользователь совершает клик.
2. Браузер фиксирует событие onkeydown.
3. Для его обработки нет специального JS-кода.
4. После клика ничего не происходит.

Нужно отметить, что программирование не всегда привязывается к активным действиям пользователя. К примеру, JavaScript-код может срабатывать при полной загрузке страницы или после определенного времени нахождения на сайте. Эти возможности активно используются для создания всплывающих Pop-up элементов и чатов.

Я использовала JS для создания всплывающего окна, которое содержит в себе информацию о тесте.

Всплывающее окно автоматически возникает после нажатия на тест, который хотим пройти. Его также можно вызвать дополнительно при прохождении теста, нажав на значок вопроса «?», который находится справа от названия теста.

Вот так выглядит информационное всплывающее окно в браузере:



Реализация всплывающего окна с помощью JS:

```
<script type="text/javascript">
function myOnLoad(){
    var Okno = document.getElementById('instruction_button');
    Okno.onclick = function(e){
        if(document.getElementById('instruction').style.display == 'none'){
            document.getElementById('instruction').style.display = 'block';
        } else {
            document.getElementById('instruction').style.display = 'none';
        }
        e.preventDefault();
    }

    var OknoClose = document.getElementById('instruction_button_close');
    OknoClose.onclick = function(e){
        document.getElementById('instruction').style.display = 'none';
        e.preventDefault();
    }

    var StartOknoClose = document.getElementById('instruction_start_button_close');
    StartOknoClose.onclick = function(e){
        document.getElementById('instruction_start').style.display = 'none';
        e.preventDefault();
    }
}
</script>
```

2.5 HTML-вёрстка сайта по разработанному макету

Когда мы уже достаточно знаем о HTML и CSS можно приступить к вёрстке сайта по разработанному ранее в Photoshop макету.

В ходе работы я вносила коррективы в содержание макета. Так я убрала с главной панели вкладку «Тестирование» и перенесла ее основной функционал в раздел «OpenSoul», а также информацию «О проекте» вынесла в отдельную вкладку.

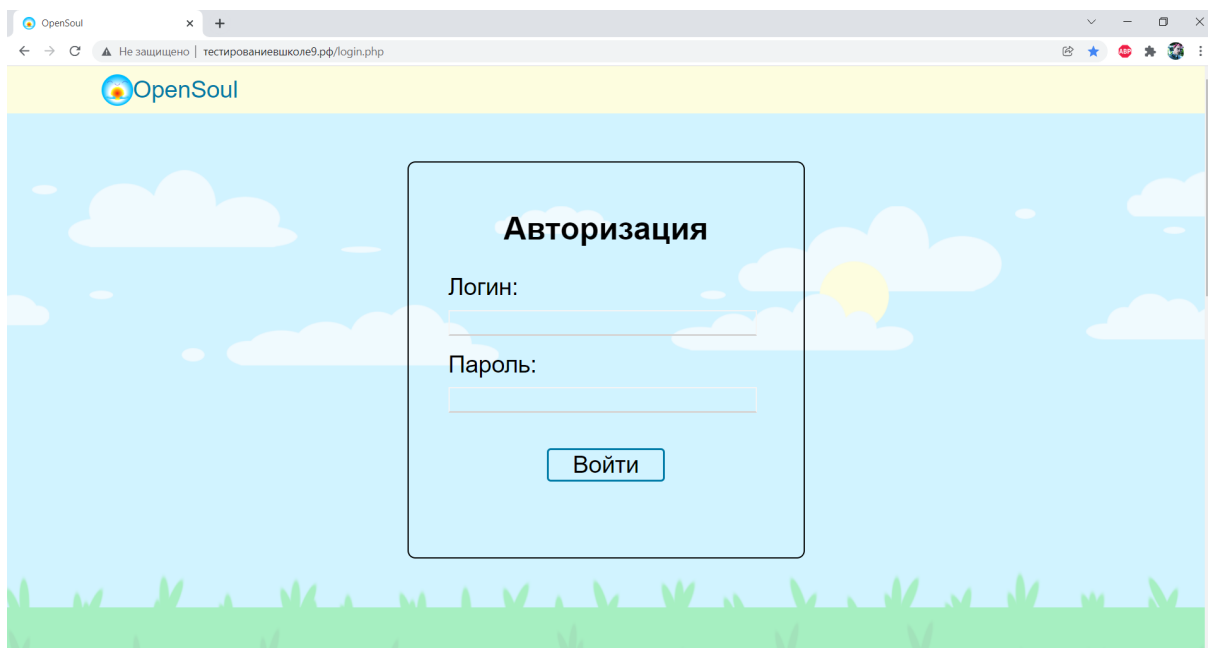
Код формы для авторизации пользователя:

```
<html>

<head>
  <meta charset="utf-8">
  <title>OpenSoul</title>
  <link rel="shortcut icon" href="image/logoo.svg">
  <link rel="stylesheet" href="style.css" type="text/css">
</head>

<body>
  <div class="header">
    <div class="header1">
      <div class="logo">
        
      </div>
      <div class="name">OpenSoul</div>
      <div style="clear:left"></div>
    </div>
  </div>
  <div class="background">
    <div class="probel"> </div>
    <div class="bgenter">
      <div class="nameenter"><b>Авторизация</b></div>
      <form method="POST" action="connect.php">
        <div class="enter">Логин:<br>
          <input type="text" size="30px" name="login"><br>
          Пароль:<br>
          <input type="password" size="30px" name="password"><br>
        </div>
        <div class="entrance"><a href="testing.html"><input type="submit" value="Войти"></a></div>
      </form>
    </div>
  </div>
</body>
</html>
```

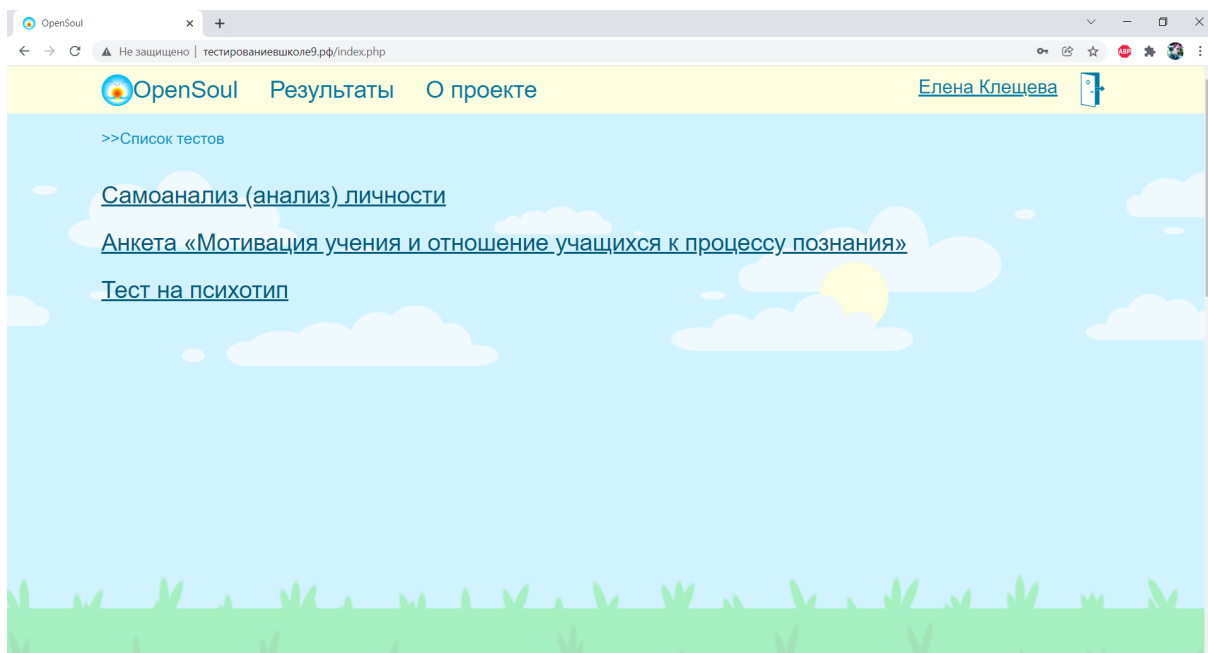
Отображение в браузере:



