

ГБОУ Школа “Покровский квартал”

Разработка сайта для определения будущей карьеры

Работу выполнили:

Ученицы 10Т класса

ГБОУ Покровский квартал

Букина Юлия Алексеевна

Лукина Вероника Дмитриевна

Научный руководитель:

Гришина Арина Александровна

Г. Москва 2024

Оглавление

Введение.....	3
Цель проекта:.....	3
Задачи:.....	3
Актуальность:.....	3
Методика выполнения:.....	3
Дизайн:.....	4
Frontend:.....	6
Backend:.....	8
Результат:.....	9
Вывод:.....	10
Список литературы:.....	10

Введение

Цель проекта:

Создать сайт с профориентационным тестом, который, исходя из ответов учеников, будет выявлять их сильные стороны и давать рекомендации по выбору профессии.

Задачи:

1. Поиск и анализ подходящих сайтов для создания теста
2. Создание дизайна
3. Разработка кода (Frontend и Backend) и его объяснение
4. Проверка сайта и подведение итогов

Актуальность:

Значение сайта с профориентационным тестированием в современном мире крайне важно. Ежегодно миллионы выпускников школ сталкиваются с необходимостью выбора профессии, что является одним из важнейших решений их жизни.

Быстро меняющийся рынок труда диктует условия: некоторые профессии теряют актуальность, в то время как другие набирают популярность. Профориентационный тест служит инструментом для определения подходящих профессий на основе индивидуальных интересов и способностей. Данный ресурс особенно ценен для старшеклассников, готовящихся к выпускным экзаменам и выбору будущей сферы деятельности.

Методика выполнения:

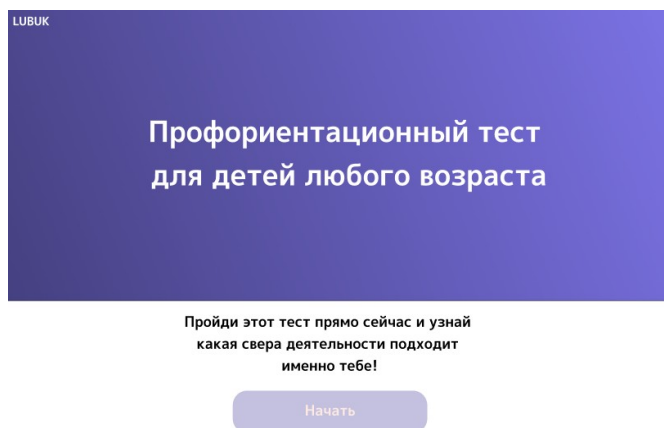
Для начала работы нужно было найти специализированного человека, который разбирается в сфере создания профориентационных тестов. После был начат анализ всех найденных тестов и выявлен самый подходящий для создания опроса на сайте.

Дизайн:

После обработки подходящего теста, реализовывался дизайн. Для начала была изучена платформа для создания дизайна сайта “Figma”. Затем была сделана первая страница сайта, которая включает в себя название теста, название

сайта, кнопку “Начать”, чтобы приступить к прохождению теста, и текст над этой кнопкой, также было сделано всплывающее окно, которое предупреждает о начале прохождения теста после нажатия кнопки “Начать” .

Начальный дизайн из “Figma”:



После главной страницы был сделан дизайн для второй страницы сайта, в которой содержится ввод имени, фамилии и класса пользователя.

Начальный дизайн из “Figma”:



