

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

УТВЕРЖДАЮ

Зав.кафедрой,

к. ф.-м. н.

_____ М. В. Огнева

ОТЧЕТ О ПРАКТИКЕ

студента 2 курса 273 группы факультета КНиИТ
Кулакова Максима Сергеевича

вид практики: производственная (научно-исследовательская работа)

кафедра: Информатики и программирования

курс: 2

семестр: 1

продолжительность: 18 нед., с 01.09.2023 г. по 14.01.2024 г.

Руководитель практики от университета,

к. э. н., доцент

Л. В. Кабанова

Руководитель практики от организации (учреждения, предприятия),

к. э. н., доцент

Л. В. Кабанова

Тема практики: «Разработка платформы единого резюме»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Анализ научной литературы	5
2 Реализация платформы единого резюме	6
2.1 Общие настройки динамического приложения	6
2.2 Написание механизма авторизации.....	7
2.3 Реализация страницы резюме	9
2.4 НН	14
2.5 Страница сравнения навыков.....	18
2.6 Страница обновления навыков	21
ЗАКЛЮЧЕНИЕ	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	26
Приложение А Код Компонента.....	27

ВВЕДЕНИЕ

Вопрос поиска работы всегда находился перед лицом человека, ведь работа должна приносить не только деньги, но и удовлетворение физических и психологических потребностей человека. Наиболее актуальной проблемой со стороны соискателя является то, где искать необходимого ему работодателя, а также с какой стороны преподнести свои навыки и умения, чтобы в ближайшие дни занимать рабочее место своей мечты.

Задачами работы являются следующие пункты:

1. Масштабирование платформы до клиент-серверного приложения;
2. Конфигурация основного функционала;
3. Реализация взаимодействия по API.
4. Обзор научной литературы (в том числе научно-технической) по теме «Разработка платформы единого резюме»;
5. Рассмотрение и анализ существующих платформ для создания резюме;
6. Формулировка собственных методов разработки единой платформы резюме;
7. Подведение итогов проведенной научно-исследовательской работы.

1 Анализ научной литературы

Рассматриваемая литература будет затрагивать тему аспектов составления резюме, принципы их составления и критерии, по которым работодателю с наибольшей вероятностью понравится грамотно составленное резюме. После проведения анализа данной темы нам предоставится возможность выделить основные пункты, которые будут учитываться при разработке собственной единой платформы резюме.

Для начала стоит рассмотреть научные статьи, связанные с доказательством важности правильного составления резюме в настоящее время, и какие изменения оно претерпевает. В статье К.В. Косолаповой «Типологические особенности современного резюме на английском языке» автор выделяет основные пункты в резюме, которые было принято считать достаточными:

1. Полные ФИО;
2. Возраст;
3. Место проживания на текущий момент;
4. Место учёбы, уровень образования;
5. Список умений;
6. Опыт работы (при его наличии);
7. Контактные данные. [1]

2 Реализация платформы единого резюме

2.1 Общие настройки динамического приложения

Для компенсации недостатков статических сайтов и динамической генерации страниц из набора входящих параметров и данных инициализируем проект с NextJS и добавим интерфейсную библиотеку NextUI для удобства разработки следующими командами:

```
1 npx create-next-app@latest
2 npm i @nextui-org/react framer-motion
```

Для безопасной работы с проектом необходимо создать файл `process.env`, в котором будут храниться переменные окружения, например токен доступа к системе.