Код раздела «Тестирование»:

```
<html>
<head>
  <meta charset="utf-8">
  <title>OpenSoul</title>
  <link rel="shortcut icon" href="image/logoo.svg">
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
  <div class="header">
    <div class="header1">
      <div class="logo">
        <a href="index.php"></a>
      </div>
      <div class="name"><a href="index.php">OpenSoul</a></div>
      <div class="results"><a href="results.php">Результаты</a></div>
      <div class="aboutproject"><a href="testing.php">О проекте</a></div>
      <div class="fullname"><a href="results.html"><u>Клещева Елена</u></a></div>
      <div class="exit">
        <a href="login.php"></a>
      </div>
    </div>
  </div>
  <div class="background">
    <div class="bg">
      <div class="idnu">>>Список тестов</div>
      <div class="tests"><a href="test1.html">Тест «Самоанализ (анализ) личности»</a><br>
        <a href="anketa4.html">Анкета «Мотивация учения и отношения учащихся к процессу познания»</a><br>
        <a href="test3.html">Тест на психотип</a><br>
      </div>
    </div>
  </div>
</body>
</html>
```

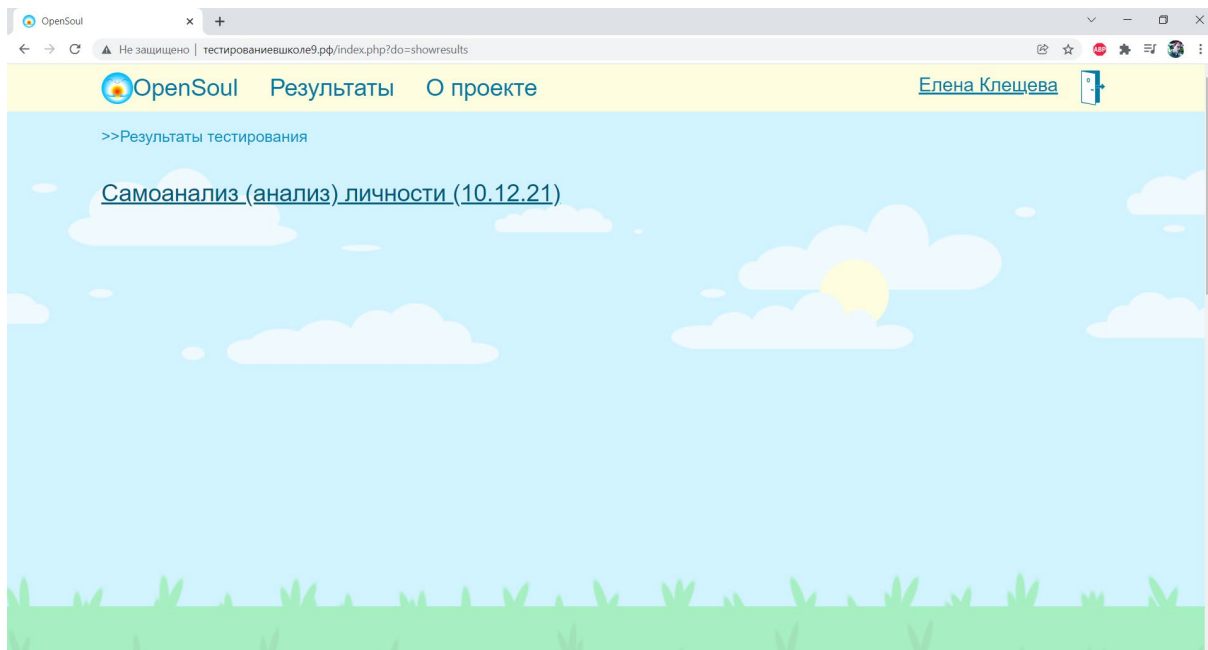
Отображение в браузере:



Код раздела «Результаты»:

```
<html>
<head>
  <meta charset="utf-8">
  <title>OpenSoul</title>
  <link rel="shortcut icon" href="image/logoo.svg">
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
  <div class="header">
    <div class="header1">
      <div class="logo">
        <a href="about.html"></a>
      </div>
      <div class="name"><a href="about.html">OpenSoul</a></div>
      <div class="testing"><a href="testing.html">Тестирование</a></div>
      <div class="results"><a href="results.html">Результаты</a></div>
      <div class="fullname"><a href="results.html"><u>Клещева Елена</u></a></div>
      <div class="exit">
        <a href="index.php"></a>
      </div>
    </div>
  </div>
  <div class="background">
    <div class="bg">
      <div class="idnu">>>Результаты тестирования</div>
      <div class="tests"><a href="result1.html">Самоанализ (анализ) личности(10.12.21)</a><br></div>
    </div>
  </div>
</body>
</html>
```

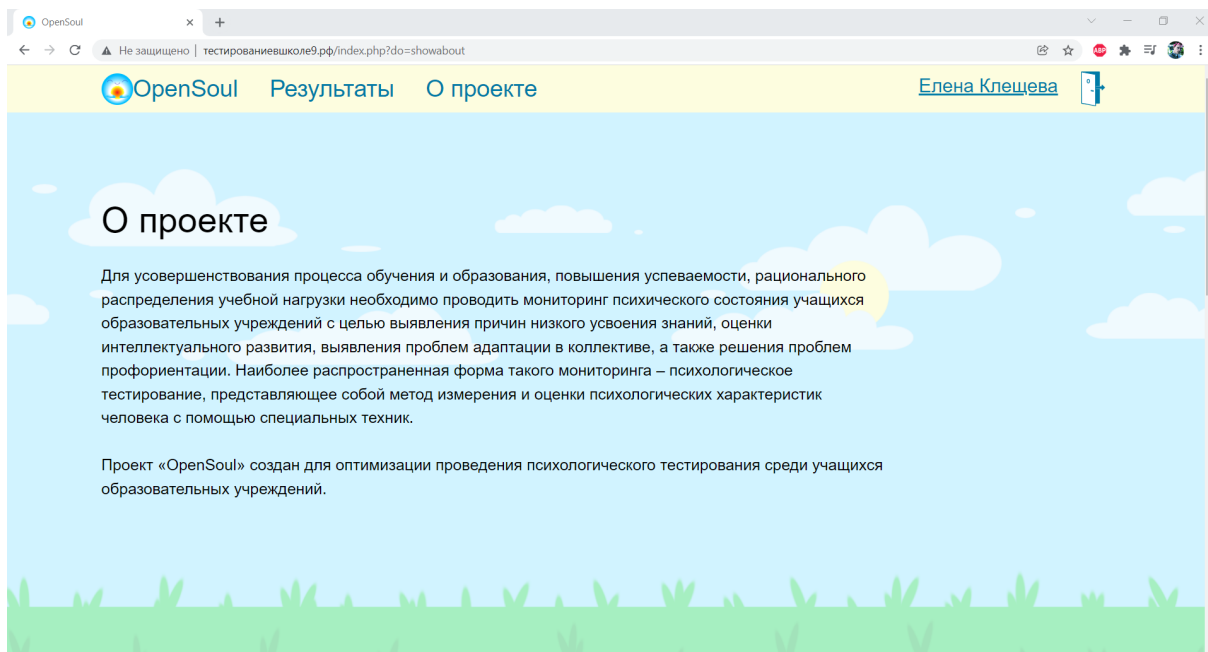
Отображение в браузере:



Код раздела «О проекте»:

```
<html>
<head>
  <meta charset="utf-8">
  <title>О проекте</title>
  <link rel="shortcut icon" href="incudes/templates/image/logoo.svg">
  <link rel="stylesheet" href="incudes/templates/style.css" type="text/css">
</head>
<body>
  <div class="header">
    <div class="header1">
      <div class="logo">
        <a href="index.php"></a>
      </div>
      <div class="name"><a href="index.php">OpenSoul</a></div>
      <div class="results"><a href="index.php?do=showresults">Результаты</a></div>
      <div class="aboutproject"><a href="about.php">О проекте</a></div>
      <div class="fullname"><a href="index.php?do=showresults"><u><?php echo $session[0]['name'] . ' ' . $session[0]['lastname']; ?></u></a></div>
      <div class="exit">
        <a href="login.php"></a>
      </div>
    </div>
  </div>
  <div class="background">
    <div class="bg">
      <div class="nameabout">О проекте</div>
      <div class="about">Для усовершенствования процесса обучения и образования, повышения успеваемости, рационального распределения учебной нагрузки необходимо проводить мониторинг психического состояния учащихся образовательных учреждений с целью выявления причин низкого усвоения знаний, оценки интеллектуального развития, выявления проблем адаптации в коллективе, а также решения проблем профориентации. Наиболее распространенная форма такого мониторинга – психологическое тестирование, представляющее собой метод измерения и оценки психологических характеристик человека с помощью специальных техник.<br>Проект «OpenSoul» создан для оптимизации проведения психологического тестирования среди учащихся образовательных учреждений.</div>
    </div>
  </div>
</body>
</html>
```

Отображение в браузере:



2.6 Домен и хостинг

Что такое домен и хостинг сайта?