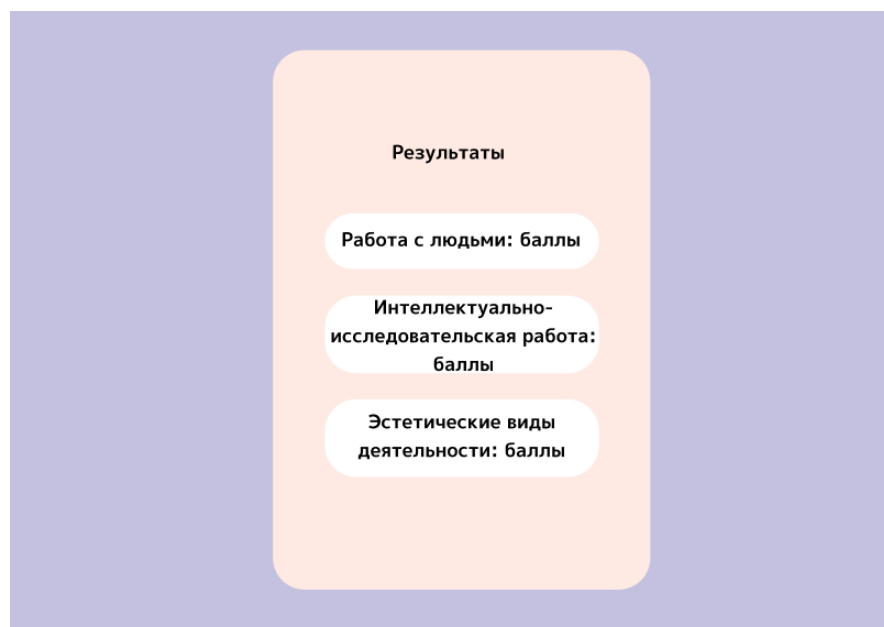
Потом был оформлен дизайн для страниц с вопросами теста, который включает в себя номер теста, сам вопрос и три варианта ответа.

Начальный дизайн одной из страниц из “Figma”:



На завершающей странице сайта будут выведены результаты пройденного теста, где будут отображаться набранные баллы по определённой категории.

Пример дизайна из “Figma”:



Frontend:

После создания дизайна началась разработка кода страниц. Для начала была сделана Frontend-часть – это то, что видит пользователь на сайте, иными словами, это оформление всех страниц. Для её создания использовались языки программирования такие как: HTML и CSS.

Пример кода основной страницы с вопросами Frontend-части:

```
body {  
  margin: 0;  
  font-family: 'Mplus 1p', sans-serif;  
  background-color: #c4c1e3;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}  
.team-work-preference-container1 {  
  display: flex;  
  justify-content: center;  
  padding: 20px;  
}
```

В этой части написан код для основной страницы на языке CSS, например блок `body` делает общий вид всей страницы: убирает отступы, подбирает шрифт и цвет фона, а также центрирует всё содержимое. Блок `team-preference-contanier1` помогает расположить все элементы внутри себя по центру и добавляет пространство вокруг них.

```
.teamwork-preference-container2 {  
  width: 100%;  
  max-width: 600px;  
  background-color: #feeae3;  
  border-radius: 20px;  
  padding: 25px;  
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);  
}  
.header-title, .team-work-preference-heading {  
  text-align: center;  
  margin-bottom: 20px;  
}  
.team-preference-container3 {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}  
.team-member-card {  
  background-color: white;  
  border-radius: 15px;  
  padding: 15px;  
  margin: 10px 0;  
  width: 100%;
```

Здесь создаётся окно светло-розового цвета, в котором будут храниться вопрос и варианты ответов. Также благодаря созданным блокам текст будет находиться в центре страницы, а все элементы будут расположены друг под другом.

```
width: 100%;
max-width: 400px;
text-align: center;
}

.my-button {
background-color: #d9d2e9;
color: black;
border: none;
border-radius: 8px;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
cursor: pointer;
transition: background-color 0.3s ease;
}

.my-button:hover {
background-color: #c6c0d0;
}
</style>
```

В этой части кода создаётся кнопка, по которой пользователи будут переходить на следующую страницу с вопросом, нажимая по ней. Также добавлена функция изменения цвета кнопки при наведении на неё курсора.

```
<body>
  <div class="team-work-preference-container1">
    <div class="teamwork-preference-container2">
      <p class="header-title">Вопрос 14 из 15</p>
      <p class="team-work-preference-heading">{{ question }}</p>
      <form method="post">
        {{csrf_token}}
        <div class="team-preference-container3">
          <div class="team-member-card">
            <input type="radio" name="answer" value="A" required>Заниматься сбытом товаров<br>
          </div>
          <div class="team-member-card">
            <input type="radio" name="answer" value="B" required>Изготавливать изделия<br>
          </div>
          <div class="team-member-card">
            <input type="radio" name="answer" value="C" required>Организовывать процесс производства товаров<br>
          </div>
          <div>
            <a href="page15" button type="submit" class="my-button">Выбрать</button></a>
          </div>
        </div>
      </form>
    </div>
  </div>
</body>
```

В данном фрагменте код написан на языке HTML, здесь создаются заголовки, подзаголовки, пишется текст с номером вопроса, сам вопрос и его варианты ответа. Также сюда добавляется кнопка “Выбрать”, которая отвечает за переход к следующему вопросу.

