

# **Исследовательская работа: Создание сайта с использованием искусственного интеллекта**

Лукина Вероника Дмитриевна 10Т  
Букина Юлия Алексеевна 10Т  
ГБОУ Школа “Покровский квартал”  
Руководитель: Коновалов Андрей  
Николаевич



# Оглавление

- Актуальность
- Цель и задачи работы
- Методика выполнения работы
- Результат
- Вывод
- Список используемой литературы



## Актуальность работы

Многие люди хотят создать свой сайт, например, интернет магазина, тестов, игр. Но не у всех есть навыки написания кода и время для обучения. Поэтому люди обращаются за помощью к искусственному интеллекту.



## Цель

Целью работы является создание сайта с профориентационным тестом, при помощи искусственного интеллекта.



# Задачи работы

1. Провести опрос: “Могут ли нейросети написать код для сайта?”
2. Определить тему сайта, для которого будет писаться код.
3. Найти искусственный интеллект, который будет помогать в написании кода.
4. Создание примерного дизайна
5. Разработка кода с помощью нейросетей (Frontend и Backend части) и его объяснение.
6. Проверка сайта и подведение итогов.



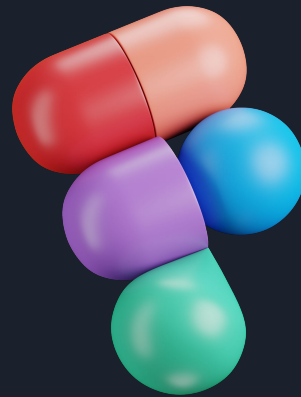
## Опрос

Перед началом работы над исследованием, был проведен опрос: “Могут ли нейросети создать сайт для человека”. Результаты показали, что из 100 опрошенных, 90 считают, что нейросеть может создать сайт за человека.

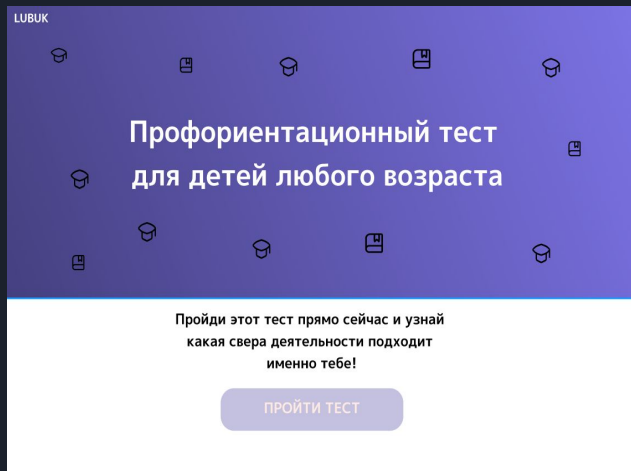
После проведения опроса, была выбрана тема сайта и поиск подходящих нейросетей. По итогу на сайте будет находиться профориентационный тест, а используемой нейросетью будет чат- GPT- бот в телеграмме.

# Методика выполнения работы

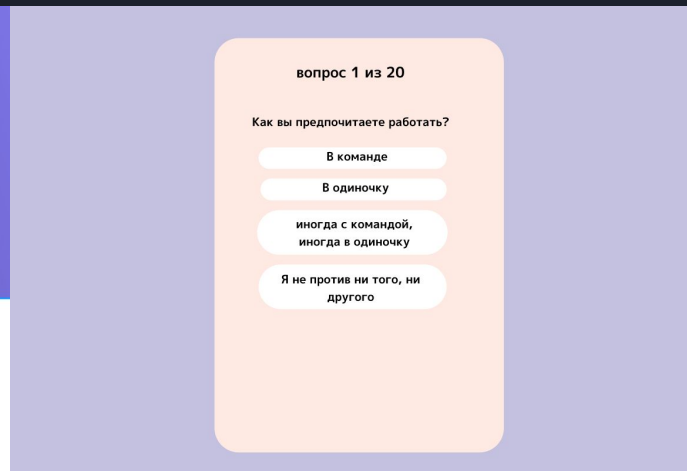
## Интерфейс и дизайн



- Предложение Figma
- Создание Desktop
- Основная страница
- Страница с анкетой
- Страница с вопросам
- Результат
- Рекомендация профессии



Вид основной страницы (пример)



Вид страницы с вопросами (пример)

# Методика выполнения работы

## Код и разработка. Frontend.

```
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;  
}
```

`body { ... }`

- Этот блок задаёт общий вид всей страницы: убирает отступы, выбирает шрифт и цвет фона, а также центрирует всё содержимое.

`team-work-preference-container1 { ... }`

- Этот блок помогает расположить элементы внутри себя по центру и добавляет пространство вокруг них.



# Методика выполнения работы

## Код и разработка. Frontend.

```
.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%;
```

teamwork-preference-container2 { ... }

- Здесь создаётся коробка с фоновым цветом, закруглёнными углами и тенью, чтобы она выглядела объёмнее.

header-title, .team-work-preference-heading { ... }

- Эти стили делают текст заголовков по центру и добавляют пространство снизу.

team-preference-container3 { ... }

- Этот блок располагает свои элементы вертикально (один под другим) и выравнивает их по центру.

team-member-card { ... }

- Это карточка для каждого варианта ответа. Она белая, с закруглёнными углами и немного внутреннего пространства.

# Методика выполнения работы

## Код и разработка. Frontend.


```
width: 100%;  
max-width: 400px;  
text-align: center;  
}  
  
.my-button {  
background-color: ■ rgb(220, 218, 233) ;  
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: ■ rgb(169, 166, 196) ;  
}  
  
</style>  
</head>
```

`my-button { ... }`

- Это кнопка с цветом, закруглёнными углами и текстом. Она меняет цвет, когда на неё наводят мышку.

`my-button; hover { ... }`

- Когда пользователь наводит курсор на кнопку (элемент с классом `my-button`), применяются стили, указанные в блоке `my-button:hover`.



# Методика выполнения работы


## Код и разработка. Backend.

```
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
```

первый блок создает словарь для хранения результатов теста, второй блок подготавливает результаты, и третий блок увеличивает счетчик для выбранного ответа. Таким образом, код позволяет хранить и обновлять результаты теста для каждой категории ответов в вашем приложении



## Методика выполнения работы Код и разработка. Backend.

```
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']})
```

Данная часть отвечает за логику и функционал. Для начала работы нужно было продумать логику сайта. Также отвечает за обработку ответа пользователя обновление результатов в сессии на основе этого ответа и передачу результатов и вопроса на следующую страницу теста.



## Результат

Был получен сайт с профорientационным тестом, в котором содержатся вопросы с вариантами ответов, где Backend часть писал чат GPT, а Frontend часть создавалась в Kombai.



## Вывод

Нейросети могут помочь создать сайт, но если вам потребуется изменить код, могут возникнуть ошибки, которые нейросеть не сможет исправить.

Нейросеть может лишь помочь написать сайт, но для устранения ошибок нужно самому изучить язык программирования.



# Список использованной литературы

- 1) Силин, П. А. "Проектирование и разработка веб-приложений." М.: Издательство, 2020.
- 2) Степанов, И. А. "Методы тестирования программного обеспечения." М.: Издательство, 2017.
- 3) Шабанов, Д. Ю. "Дизайн пользовательского интерфейса." М.: Издательство, 2020.