Дополнительно для каждой страницы необходимо задать общий шаблон в файле `layout.tsx`, находящимся в корневой папке приложения. Данная структура представляет из себя обёртку над стандартным представлением HTML файла, внутрь которого добавлено реактивное представление. Дополнительно внутри атрибутов указываются стили для блоков, такие как минимальная высота контейнера, цвет фона, размеры контейнеров, отступов и их поведение, выбор шрифта и текущая цветовая схема. В дальнейшем содержимое атрибутов `className` будет игнорироваться.

```
1 export default function RootLayout({ children, }: {
2     children: React.ReactNode;
3 }) { return (
4 <html lang="en" suppressHydrationWarning>
5     <head></head>
6     <body className={clsx( "min-h-screen bg-background font-sans antialiased",
7       ↪ fontSans.variable )}>
8     <Providers themeProps={{ attribute: "class", defaultTheme: "dark" }}>
9     <div className="relative flex flex-col h-screen">
10    <Navbar/>
11    <main className="container mx-auto max-w-7xl pt-16 px-6 flex-grow">
12    {children}
13    </main></div></Providers></body>
14 </html>
15 );}
```

2.2 Написание механизма авторизации

Для реализации авторизации создадим файл `auth.ts`, в котором укажем конфигурацию с сервисом Github посредством открытого протокола безопасности OAuth и с классической проверкой логина и пароля. Для этого создадим два соответствующих свойства и в первое передадим переменные приложения для авторизации через сторонний сервис, а в втором реализуем проверку введённых в поля данных и в случае соответствия вернём самого пользователя с его именем, фотографией профиля и ролью.

```
1 export const authConfig: AuthOptions = { providers: [  
2   GithubProvider({  
3     clientId: process.env.GITHUB_ID!,  
4     clientSecret: process.env.GITHUB_SECRET!,}),  
5   Credentials({  
6     credentials: {  
7       email: { label: 'email', type: 'email', required: true },  
8       password: { label: 'password', type: 'password', required: true }, },  
9     async authorize(credentials) {  
10      if (!credentials?.email || !credentials.password) return null;  
11      const currentUser = users.find(user => user.email ===  
12        ↪ credentials.email)  
13      if (currentUser && currentUser.password === credentials.password) {  
14        const { password, ...userWithoutPass } = currentUser;  
15        return userWithoutPass as User;  
16      }  
17      return null })  
18    ], pages: { signIn: '/signin' } } }
```

Ограничение доступа неавторизованного пользователя к внутренним страницам достигается созданием файла `middleware.ts`, в котором через параметры указываются пути навигации и, в случае попадания в заданный путь, перевод клиента на страницу `signIn`.

```
1 export const config = { matcher: ['/profile/:path*', '/hh',  
2   ↪ '/resume', '/protected/:path*'] }
```

Дополнительно к предыдущему пункту контроля доступа опишем корневой файл `providers.tsx`, созданный для окружения корневого элемента страницы в её компоненты, свойства стилей и сессию пользователя.

```
1 export function Providers({ children, themeProps }: ProvidersProps) {  
2   return (  
3     <AuthProvider {...themeProps}>  
4       <Providers>{ children }</Providers>  
5     </AuthProvider>  
6   )  
7 }
```

```

3 <SessionProvider><NextUIProvider>
4   <NextThemesProvider {...themeProps}>
5     {children}
6   <NextThemesProvider>
7 </NextUIProvider></SessionProvider>;}

```

Процесс авторизации реализуется на клиентской и серверной части приложения, поэтому для корректного взаимодействия необходимо создать обработчик событий по API в динамическом маршруте [..auth]/route.tsx с следующими параметрами:

```

1 const handler = NextAuth(authConfig);
2 export { handler as GET, handler as POST }

```

Реализуем форму авторизации через компонент SignInForm с двумя текстовыми формами, для этого инициализируем роутер переходов, обратные вызовы для динамического контента и обработчик нажатия кнопки отправки формы, получающий данные полей и выполняющий вход в систему.

```

1 const SignInForm = () => {
2   const router = useRouter();
3   const [isVisible, setIsVisible] = React.useState(false);
4   const toggleVisibility = () => setIsVisible(!isVisible);
5   const handleSubmit: FormEventHandler<HTMLFormElement> = async (event) => {
6     event.preventDefault();
7     const formData = new FormData(event.currentTarget);
8     const res = await signIn("credentials", {
9       email: formData.get("email"),
10      password: formData.get("password"),
11      redirect: false