Домен и хостинг — это базовые услуги, без которых невозможно создать ни один сайт в Интернете. Чем отличается домен от хостинга? Домен — онлайн-адрес сайта, а хостинг для сайта — это пространство, на котором он «живет». Чтобы создать сайт, сначала надо зарегистрировать домен, а затем заказать услугу хостинга.

Любой сайт — это по факту набор файлов (тексты, изображения, видео, программные скрипты, база данных и т. д.). Но для чего нужен хостинг для сайта? Чтобы

пользователи Интернета вообще смогли увидеть его. Недостаточно иметь файлы сайта на своем ПК, их нужно выложить в сеть.

Выкладываются файлы сайтов на компьютеры со специальными ПО. Они называются серверами. Серверы находятся в сети 24 часа в сутки, обеспечивают непрерывную работу сайта и защищают его от вредоносных атак. Чтобы разместить файлы сайта, нужно арендовать место на сервере или целый сервер, если сайт содержит много контента и нацелен на широкий поток посетителей. Таким образом, хостинг — это, простыми словами, аренда мощностей и места.

Для регистрации доменного имени и хостинга был выбран Masterhost — российский хостинг-провайдер.

Выбор доменного имени для сайта является очень ответственной задачей. Он должен отражать суть сайта и легко запоминаться. И поэтому, тщательно обдумав все варианты, я сделала выбор в пользу «тестированиевшколе9.рф».

2.7 Основы PHP-программирования

PHP — один из самых распространенных языков web-разработки. Он применяется для создания сайтов и web-приложений любой сложности: от лендингов и блогов до интернет-магазинов и браузерных игр.

Его преимущества — в широких возможностях и защищенности благодаря закрытому исходному коду. К минусам можно отнести то, что без сервера с интерпретатором код работать не будет (исправлено в новых версиях языка).

Пример:

Для начала работы создается страница с расширением .php или .html (в зависимости от конфигурации сервера). Исходный код этой страницы содержит теги разметки html и непосредственно PHP-команды. Они заключаются между операторами <?PHP (допустимо писать <?) и ?>. С помощью «\$» обозначается переменная, а команда echo используется для вывода данных на экран. Вот как выглядит подобный код:

```
1 ▼ <?php
2     $username = "Петя"; //Переменная с именем пользователя
3     ?>
4 ▼ <div>
5     <p>Здравствуйте, <?php echo $username; ?>!</p>
6 </div>
```

Открыв эту страницу, пользователь увидит надпись: «Здравствуйте, Петя!».

В PHP поддерживаются применение функций, математические вычисления, работа с переменными (как в примере выше) и массивами, прописывание условий «если — то», создание объектов и многое другое. Чаще всего это используется:

- для отправки форм;
- работы с базами данных;
- создания динамичных страниц;
- использование сессий и cookies;
- загрузки и обработки файлов;
- создания изображений;
- парсинга (процесс автоматического сбора данных и их структурирования).

2.8 База Данных

База данных (БД) — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов, формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

SQL — это язык программирования, используемый в большинстве реляционных баз данных для запросов, обработки и определения данных, а также контроля доступа. SQL был разработан в IBM в 1970-х годах. Со временем у стандарта SQL ANSI появились многочисленные расширения, разработанные такими компаниями как IBM, Oracle и Microsoft. Хотя в настоящее время SQL все еще широко используется, начали появляться новые языки программирования запросов.

В чем заключается различие между базой данных и электронной таблицей?

Базы данных и электронные таблицы (в частности, Microsoft Excel) предоставляют удобные способы хранения информации. Основные различия между ними заключаются в следующем.

- Способ хранения и обработки данных
- Полномочия доступа к данным

- Объем хранения данных

Электронные таблицы изначально разрабатывались для одного пользователя, и их свойства отражают это. Они отлично подходят для одного пользователя или небольшого числа пользователей, которым не нужно производить сложные операции с данными. С другой стороны, базы данных предназначены для хранения гораздо больших наборов упорядоченной информации—иногда огромных объемов. Базы данных дают возможность множеству пользователей в одно и то же время быстро и безопасно получать доступ к данным и запрашивать их, используя развитую логику и язык запросов.

Для базы данных обычно требуется комплексное программное обеспечение, которое называется системой управления базами данных (СУБД). СУБД служит интерфейсом между базой данных и пользователями или программами, предоставляя пользователям возможность получать и обновлять информацию, а также управлять ее упорядочением и оптимизацией. СУБД обеспечивает контроль и управление данными, позволяя выполнять различные административные операции, такие как мониторинг производительности, настройка, а также резервное копирование и восстановление.

В качестве примеров популярного программного обеспечения для управления базами данных, или СУБД, можно назвать MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, СУБД Oracle Database и dBASE.

Для управления базами данных я использовала MySQL.

MySQL — это реляционная система управления базами данных с открытым исходным кодом на основе языка SQL. Она была разработана и оптимизирована для веб-приложений и может работать на многих платформах. Она обладает всеми возможностями, которые требуются веб-разработчикам. База данных MySQL предназначена для обработки миллионов запросов и тысяч транзакций, поэтому ее часто выбирают компании электронной коммерции, которым требуется управлять большим количеством денежных переводов. Гибкость по мере необходимости — основная характеристика MySQL.

Многие ведущие веб-сайты и веб-приложения используют СУБД MySQL, в том числе Airbnb, Uber, LinkedIn, Facebook, Twitter и YouTube.

2.9 Программирование сайта с использованием PHP и БД

Разберем наглядный пример, как можно использовать PHP и БД.

С помощью HTML нельзя просто так получить информацию, введенную пользователем. То есть можно создать красивую форму, но без PHP или другого подобного языка все данные просто исчезнут, когда он нажмет кнопку «Отправить».

Для начала создается форма подобного вида:

```
<form action="/auth.php" method="post"> <!--форма-->
  <input type="text" name="login" placeholder="Логин"><br> <!--Поле ввода логина-->
  <input type="password" name="password" placeholder="Пароль"><br> <!--Поле ввода пароля-->
  <input type="submit" value="Войти"> <!--Кнопка отправки данных-->
</form>
```

У тега `<form>` есть два атрибута: `action` и `method`. В первом указывается страница, куда будут отправлены данные, а во втором — метод передачи. Всего их два:

1. POST — безопасный, но медленный метод. Им пользуются для передачи конфиденциальной информации: пароли, логины, адреса и так далее.
2. GET — простой, но уязвимый метод. Вы могли видеть ссылки вида `index.php?page=8&category=2`. Информация записывается прямо в ссылку, и любой может поменять ее значение или поделить ею, поэтому лучше не передавать так ничего секретного и позаботиться о дополнительной безопасности.

Из формы на страницу `auth.php` будут переданы переменные `login` и `password`. В зависимости от метода они попадут в супермассив `$_POST` или `$_GET`. Дальше на этой странице можно будет сделать с ними все что угодно, в том числе и проверить их наличие в базе данных.

К сайту на PHP подключается база данных. Это во многом упрощает разработку, потому что вся информация будет храниться в таблицах, а не в самом коде. Так быстрее редактировать и добавлять данные на сайт.