Следующие страницы были сделаны по аналогии с приведенным примером.

Backend:

Следующим этапом было написание Backend-части, для её создания использовались языки Python и Django, а также сам сайт создавался в VSCode. Данная часть отвечает за логику и функционал. Для начала работы нужно было продумать логику сайта. Её суть заключается в том, что исходя из ответов пользователя, будет выдаваться сфера деятельности, которая больше всего будет подходить человеку. Для этого каждый вариант ответа относился к определённой категории, заданная в коде. В течение теста человек отвечает на вопросы. Когда он выбирает вариант ответа одной из категорий, то в эту категорию засчитываются баллы, которые будут выведены на последней странице сайта. Исходя из количества баллов, набранные в разных категориях, человек увидит сферу деятельности с наибольшим количеством баллов и таким образом сможет определиться со своей будущей карьерой.

Код:

```
results = {'A': 0, 'B': 0, 'C': 0}

def initialize_results(request):
    if 'results' not in request.session:
        request.session['results'] = {'A': 0, 'B': 0, 'C': 0}

def update_results(request, answer):
    if answer in request.session['results']:
        request.session['results'][answer] += 1
```

В этой части кода задаётся словарь “results” для хранения результатов теста. После него идет блок под названием “initialize_results”, благодаря нему

подготавливаются результаты теста. И самый последний блок на изображении “update_results” отвечает за обновление результатов теста, то есть эта функция работает как счетчик. Таким образом, код позволяет хранить и обновлять результаты теста для каждой категории ответов.

```
def page14(request):
    initialize_results(request)
    if request.method == 'POST':
        answer = request.POST.get('answer')
        if answer in request.session['results']:
            update_results(request, answer)
            return redirect('path15')
    return render(request, 'main/page14.html', {'question': 'Мне было бы интереснее:', 'results': request.session['results']})
```

В данной части кода приведен пример принципа работы одной страницы сайта. Суть данного фрагмента заключается в следующем: сначала функция проверяет, если ли уже результаты теста, если их нет, то она создаёт их. Следующим шагом идет обработка ответа пользователя, после чего выбранный вариант извлекается из Post-запроса. Затем идет проверка, есть ли выбранный ответ в заданных категориях, и если есть, то вызывается функция, которая обновляет результат теста на основе выбранного ответа. Потом пользователь переходит на указанную страницу в коде. Если же пользователь просто смотрит на страницу, то ему показывается текст вопроса и варианты ответа.

Все остальные страницы сделаны по такой же логике.

Результат:

Получаем сайт с профориентационным тестом, в котором содержатся вопросы с вариантами ответов для пользователя. Сайт сохраняет и обновляет результаты теста в зависимости от выбора ответа. Также на сайте присутствует правильная последовательность перехода между страницами и их передачей данных. После теста последняя страница выводит результаты, основываясь на ответах человека. В итоге, результатом проекта является понятный функционал и работающий сайт, который помогает определиться людям с будущей профессией.

Вывод:

Получился полноценный сайт, предназначенный для проведения тестирования пользователей с целью определения их профессиональных предпочтений и навыков

Список литературы:

1. Баранов, С. В. "Основы работы с Django." М.: Издательство, 2021.
2. Джон Дакетт. "HTML и CSS. Разработка и дизайн веб-сайтов"
3. Шабанов, Д. Ю. "Дизайн пользовательского интерфейса." М.: Издательство, 2020
4. Силин, П. А. "Проектирование и разработка веб-приложений." М.: Издательство, 2020.