```

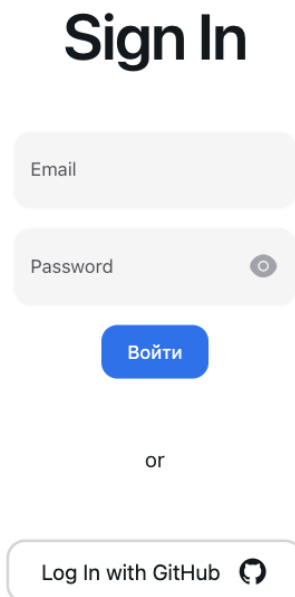
Кнопка авторизации через Github работает через параметры поисковой строки, получаемые в процессе возврата на страницу платформы после успешного входа на стороне внешнего сервиса.

```

1 export const GithubButton = () => {
2   const searchParams = useSearchParams();
3   const callbackUrl = searchParams.get("callbackUrl") || "/profile";
4   return (
5     <Button onClick={() => signIn("github", { callbackUrl })}>
6       Log In with GitHub
7     </Button>
8   );}

```


После написания компонентов необходимо создать саму страницу авторизации, для этого внутри всей иерархической оболочки страниицы передадим заголовок вместе с компонентами `SignInForm` и `GithubButton`, после которых страница будет располагаться по маршруту `app/signIn`. Итоговый вид страницы авторизации отображён на рисунке 2.1.



The image shows a 'Sign In' form. At the top, the text 'Sign In' is displayed in a large, bold, black font. Below it, there are two input fields: 'Email' and 'Password'. The 'Email' field is a simple light gray box. The 'Password' field is a light gray box with a small eye icon on the right side to toggle visibility. Below these fields is a blue button with the text 'Войти' (Login) in white. Underneath the button is the word 'or' in a small, gray font. At the bottom, there is a button with the text 'Log In with GitHub' and the GitHub logo icon.

Рисунок 2.1 – Страница авторизации

2.3 Реализация страницы резюме

Для написания модуля необходимо создать структуру данных резюме. Информация о профиле хранится в формате текстовом формате обмена данными JSON и содержит в себе следующие поля:

1. Id – генерируемый уникальный идентификатор резюме;
2. Name – фамилия и имя человека;
3. Description – краткое описание профиля;
4. Social links – ссылки на социальные сети;
5. Profile image – изображение работника для резюме;
6. About – текстовое поле с рассказом о себе, являющееся аналогом сопроводительного письма;
7. Education, experience – группы образования и опыта работы, включающие в себя следующие пункты:
 - а) Name – наименование места работы или учебного учреждения;
 - б) Description – описание чем занимался соискатель в данное время;

- в) Start date, end date – дата начала и окончания;
 - з) Current date – поле, показывающее что дата окончания отсутствует и при подсчёте её следует считать текущим днём;
8. Projects – поле проектов, следующее из опыта работы. Включает в себя:
- а) Name – наименование проекта;
 - б) Description – описание проекта и над чем происходила работа;
 - в) Image – изображение проекта, его логотип, интерфейс.

Последовательно реализуем компоненты, начиная с блока контактов и ссылок на социальные сети, получающий на вход список адресов и возвращающий горизонтальный список кнопок с переходом и миниатюрами соответствующих сервисов. Изображения предварительно описываются в формате svg в компоненте icons.tsx, после чего из строки выделяется имя хостинга с последующим добавлением в массив элементов для отображения на странице.

```

1 function IconVariant(service: string, icon_color:string, icon_size:number) {
2   switch(service) {
3     case "github.com":
4       { return <GithubIcon className={icon_color} size={icon_size}/>; }
5     default:
6       { return <InternetIcon className={icon_color} size={icon_size}/>; }
7   };
8   export const SocialLink = (props : SocialProps) => {
9     const icon_size, icon_color = 36, "text-default-500";
10    const social_icons = props.links.map((element, index) => {
11      var url = new URL(element.toString());
12    return
13      <Link isExternal href={element.toString()} key={index}>
14        {IconVariant(url.host.toString(), icon_color, icon_size)}
15      </Link>;
16  });};