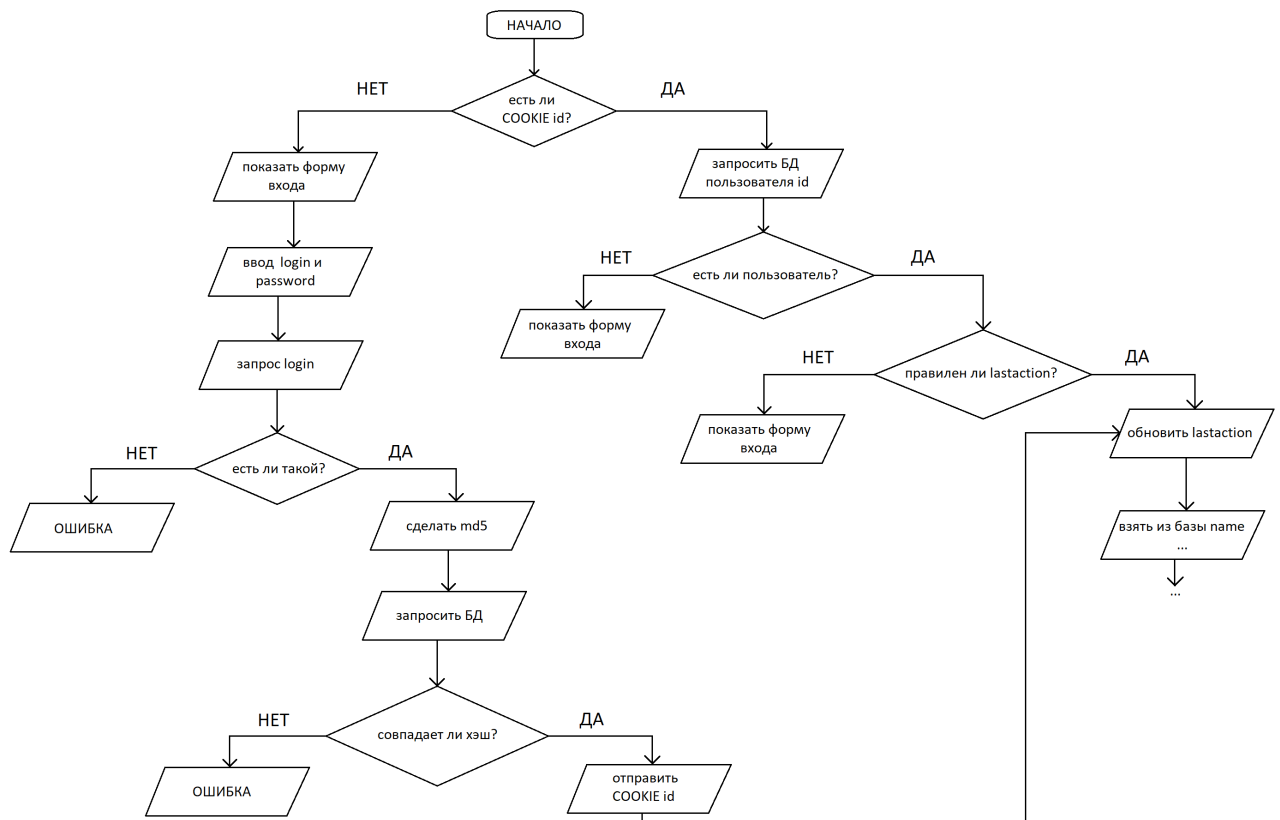
Например, сведения о пользователе. Если бы не базы данных, пришлось бы хранить все авторизационные данные в коде, а это не лучший вариант, потому что тогда загрузка длилась бы вечно.

Продолжим пример с авторизацией. Страница `auth.php` получает логин и пароль. Дальше она может сделать запрос в базу данных, чтобы узнать, существует ли такая комбинация.

```
<?php
//Получаем данные из супермассива $_POST в переменные для удобства
$login = $_POST['login'];
$password = $_POST['password'];
//делаем запрос в базу данных, чтобы проверить наличие пользователя с таким логином и паролем
$queryResult = mysql_query("SELECT * FROM userList WHERE login='$login' AND password='$password'");
//Выполняем проверку
if(mysql_num_rows($queryResult) == 1) {
    //Если есть совпадение, то выводим надпись об успешности операции
    echo "Вы успешно вошли!";
} else {
    //Иначе пишем, что комбинация неверна
    echo "Неверный логин или пароль!";
}
?>
```

Разумеется, это очень упрощенная версия кода, которую вряд ли можно назвать защищенной. На самом деле нужно выполнить множество проверок, чтобы исключить вероятность SQL-инъекции — это когда пользователь пытается ввести некорректную информацию, чтобы получить доступ к базе данных.

Теперь понимая, как работают PHP и БД, можно составить алгоритм авторизации для сайта:



А далее написать и сам код для авторизации пользователей:

```
1 <?php
2 $timeout = 86400;
3 if(!isset($_POST['login'])){
4     if(isset($_COOKIE['id']) == true){
5         $querystring = "SELECT lastaction FROM students WHERE id = " . $_COOKIE['id'];
6         $connect = db_connect();
7         $result = mysqli_query($connect, $querystring);
8         $res = mysqli_fetch_array($result);
9         if($res[0] == true){
10             if(time() - $res[0] <= $timeout){
11                 $querystring = "UPDATE students SET(lastaction=" . time() . ") WHERE id = " . $_COOKIE['id'];
12                 $connect = db_connect();
13                 $result = mysqli_query($connect, $querystring);
14                 $res = mysqli_fetch_array($result);
15                 $querystring = "SELECT * FROM students WHERE id = " . $_COOKIE['id'];
16                 $connect = db_connect();
17                 $result = mysqli_query($connect, $querystring);
18                 $res = mysqli_fetch_array($result);
19             }
20             include('index.php');
21         }
22     }
23 } else {
24     $login = $_POST['login'];
25     $pass = $_POST['password'];
26     $querystring = "SELECT lastaction, password, login, salt, name, lastname FROM students WHERE login = " . $login . " ";
27     $connect = db_connect();
28     $result = mysqli_query($connect, $querystring);
29     $res = mysqli_fetch_array($result);
30     $id = $res['id'];
31     if($res){
32         $passw = getpass($pass, $res['salt']);
33         if($passw == $res['password']){
34             setcookie('id', $id, 99999);
35             include('testing.html');
36         } else {
37             include('message.html');
38         }
39     } else {
40         include('message2.html');
41     }
42 }
43 ?>
```

Таким образом, мы разобрались в том, как работают PHP и БД. Теперь с полученными знаниями можно приступить к самому программированию автоматической системы психологического тестирования.

2.10 Программирование автоматической системы психологического тестирования

Программирование автоматической системы психологического тестирования начинается со структурирования файлов для загрузки в FTP клиент.

Что такое FTP?

FTP является сокращением от англ. File Transfer Protocol — протокол передачи файлов, который применяется для обмена файлами по TCP/IP сетям между двумя компьютерами (клиент и сервер). Протоколу передачи файлов больше 40 лет, он был разработан прежде чем появился TCP/IP и тем более HTTP, однако он до сих пор актуален и используется для подключения к удаленным серверам и обмена файлами.

Данный протокол применяет различные сетевые соединения для передачи команд и файлов между клиентом и сервером. FTP сервер, представляет собой компьютер с установленным на него специальным программным обеспечением и ожидающим внешнего подключения от других компьютеров.