```

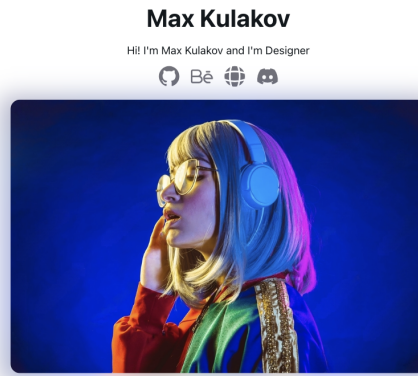
Итоговый вид компонента показан на рисунке 2.2.

Опишем компонент вывода образования и опыта работы. Для этого

```

1 function diffDates(day_one:any, day_two:any) {
2   return (day_one - day_two) / (60 * 60 * 24 * 1000);
3 };
4 function getFormatedStringFromDays(numberOfDays:number) {
5   numberOfDays = Math.abs(numberOfDays)
6   var years = Math.floor(numberOfDays / 365);

```



ABOUT ME

I am currently a senior designer at YKSoft. I live and work in Saratov and pursue a Master's degree in Computer Science at the Saratov State University.

I began my career in design in 2018. I started as a graphic designer and eventually realized that I'm more interested into the sphere of interfaces and products. Since then, I managed to work with many companies and in various projects, such as the Venrok sports app, the Lifetables habit tracker, and the Easy-Traveling travel app.

More cool and interesting cases can be viewed below on the page.

At the same time, I continue to work as a graphic designer, helping such companies as Youcon Events Group, Jedi.University and LoyalMed with visual and identity issues.

Рисунок 2.2 – Компонент отображения контактов и ссылок

```

7   var months = Math.floor(numberOfDays % 365 / 30);
8   var yearsDisplay = years > 0 ?
9     years + (years == 1 ? " year " : " years ") : "";
10  var monthsDisplay = months > 0 ?
11    months + (months == 1 ? " month " : " months ") : "";
12  return yearsDisplay + monthsDisplay;
13 };
14 export const Experience = (props : ExperienceProps) => {
15   const experience_item_list = props.experience_list.map((element, index)=>{
16     const date1 = new Date(element.start_date);
17     const date2 = element.current_date ?
18       new Date() : new Date(element.end_date)
19     return { getFormattedStringFromDays(diffDates(date1, date2)) }
20   })
21   return <{experience_item_list}</>
22 };

```

Итоговый вид компонента показан на рисунке 2.3.

Компонент проектов, выводящий шахматную структуру

```

1  export const Projects = (props : ProjectsProps) => {
2    const projects = props.projects_list.map((element, index) => {
3      var position_image = index % 2 == 0 ? "col-start-7 col-end-13" :
↪  "col-start-1 col-span-6 ";
4      var position_text = index % 2 == 0 ? "col-start-1 col-end-6" :
↪  "col-start-8 col-end-13 order-1";
5

```

EDUCATION		
2022-09-01 - 2023-10-30 1 year 4 months	Master's degree, Saratov State University	Computer Science, Mathematical Support and Administration of Information Systems
2017-09-01 - 2022-06-30 4 years 10 months	Bachelor's degree, Saratov State University	Computer Science, Mathematical Support and Administration of Information Systems. Graduate work: «Development of mobile application with implementation of accessibility»
EXPERIENCE		
2021-04-25 - 2023-10-30 2 years 8 months	Senior Designer, YKSoft	Audit, development and implementation of solutions in the products of various companies. Implementation of new projects and bringing them to release. Support for existing sites and the application in order to refine the functionality. Interaction with customers and related teams in the process of working on projects. Managing a mini-team of designers
2019-10-01 - 2021-02-01 1 year 4 months	Interface Designer, Venrok	Launch and development of projects from an idea to a final solution with subsequent support
2019-02-01 - 2019-10-01 8 months	Designer, Youcon Events Group	Participation in the organization of events. Work on the website, printed and digital products