FTP клиент является программой, которая делает попытку соединится с серверным компьютером, как правило к порту номер 21. После успешного подключения к FTP серверу, можно совершать всевозможные операции над данными

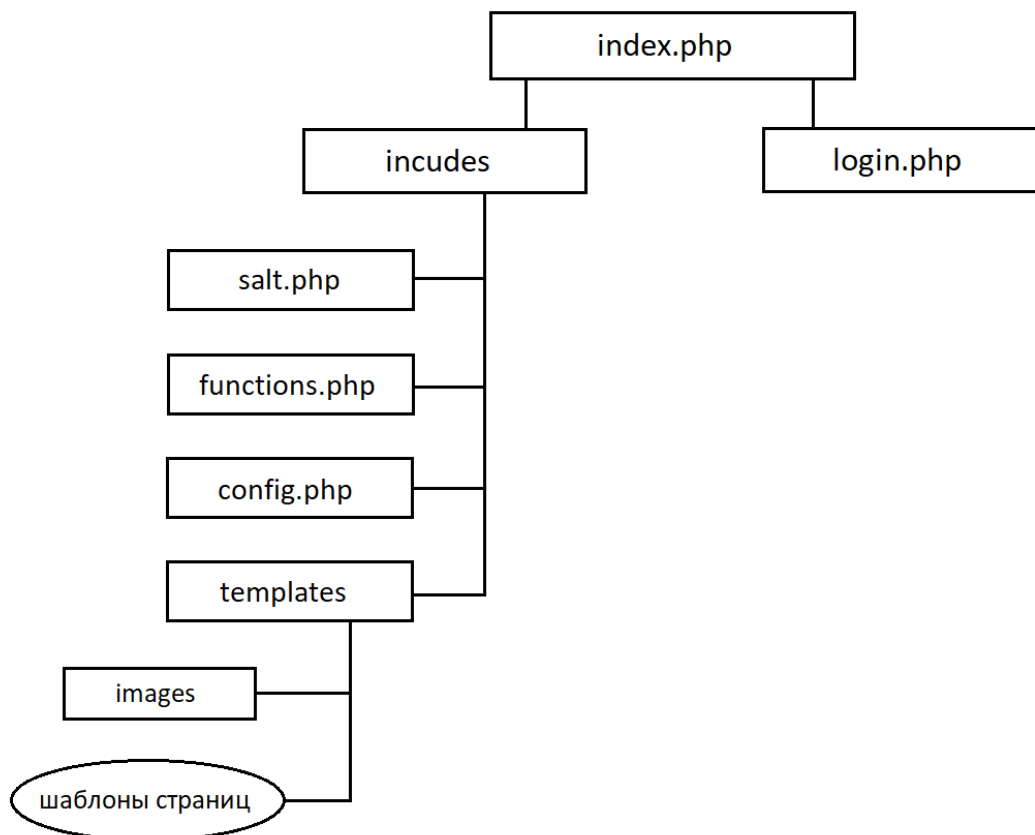
располагающимися на нем, в частности, просмотреть содержимое каталогов, загружать и скачивать файлы с FTP сервера, переименовывать, назначать права доступа, удалять файлы с сервера и прочее.

Соединяясь с FTP сервером допустимо пройти авторизацию предоставляя данные для входа без использования шифрования, а также можно подключиться анонимно, если это позволяет FTP сервер. Для защищенной передачи, которая скрывает (шифрует) логин, пароль и данные, FTP соединение может быть зашифровано при помощи SSL/TLS (FTPS). SSH протокол передачи файлов (SFTP) иногда также используется, но между ним и FTPS есть существенная разница.

Основное назначение FTP протокола - это загрузка файлов и их скачивание с удаленного сервера. Для передачи файлов в пассивном режиме иницируется соединение FTP клиентом из обусловленного диапазона портов к порту сервера. В активном режиме FTP сервер подключается к клиенту из порта 20 к определенному порту, который сообщил ему клиент. Основное различие между данными режимами, состоит в том, с какой стороны открывается соединение для передачи файлов.

Для работы с FTP я воспользовалась FTP менеджером FileZilla.

Вернемся к структурированию файлов. Вот так выглядит структура сайта:



Рассмотрим каждый файл подробнее:

index.php – вершина платформы. Здесь происходит вся основная навигация пользователя по сайту:

```
1 <?php
2 require_once('includes/functions.php');
3 db_connect();
4 $session = check_session();
5 if($session != false){
6     $available_tests = $session[0]['availabletests'];
7     $tests = "SELECT idtest, nametest FROM tests WHERE idtest IN(" . $available_tests . ")";
8     $result = query_select($tests);
9
10    if(isset($_REQUEST['test'])){
11        $tests = "SELECT idtest, nametest FROM tests WHERE idtest = " . $_REQUEST['test'];
12        $result = query_select($tests);
13        $questions = "SELECT * FROM questions WHERE idtest = " . $_REQUEST['test'];
14        $result_2 = query_select($questions);
15
16        include('includes/templates/test.php');
17    }
18    else{
19        if(isset($_REQUEST['do'])){
20            if($_REQUEST['do'] == 'getanswers'){
21                $answers = "INSERT INTO answers VALUES(0, " . $session[0]['id'] . ", " . $_POST['idtest'] . ", " . serialize($_POST) . ", " . time() . ")";
22                query($answers);
23                include('includes/templates/message_ok.php');
24            } else if($_REQUEST['do'] == 'showresults'){
25                if(isset($_REQUEST['rid'])){
26                    $results = "SELECT * FROM answers WHERE idstudent = " . $session[0]['id'];
27                    $test_results = query_select($results);
28
29                    include('includes/templates/results.php');
30                }
31                } else{
32                    $results = "SELECT * FROM answers WHERE idstudent = " . $session[0]['id'] . " AND id = " . $_REQUEST['rid'];
33                    $test_results = query_select($results);
34
35                    include('includes/templates/result.php');
36                }
37            } else if($_REQUEST['do'] == 'showabout'){
38                include('includes/templates/about.php');
39            }
40        } else{
41            include('includes/templates/testing.php');
42        }
43    }
44 }
45 else{
46     $new_url = 'http://тестированиевшколе9.рф/login.php';
47     header('Location: ' . $new_url);
48     exit();
49 }
50
51
52
53
54
55
56
57 ?>
```

login.php – отвечает за авторизацию пользователя:

```
1 <?php
2 if(isset($_REQUEST['do']) && $_REQUEST['do'] == 'logout'){
3     require_once('includes/functions.php');
4     logout();
5     $new_url = 'http://тестированиевшколе9.рф/index.php';
6     header('Location: ' . $new_url);
7     exit();
8 }
9 else{
10
11    if(!isset($_POST['login'])){
12        include('includes/templates/login.php');
13    }
14    else{
15        require_once('includes/functions.php');
16        db_connect();
17        $login = auth();
18        if($login == true){
19            $new_url = 'http://тестированиевшколе9.рф/index.php';
20            header('Location: ' . $new_url);
21            exit();
22        }
23        else{
24            include('includes/templates/message.php');
25        }
26    }
27 }
28
29 ?>
```

functions.php — здесь собраны все основные функции сайта, к которым мы обращаемся с index.php и login.php:

```
1 <?php
2 require_once('config.php');
3 $connection = null;
4
5 function db_connect(){
6     global $connection;
7     global $config;
8     $connect = mysqli_connect($config['server'] . ':' . $config['port'], $config['dbuser'], $config['password']);
9     mysqli_select_db($connect, $config['dbname']);
10
11     mysqli_query($connect, "SET NAMES 'UTF8'");
12
13     $connection = $connect;
14
15     return true;
16 }
17
18 function get_salt($salt_length){
19     $salt = '';
20     for($i = 0; $i < $salt_length; $i++){
21         $salt = $salt . chr(rand(33, 47));
22     }
23     return $salt;
24 }
25
26 function getpass($pass, $salt){
27     return md5(md5($pass . $salt));
28 }
29
30 function query_select($querystring){
31     global $connection;
32     $result = mysqli_query($connection, $querystring);
33
34     if($result == false) return false;
35
36     $res = array();
37     $return_array = array();
38
39     $res = mysqli_fetch_array($result);
40
41     while ($res){
42         $return_array[] = $res;
43         $res = mysqli_fetch_array($result);
44     }
45     return $return_array;
46 }
47
48 function query($querystring){
49     global $connection;
50     $result = mysqli_query($connection, $querystring);
51     if($result){return true;
52     } else return false;
53 }
54
55 function check_session(){
56     global $config;
57
58     if(isset($_COOKIE['id']) == true){
59         $querystring = "SELECT lastaction FROM students WHERE id = " . $_COOKIE['id'];
60
61
62         $result = query_select($querystring);
63
64         if($result == true){
65
66             if(time() - $result[0]['lastaction'] <= $config['cookie_time_out']){
67                 $querystring = "UPDATE students SET lastaction=" . time() . " WHERE id = " . $_COOKIE['id'];
68
69
70                 $result = query($querystring);
71
72                 $querystring = "SELECT * FROM students WHERE id = " . $_COOKIE['id'];
73
74                 $result = query_select($querystring);
75
76                 return $result;
77             } else return false;
78         } else return false;
79     } else return false;
80 }
81
82 function auth(){
83     $login = $_POST['login'];
84     $pass = $_POST['password'];
85     $querystring = "SELECT id, lastaction, password, login, salt, name, lastname FROM students WHERE login = '" . $login . "'";
86
87     $result = query_select($querystring);
88 }
```

```

96
97 ▼      if($result){
98
99          $passw = getpass($pass, $result[0]['salt']);
100 ▼      if($passw == $result[0]['password']){
101          $id = $result[0]['id'];
102          setcookie('id', $id, 999999999);
103
104          $querystring = "UPDATE students SET lastaction=" . time() . " WHERE id =" . $id;
105          echo($querystring);
106          $result = query($querystring);
107          return true;
108 ▼      } else {
109          return false;
110      }
111 ▼      } else {
112          return false;
113      }
114  }
115
116 ▼  function logout(){
117      setcookie('id', '');
118  }
119  ?>