Рисунок 2.3 – Страница обновления навыков на платформе hh.ru

```

6      return (
7          <div className="grid grid-cols-12 gap-2 mb-12" key={index}>
8              <div className={position_text + " h-full flex flex-col
↪ justify-center">
9                  <h2 className="h6 uppercase text-start pb-4"
↪ id="projects">{element.name}</h2>
10                     <p className="text-start">{element.description}</p>
11                 </div>
12                 <div className={position_image}>
13                     <Image/>
14                 </div>
15             </div>
16         )
17     })
18
19     return <>{projects}</>
20 };

```

Итоговый вид компонента показан на рисунке

Произведём запрос данных у сервера и отобразим их в описанных ранее компонентах, передав в них

```

1  export default function Template1Page() {
2      const profile_data = require("@/data-template/template-1-data.json");
3      return (
4          <div>
5              <h1 className="h6 text-uppercase text-center pb-4">
6                  {profile_data.name}
7              </h1>
8              <p className="text-uppercase text-center pb-2">

```

PROJECTS BY COMPANIES

■ YKSOFT

Work on projects for businesses in various fields from construction companies to the agro-industrial complex. Creation, revision and support of websites and products. Development of internal services and projects of the company.



■ CLIMBING LINES

An app for climbers that allows you to discover new places to climb and find different routes in locations. It contains selections based on users interests and collections moderated by experienced guides. In addition, the trip is accompanied by a convenient route tracking, which allows anyone to get the maximum of emotions from the ascent and not get lost. It is also possible to share feedback on the route and update information about it.

Рисунок 2.4 – Страница обновления навыков на платформе hh.ru

```

9         {profile_data.description}
10     </p>
11     <SocialLink links = {profile_data.social_links}/>
12     <div className="grid grid-cols-8 gap-4 mb-12">
13         <div className="col-start-3 col-end-7">
14             <Image
15                 isZoomed
16                 isBlurred
17                 alt={profile_data.name + " Image"}
18                 src={profile_data.profile_image}
19             />
20         </div>
21     </div>
22     <div className="container pt-5 pb-5 text-left">
23         <h2 className="h6 uppercase text-end pb-4"
24     ↪ id="about">about me</h2>
25         <About about_text={...profile_data.about}/>
26     </div>
27     <div className="container pt-5 pb-5 text-left ">
28         <h2 className="h6 uppercase text-end pb-4"
29     ↪ id="experience">education</h2>
30         <Experience experience_list={...profile_data.education}/>
31     </div>
32     <div className="container pt-5 pb-5 text-left ">

```

```

31         <h2 className="h6 uppercase text-end pb-4"
↪   id="experience">experience</h2>
32         <Experience experience_list={...profile_data.experience}/>
33     </div>
34     <div className="container pt-5 pb-5">
35         <h2 className="h6 uppercase text-end pb-4"
↪   id="projects">projects by companies</h2>
36         <Projects projects_list={...profile_data.projects}/>
37     </div>
38
39     </div>
40 );
41 }

```

2.4 НН

Процесс авторизации пользователя через hh.ru представляет из себя 3 ступени:

1. Перенаправление пользователя по адресу, с переданными параметрами приложения
2. Получение временного кода авторизации
3. Получение токена, кода доступа к API и кода обновления

В `api/hh/route.tsx` опишем получение токена авторизации и если он есть, получения резюме

```

1  export async function GET(req: Request) {
2
3  const { searchParams } = new URL(req.url)
4  const code = searchParams.get('code');
5  const token = searchParams.get('token');
6
7
8  if (code != null) {
9      var myHeaders = new Headers();
10     myHeaders.append("Content-Type", "application/x-www-form-urlencoded");
11
12     var urlencoded = new URLSearchParams();
13     urlencoded.append("grant_type", "authorization_code");
14     urlencoded.append("client_id", process.env.HH_ID!);
15     urlencoded.append("client_secret", process.env.HH_SECRET!);
16     urlencoded.append("code", code!);

```

```

17
18     try {
19         const res = await fetch("https://hh.ru/oauth/token", {
20             method: 'POST',
21             headers: myHeaders,
22             body: urlencoded,
23             redirect: 'follow',
24         });
25
26         const result = await res.json();
27         return NextResponse.json(result);
28     }
29     catch (error) {
30         console.log(error);
31     }
32 }
33
34
35 if (token != null) {
36     var myHeaders = new Headers();
37     myHeaders.append("Authorization", `Bearer ${token}`);
38
39     try {
40         const res = await fetch("https://api.hh.ru/resumes/mine", {
41             method: 'GET',
42             headers: myHeaders,
43             redirect: 'follow',
44         });
45
46         const result = await res.json();
47         return NextResponse.json(result);
48     }
49     catch (error) {
50         console.log(error);
51     }
52 }
53 return NextResponse.json("bad request");
54 }

```

В api/hh/route.tsx опишем обновление резюме

```

1 export async function PUT(req: Request) {
2     const { searchParams } = new URL(req.url)

```

```

3      const resume = searchParams.get('resume');
4
5      if (resume !== null) {
6          const body = await req.json()
7
8          var myHeaders = new Headers();
9          myHeaders.append("User-Agent", process.env.HH_APP_DATA!);
10         myHeaders.append("Content-Type", "application/json");
11         myHeaders.append("Authorization", `Bearer ${process.env.HH_USER}`);
12
13         var raw = JSON.stringify( body );
14
15         try {
16             const res = await fetch(`https://api.hh.ru/resumes/${resume}`, {
17                 method: 'PUT',
18                 headers: myHeaders,
19                 body: raw,
20                 redirect: 'follow'
21             })
22             return NextResponse.json({ message: "Resume has been updated" });
23         }
24         catch (error) {
25             console.log(error);
26             return NextResponse.json({ message: error });
27         }
28     }
29 }

```

Для взаимодействия с апи между клиентской и серверной частью опишем методы `getUserResume` и `getUserToken`, получающие токен и код данных

```

1  export async function getUserResume(token: string) {
2      try {
3          const response = await fetch('/api/hh?token=' + token);
4          const data = await response.json();
5
6          if (!data.error) {
7              return data.access_token;
8          } else {
9              return data.error;
10         }
11     }

```



```

12         } catch (error){
13             console.log(error);
14         }
15     }

```

И теперь опишем страницу авторизации через hh

```

1  export default function HHSignInPage() {
2
3  const searchParams = useSearchParams()
4  const code = searchParams.get('code')
5
6  const [url, setUrl] = React.useState('')
7  React.useEffect(() => {
8      getDownloadURL().then(setUrl)
9  }, [])
10 async function getDownloadURL(){
11     const resultData = getUserToken(code!);
12     return await Promise.resolve(resultData)
13 }
14 return (
15     <section className="flex flex-col items-center justify-center gap-24 ">
16         <h1>HH Sign In Page</h1>
17
18         <div className="flex flex-col items-center justify-center gap-2">
19             SignIn Token:
20             <Code size="lg">{code}</Code>
21         </div>
22
23         <div className="flex flex-col items-center justify-center gap-2">
24             Access Token:
25             <Suspense fallback={<div>Loading...</div>}>
26                 <Code size="lg">{url}</Code>
27                 <p className="opacity-60 mb-24">Сохраните токен из поля выше
28                     ↪ для дальнейшей работы с сайтом</p>
29
30                 <Link isExternal href={`'/api/hh?token=' + url}>
31                     <Button>Посмотреть мои резюме</Button>
32                 </Link>
33             </Suspense>
34         </div>