```

salt.php – здесь происходит шифрование пароля для увеличения безопасности данных пользователя:

```

1 ▼ <?php
2     $salt = get_salt(8);
3     echo md5(md5($_REQUEST['p'] . $salt));
4     echo '<br>' . $salt;
5 ▼     function get_salt($salt_length){
6         $salt = '';
7 ▼         for($i = 0; $i < $salt_length; $i++){
8             $salt = $salt . chr(rand(33, 47));
9         }
10        return $salt;
11    }
12    ?>

```

header.php – шаблон, включающий в себя содержание тега <HEAD>:

```

1 <!doctype html>
2 <html>
3
4 ▼ <head>
5     <meta charset="utf-8">
6     <title>OpenSoul</title>
7     <link rel="shortcut icon" href="includes/templates/image/logoo.svg">
8     <link rel="stylesheet" href="includes/templates/style.css" type="text/css">
9 </head>

```

login.php (в папке includes) – шаблон формы входа для пользователей:

```

1 <?php
2     include('includes/templates/header.php');
3     ?>
4 <body>
5 ▼     <div class="header">
6         <div class="header1">
7             <div class="logo">
8                 
9             </div>
10            <div class="name">OpenSoul</div>
11        </div>
12    </div>
13 ▼    <div class="background">
14        <div class="probel"> </div>
15 ▼        <div class="bgender">
16            <div class="nameenter"><b>Авторизация</b></div>
17            <form method="POST" action="login.php">
18                <div class="enter">Логин:<br>
19                    <input type="text" size="30px" name="login"><br>
20                    Пароль:<br>
21                    <input type="password" size="30px" name="password"><br>
22                </div>
23                <div class="entrance"><a href=""><input type="submit" value="Войти"></a></div>
24            </form>
25        </div>
26    </div>
27 </body>
28
29 </html>

```

testing.php – шаблон раздела «Тестирование». С помощью цикла выводит запрашиваемые данные из БД на экран:

```
1 <?php
2     include('incudes/templates/header.php');
3
4 >
5 <body>
6     <div class="header">
7         <div class="header1">
8             <div class="logo">
9                 <a href="index.php"></a>
10            </div>
11            <div class="name"><a href="index.php">OpenSou</a></div>
12            <div class="results"><a href="index.php?do=showresults">Результаты</a></div>
13            <div class="aboutproject"><a href="index.php?do=showabout">о проекте</a></div>
14            <div class="fullname"><a href="index.php?do=showresults"><u><?php echo $session[0]['name'] . ' ' . $session[0]['lastname']; ?></u></a></div>
15            <div class="exit">
16                <a href="/login.php?do=logout"></a>
17            </div>
18        </div>
19    </div>
20    <div class="background">
21        <div class="bg">
22            <div class="idnu">>>Список тестов</div>
23            <div class="tests">
24                <?php
25                    for($i = 0; $i < count($result); $i++):
26
27                    >
28                        <a href="index.php?test=<?php echo $result[$i]['idtest']; ?>"><?php echo $result[$i]['nametest']; ?></a><br>
29                    <?php endfor; ?>
30            </div>
31        </div>
32    </div>
33 </body>
34
35 </html>
```