```

```

35
36
37     </section>
38 );
39 }

```

2.5 Страница сравнения навыков

Для страницы сравнения навыков реализуем компонент строки, принимающей строку данных

```

1  export const ActionFieldButton = () => {
2      return (
3          <div className="flex gap-4 items-center">
4              <Button color="success" variant="bordered">
5                  Add item
6              </Button>
7              <Button color="danger" variant="bordered" className="min-w-0">
8                  <DeleteIcon/>
9              </Button>
10         </div>
11     )
12 }

13
14 export const Comparison = (props: ComparisonItemProps) => {
15
16
17     let cv_editor_item_list;
18     if (typeof props.cv_editor == "string") {
19         cv_editor_item_list = <div>{props.cv_editor}</div>;
20     } else {
21         Object.keys(props.cv_editor).map((element, index) => {
22             let arrObj = props.cv_editor[element];
23             if (typeof arrObj == "object") {
24                 cv_editor_item_list = props.cv_editor.map((element: any,
↪   index: any) => {
25                     return (
26                         <div key={index}>
27                             {Object.keys(element).map((el, index) => {
28                                 return(
29                                     <div key={index}>
30                                         <span className="font-bold
↪   opacity-40">{el}: <span>{element[el]}

```

```

31         </div>
32     );
33     }}}
34     <Spacer y={4} />
35 </div>
36 );
37 });
38 } else {
39     cv_editor_item_list = props.cv_editor.map((element: any,
↪ index:any) => {
40         return (
41             <div key={index}>
42                 {element}
43                 <Spacer y={2} />
44             </div>
45         );
46     })
47 }
48 });
49 }
50
51
52 let hh_ru_item_list;
53 if (typeof props.hh_ru == "string") {
54     hh_ru_item_list = <div>{props.hh_ru}</div>;
55 } else {
56     Object.keys(props.hh_ru).map((element, index) => {
57         let arrObj = props.hh_ru[element];
58         if (typeof arrObj == "object") {
59             hh_ru_item_list = props.hh_ru.map((element: any, index:any) =>
↪ {
60                 return (
61                     <div key={index}>
62                         {Object.keys(element).map((el, index) => {
63                             return(
64                                 <div key={index}>
65                                     <span className="font-bold
↪ opacity-40">{el}: </span>{element[el]}
66                                     </div>
67                                 );
68                             }}}

```

```

69         <Spacer y={4} />
70     </div>
71 );
72 });
73 } else {
74     hh_ru_item_list = props.hh_ru.map((element: any, index:any) =>
↪ {
75         return (
76             <div key={index}>
77                 {element}
78                 <Spacer y={2} />
79             </div>
80         );
81     })
82 }
83 });
84 }
85
86
87
88 return (
89     <section className="">
90         <Divider className="my-8" />
91
92         <RadioGroup orientation="horizontal" defaultValue="answer-1"
↪ className="flex gap-2 mb-4 text-start ">
93             <div className="w-1/6 items-start font-bold opacity-80">
94                 {props.name}
95             </div>
96             <Radio value="answer-1" className="w-2/6 max-w-full
↪ items-start">
97                 {cv_editor_item_list}
98             </Radio>
99             <Radio value="answer-2" className="w-2/6 max-w-full
↪ items-start">
100                 {hh_ru_item_list}
101             </Radio>
102             <Radio value="answer-3" className="w-1/6 max-w-full
↪ items-start">
103                 <ActionFieldButton />
104             </Radio>

```

```

105     </RadioGroup>
106
107     </section>
108
109     });