test.php – шаблон теста. С помощью цикла выводит запрашиваемые данные из БД на экран:

```
1 <!doctype html>
2 <html>
3
4 <head>
5     <meta charset="utf-8">
6     <title><?php echo $result[0]['nametest']; ?></title>
7     <link rel="shortcut icon" href="incudes/templates/image/logoo.svg">
8     <link rel="stylesheet" href="incudes/templates/style.css" type="text/css">
9     <script type="text/javascript">
10         function myOnload(){
11             var Okno = document.getElementById('instruction_button');
12             Okno.onclick = function(e){
13                 if(document.getElementById('instruction').style.display == 'none'){
14                     document.getElementById('instruction').style.display = 'block';
15                 } else {
16                     document.getElementById('instruction').style.display = 'none';
17                 }
18                 e.preventDefault();
19             }
20
21             var OknoClose = document.getElementById('instruction_button_close');
22             OknoClose.onclick = function(e){
23                 document.getElementById('instruction').style.display = 'none';
24                 e.preventDefault();
25             }
26
27             var StartOknoClose = document.getElementById('instruction_start_button_close');
28             StartOknoClose.onclick = function(e){
29                 document.getElementById('instruction_start').style.display = 'none';
30                 e.preventDefault();
31             }
32
33             document.getElementById('test_block_1').style.display = 'block';
34
35             <?php
36                 if(count($result_2) % 6 == 0){
37                     $cycles = count($result_2) / 6;
38                 }else{
39                     $cycles = count($result_2) / 6 + 1;
40                 }
41                 $p = 1;
42                 for($p; $p < $cycles; $p++):
43                     >
44                         var show_test_block_<?php echo $p + 1;?> = document.getElementById('continue_<?php echo $p;?>');
45                         show_test_block_<?php echo $p + 1;?>.onclick = function(e){
46                             document.getElementById('test_block_<?php echo $p;?>').style.display = 'none';
47                             document.getElementById('test_block_<?php echo $p + 1;?>').style.display = 'block';
48                         }
49                 }
50
51             <?php endfor;
52             if($p == $cycles):
53                 >
54                     var finish_button = document.getElementById('continue_<?php echo $p;?>');
55                     finish_button.value = "завершить тест";
56                     finish_button.onclick = function(){
57                         document.forms[0].submit();
58                         return false;
59                     }
60                 <?php endif;?>
61             }
62         </script>
63     </head>
64
65 </html>
```

```

66 <body onload="myOnLoad();">
67 <div class="header">
68 <div class="header1">
69 <div class="logo">
70 <a href="index.php"></a>
71 </div>
72 <div class="name"><a href="index.php">OpenSoul</a></div>
73 <div class="results"><a href="index.php?do=showresults">Результаты</a></div>
74 <div class="aboutproject"><a href="index.php?do=showabout">О проекте</a></div>
75 <div class="fullname"><a href="index.php?do=showresults"><u><?php echo $session[0]['name'] . ' ' . $session[0]['lastname']; ?></u></a></div>
76 <div class="exit">
77 <a href="login.php"></a>
78 </div>
79 </div>
80 </div>
81 <div class="background">
82 <div class="bg">
83 <div class="nametest"><?php echo $result[0]['nametest']; ?> <a href="" id="instruction_button"></a>
84 </div>
85 <div class="test">
86 <?php
87 if(count($result_2) % 6 == 0){
88 $the_number_of_cycles = count($result_2) / 6;
89 }else{
90 $the_number_of_cycles = count($result_2) / 6 + 1;
91 }
92
93 $j = 1;
94 $q = 0;
95 $q_limit_block = 6;
96
97 for ($i = 1; $i <= $the_number_of_cycles; $i++){
98
99
100
101 <div class="test_block" id="test_block_<?php echo $i;?>" style="display: none">
102 <div class="numberq"><?php if(count($result_2) - 6 * $j >= 0){echo(6 * $j); $j++;} else echo(count($result_2)); ?> из <?php
103 echo count($result_2); ?> вопросов</div>
104
105 <?php
106 for ($q; $q < $q_limit_block; $q++){
107
108 <?php echo $result_2[$q]['question']; ?><br>
109
110 <div class="variants">
111 <?php
112 $variants = "SELECT * FROM variants WHERE idquestion = ".$result_2[$q]['idquestion'];
113 $result_3 = query_select($variants);
114
115 for($v = 0; $v < count($result_3); $v++){
116
117 <input names="<?php echo $q;?>" type="<?php echo $result_2[$q]['type'];?>" value="<?php echo $result_3[$v]
118 ['value'];?>" required"><?php echo $result_3[$v]['variant'];?>
119
120 <?php endfor; ?>
121 <br><br>
122 </div>
123
124 <?php endfor; ?>
125 <?php if(count($result_2) - 6 * $j >= 0){$q_limit_block = 6 * $j;} else $q_limit_block = count($result_2); ?>
126
127 <div class="continue">
128 <input id="continue_<?php echo $i;?>" type="button" value="Продолжить >>">
129 </div>
130 </div>
131 <?php endfor; ?>
132
133 </div>
134 <input type="hidden" name="idtest" value="<?php echo $result[0]['idtest']; ?>">
135 </form>
136 </div>
137 </div>
138 <div class="vsploknokno" id="instruction_start" style="display: block">
139 <div class="probel"></div>
140 <div class="okno">
141 <div class="tinstruction">Инструкция по выполнению теста «Самоанализ (анализ) личности:</div><br>
142 <div class="txtinstruction">
143 <span class="marker">></span>Тест позволяет оценивать уровень проявления социально-ценных качеств личности:<br>
144 - активность нравственной позиции;<br>
145 - коллективизм;<br>
146 - гражданственность в труде;<br>
147 - трудолюбие;<br>
148 - волевые качества.<br><br>
149 <span class="marker">></span>На каждый вопрос даётся пять вариантов ответа:<br>
150 <span class="between">___</span>о всегда<span class="between">___</span>о почти всегда<span class="between">___</span>о иногда<span
151 class="between">___</span>о очень редко<span class="between">___</span>о никогда<br><br>
152 <span class="marker">></span>Необходимо ответить на каждый вопрос.<br><br>
153 <span class="marker">></span>Создайтесь отвечать быстро и честно.<br><br>
154 <span class="marker">></span>Ознакомьтесь с результатом тестирования можно, перейдя во вкладку <span class="marker">Результаты</span> или нажав
155 на <span class="marker">></span>свое имя</u>.</span><br><br>
156 <span class="marker">></span>Если Вам понадобится обратиться к инструкции во время прохождения теста, то нажмите на значок , который находится справа от наименования теста.<br>
158 </div>
159 <button id="instruction_start_button_close">Начать тест</button>
160 </div>
161 </div>
162 <div class="vsploknokno" id="instruction" style="display: none">
163 <div class="probel"></div>
164 <div class="okno">
165 <div class="tinstruction">Инструкция по выполнению теста «Самоанализ (анализ) личности:</div><br>
166 <div class="txtinstruction">
167 <span class="marker">></span>Тест позволяет оценивать уровень проявления социально-ценных качеств личности:<br>
168 - активность нравственной позиции;<br>
169 - коллективизм;<br>
170 - гражданственность в труде;<br>
171 - трудолюбие;<br>
172 - волевые качества.<br><br>
173 <span class="marker">></span>На каждый вопрос даётся пять вариантов ответа:<br>
174 <span class="between">___</span>о всегда<span class="between">___</span>о почти всегда<span class="between">___</span>о иногда<span
175 class="between">___</span>о очень редко<span class="between">___</span>о никогда<br><br>
176 <span class="marker">></span>Необходимо ответить на каждый вопрос.<br><br>
177 <span class="marker">></span>Создайтесь отвечать быстро и честно.<br><br>
178 <span class="marker">></span>Ознакомьтесь с результатом тестирования можно, перейдя во вкладку <span class="marker">Результаты</span> или нажав
179 на <span class="marker">></span>свое имя</u>.</span><br><br>
180 <span class="marker">></span>Если Вам понадобится обратиться к инструкции во время прохождения теста, то нажмите на значок , который находится справа от наименования теста.<br>
182 </div>
183 <button id="instruction_button_close">Продолжить</button>
184 </div>
185 </div>
186 </body>
187 </html>

```

results.php – шаблон раздела «Результаты». С помощью цикла выводит запрашиваемые данные из БД на экран:

```
1 <?php
2     include('includes/templates/header.php');
3
4 >
5 <body>
6 <div class="header">
7 <div class="header1">
8 <div class="logo">
9     <a href="index.php"></a>
10 </div>
11 <div class="name"><a href="index.php">OpenSoul</a></div>
12 <div class="results"><a href="index.php?do=showresults">Результаты</a></div>
13 <div class="aboutproject"><a href="index.php?do=showabout">О проекте</a></div>
14 <div class="fullname"><a href="index.php?do=showresults"><u><?php echo $session[0]['name'] . ' ' . $session[0]['lastname']; ?></u></a></div>
15 <div class="exit">
16 <a href="login.php"></a>
17 </div>
18 </div>
19 </div>
20 <div class="background">
21 <div class="bg">
22 <div class="idnu"><div class="tests">
23 <div class="tests">
24 <?php for($i = 0; $i < count($test_results); $i++):
25     $name_test_results = "SELECT nametest FROM tests WHERE idtest = ".$test_results[$i]['idtest'];
26     $name = query_select($name_test_results);
27     $date_test = date("d.m.y",$test_results[$i]['date']);
28 <?php
29     <a href="index.php?do=showresults&rid=<?php echo $test_results[$i]['id']; ?>"><?php echo $name[0]['nametest'], " ", "(", $date_test,
30     ")";?></a><br>
31 <?php endforeach; ?>
32 </div>
33 </div>
34 </body>
35
36 </html>
```

message.php – шаблон сообщения при неудачной попытке пользователя авторизоваться:

```
1 <?php
2     include('includes/templates/header.php');
3
4 >
5 <div class="header">
6 <div class="header1">
7 <div class="logo">
8     
9 </div>
10 <div class="name">OpenSoul</div>
11 <div style="clear:left"></div>
12 </div>
13 </div>
14 <div class="background">
15 <div class="bg">
16 <div class="probel"> </div>
17 <div class="message">Неверно введен логин или пароль.
18 <a href="login.php"><input type="button" value="Повторить попытку"></a></div>
19 </div>
20 </div>
21 </body>
22
23 </html>
```

message_ok.php – шаблон сообщения при удачной попытке отправки результатов тестирования:

```
1 <?php
2     include('includes/templates/header.php');
3
4 >
5 <div class="header">
6 <div class="header1">
7 <div class="logo">
8     
9 </div>
10 <div class="name">OpenSoul</div>
11 </div>
12 </div>
13 <div class="background">
14 <div class="bg">
15 <div class="probel"> </div>
16 <div class="message" style="background-color: rgba(40,255,0,0.45)">Ваш ответ успешно записан.
17 <a href="index.php"><input type="button" value="Понятно"></a></div>
18 </div>
19 </div>
20 </body>
21
22 </html>
```

result.php – шаблон результата. С помощью цикла выводит запрашиваемые данные из БД на экран:

```
1 <?php
2 include('incudes/templates/header.php');
3
4 ?>
5 <body>
6 <div class="header">
7 <div class="header1">
8 <div class="logo">
9 <a href="index.php"></a>
10 </div>
11 <div class="name"><a href="index.php">OpenSoul</a></div>
12 <div class="results"><a href="index.php?do=showresults">Результаты</a></div>
13 <div class="aboutproject"><a href="index.php?do=showabout">О проекте</a></div>
14 <div class="fullname"><a href="index.php?do=showresults"><?php echo $session[0]['name'] . ' ' . $session[0]['lastname']; ?></a></div>
15 <div class="exit">
16 <a href="login.php"></a>
17 </div>
18 </div>
19 </div>
20 <div class="background">
21 <div class="bg">
22
23 <div class="nametest">
24 <?php
25 $name_test_results = "SELECT nametest FROM tests WHERE idtest = ".$test_results[0]['idtest'];
26 $name = query_select($name_test_results);
27 $date_test = date("d.m.y", $test_results[0]['date']);
28 echo $name[0]['nametest'], " ", "(", $date_test, ")";
29
30 ?>
31 </div>
32
33 <div class="result">
34 <?php $answer = unserialize($test_results[0]['answers']);?>
35 <?php
36 function factor_level($temp){
37     if($temp > 4.32){echo "Высокий";}
38     else if($temp < 4.33 && $temp > 3.65){echo "Средний";}
39     else if($temp < 3.66){echo "Низкий";}
40 }
41
42 ?>
43 Уровень проявления социально-ценных качеств личности:<br><br>
44
45 Активность нравственной позиции: <b><?php $temp = ($answer[0] + $answer[1] + $answer[12] + $answer[13]) / 4; factor_level($temp); ?></b><br>
46
47 о уважение к людям, сочувствие: <?php $temp = ($answer[0] + $answer[12]) / 2; factor_level($temp); ?><br>
48 о стремление к нравственному самосовершенствованию: <?php $temp = ($answer[1] + $answer[13]) / 2; factor_level($temp); ?><br><br>
49
50 Коллективизм: <b><?php $temp = ($answer[2] + $answer[3] + $answer[14] + $answer[15]) / 4; factor_level($temp); ?></b><br>
51 о ответственность перед коллективом: <?php $temp = ($answer[2] + $answer[14]) / 2; factor_level($temp); ?><br>
52 о чуждость и взаимопомощь: <?php $temp = ($answer[3] + $answer[15]) / 2; factor_level($temp); ?><br><br>
53
54 Гражданственность в труде: <b><?php $temp = ($answer[4] + $answer[5] + $answer[16] + $answer[17]) / 4; factor_level($temp); ?></b><br>
55 о осознание значимости своего труда для общества: <?php $temp = ($answer[4] + $answer[16]) / 2; factor_level($temp); ?><br>
56 о бережное отношение к результатам труда, к природе: <?php $temp = ($answer[5] + $answer[17]) / 2; factor_level($temp); ?><br><br>
57
58 Трудоспособность: <b><?php $temp = ($answer[6] + $answer[7] + $answer[18] + $answer[19]) / 4; factor_level($temp); ?></b><br>
59 о добросовестность: <?php $temp = ($answer[6] + $answer[18]) / 2; factor_level($temp); ?><br>
60 о самостоятельность в преодолении трудностей: <?php $temp = ($answer[7] + $answer[19]) / 2; factor_level($temp); ?><br><br>
61
62 Творческая активность: <b><?php $temp = ($answer[8] + $answer[9] + $answer[20] + $answer[21]) / 4; factor_level($temp); ?></b><br>
63 о стремление к улучшению процесса работы: <?php $temp = ($answer[8] + $answer[20]) / 2; factor_level($temp); ?><br>
64 о стремление к новому, инициатива: <?php $temp = ($answer[9] + $answer[21]) / 2; factor_level($temp); ?><br><br>
65
66 Волевые качества: <b><?php $temp = ($answer[10] + $answer[11] + $answer[22] + $answer[23]) / 4; factor_level($temp); ?></b><br>
67 о целеустремленность: <?php $temp = ($answer[10] + $answer[22]) / 2; factor_level($temp); ?><br>
68 о настойчивость и самообладание: <?php $temp = ($answer[11] + $answer[23]) / 2; factor_level($temp); ?><br><br>
69
70 <div class="continue">
71 <a href="index.php?do=showresults"><input type="button" value="Понятно"></a>
72 </div>
73
74 </div>
75 </body>
76
77 </html>
```

Таким образом, я рассмотрела все основные этапы проектировки платформы для проведения онлайн-тестирования и разработала веб-приложение для проведения психологических тестирований.

3 Проведение психологического тестирования с использованием разработанной автоматической системы

На одном из уроков информатики в 11 классе было проведено онлайн-тестирование с использованием разработанной мной автоматической системы психологического тестирования.

Учащиеся затратили на знакомство с сайтом, прохождение теста и ознакомление с результатами тестирования менее 10 минут. Сайт произвел на выпускников хорошее впечатление, они остались довольны новым форматом проведения психологического тестирования.

Результаты онлайн-тестирования «Самоанализ (анализ) личности»:



Пояснение (фактор – название фактора – уровень проявления фактора):

- Ф1 – Активность нравственной позиции – Средний
- Ф1.1 – уважение к людям, совестливость – Средний
- Ф1.2 – стремление к нравственному самовоспитанию – Средний
- Ф2 – Коллективизм – Средний
- Ф2.1 – ответственность перед коллективом – Низкий
- Ф2.2 – чуткость и взаимопомощь – Средний
- Ф3 – Гражданственность в труде – Средний
- Ф3.1 – осознание значимости своего труда для общества – Низкий
- Ф3.2 – бережное отношение к результатам труда, к природе – Средний
- Ф4 – Трудолюбие – Средний
- Ф4.1 – добросовестность – Средний
- Ф4.2 – самостоятельность в преодолении трудностей – Средний
- Ф5 – Творческая активность – Средний
- Ф5.1 – стремление к улучшению процесса работы – Низкий
- Ф5.2 – стремление к новому, инициатива – Средний
- Ф6 – Волевые качества – Средний
- Ф6.1 – целеустремлённость – Средний
- Ф6.2 – настойчивость и самообладание – Низкий

Таким образом, автоматическая система психологического тестирования в формате онлайн-тестирования показала свою эффективность и доказала необходимость в автоматизации процесса проведения психологического тестирования в образовательных учреждениях.

Заключение

В результате достижения цели была спроектирована и разработана автоматическая система психологического тестирования школьников в формате онлайн-тестирования на сайте тестированиевшколе9.рф для пользования учащимися образовательного учреждения.

Проведение психологических тестирований позволяет оценивать ситуацию в коллективе и, как следствие, формировать благоприятный психологический климат в группе и способствовать росту заинтересованности школьников в учебе.

Основные результаты проведенной работы:

1. Проведен обзор на тему психологических тестирований.
2. Спроектирована платформа тестирования для школьников.
3. Выполнена программная реализация платформы психологического тестирования для школьников.
4. Описаны результаты проделанной работы. В результате работы было разработано веб-приложение для проведения психологических тестирований.
5. Проведено онлайн-тестирование на сайте тестированиевшколе9.рф среди 11-классников.
6. Проанализированы результаты онлайн-тестирования.

При разработке использовались такие языки программирования как PHP, HTML-и CSS-шаблоны, включая JavaScript – расширения.

Данная платформа была разработана для проведения онлайн-тестирований, с целью оптимизации рабочего времени как классных руководителей, так и учащихся образовательных учреждений, минимизации вычислительных ошибок и недостоверности оценки способностей тестируемого.

Я планирую продолжить работу над данным сайтом, а именно:

- Спроектировать и разработать «Панель администратора»;
- Пополнить БД новыми психологическими тестами;
- Улучшить автоматическую систему вывода тестов из БД;
- Пересмотреть дизайн сайта.

Список источников

1. Методика «Самоанализ (анализ) личности» (методика О.И. Моткова, модифицирована Т. А. Мироновой) [Электронный ресурс] – <https://nsportal.ru/shkola/materialy-k-attestatsii/library/2019/05/10/metodika-samoanaliz-lichnosti>
2. HTML | Введение в веб-разработку [Электронный ресурс] – https://ru.hexlet.io/courses/intro_to_web_development/lessons/html/theory_unit
3. Что такое CSS: объясняем простыми словами [Электронный ресурс] – <https://gb.ru/posts/что-такое-css-obyasnyаем-prostymi-slovami>
4. JavaScript [Электронный ресурс] – <https://blog.ingate.ru/seo-wikipedia/java-script/>
5. Что такое FTP – FileZilla [Электронный ресурс] – <https://filezilla.ru/article/что-такое-ftp>