```

HH Dev Page

Field Name	CV Editor	hh.ru	Custom
Full Name	<input checked="" type="radio"/> Max Kulakov	<input type="radio"/> Max Kulakov	<input type="radio"/> Add item 🗑
Description	<input checked="" type="radio"/> Hi! I'm Max Kulakov and I'm Designer	<input type="radio"/> Hi! I'm Max Kulakov and I'm Designer	<input type="radio"/> Add item 🗑
Social Links	<input checked="" type="radio"/> https://github.com/nesbox/TIC-80 https://www.behance.net/KulakovMax https://t.me/MaxKulakov https://discord.com/servers	<input type="radio"/> https://github.com/nesbox/TIC-80 https://www.behance.net/KulakovMax https://t.me/MaxKulakov https://discord.com/servers	<input type="radio"/> Add item 🗑
About	<input checked="" type="radio"/> I am currently a senior designer at YKSoft. I live and work in Saratov and pursue a Master's degree in Computer Science at the Saratov State University.	<input type="radio"/> I am currently a senior designer at YKSoft. I live and work in Saratov and pursue a Master's degree in Computer Science at the Saratov State University.	<input type="radio"/> Add item 🗑

Рисунок 2.5 – Страница сравнения навыков

2.6 Страница обновления навыков

Для страницы обновления навыков обратимся к ранее описанному api

```

1  export default function ResumePage() {
2
3      const session = useSession();
4      const profile_data = require("@/data-template/template-hh-data.json");
5
6      const initialSkills = profile_data.skill_set
7      const [skill_set, setSkill_set] = React.useState(initialSkills);
8
9      const handleClose = (skillToRemove:any) => {
10         setSkill_set(skill_set.filter((fruit: any) => fruit !==
↪ skillToRemove));
11         if (skill_set.length === 1) {
12             setSkill_set(initialSkills);
13         }
14     };
15
16     const skills_body = JSON.parse(JSON.stringify(`{"skill_set":
↪ [${skill_set.map((x: any) => `"${x}"`) } ] }`))

```

```

17     function updateResume() {
18         const postData = async () => {
19
20             const response = await
↪   fetch("/api/hh?resume=986544fbff0b05dce60039ed1f53716830586e", {
21                 method: "PUT",
22                 body: skills_body,
23             });
24             console.log(skills_body);
25             return response.json();
26         };
27         postData().then((data) => {
28             console.log(data.message);
29         });
30     }
31
32     const [inputValue, setInputValue] = React.useState("");
33
34
35     return (
36         <section className="flex flex-col items-center justify-center
↪   gap-24">
37             <h1 className={title()}>Резюме HH.ru</h1>
38
39
40
41             <div className="">
42                 <h2 className="mb-4">
43                     НАВЫКИ
44                     <span className="opacity-100">
45                         {skill_set == initialSkills?
46                             <div
↪   className="text-green-600"> [ не изменено ]</div>
47                             :
48                             <div
↪   className="text-yellow-600"> [ изменено ]</div>}
49                             </span>
50                         </h2>
51                     <div className="flex flex-wrap gap-2">
52                         {skill_set.map((skill: any, index:
↪   any) => (

```

```

53                                     <Chip key={index} onClose={()
↳ => handleClose(skill)} variant="flat">
54                                     {skill}
55                                     </Chip>
56                                 )))}
57                             </div>
58                         </div>
59
60                 <div className="w-full flex flex-col gap-2
↳ items-center justify-center max-w-[640px]">
61                     <Input
62                         label="Skill"
63                         placeholder="Add skill to pool"
64                         value={inputValue}
65                         onChange={setInputValue}
66                     />
67                     <Button
68                         className="max-w-[240px]"
69                         onClick={() => {
↳ setSkill_set([...skill_set, inputValue]); }}
70                     >
71                         Добавить навык
72                     </Button>
73                 </div>
74
75                 <Button onClick={updateResume}>Обновить
↳ резюме</Button>
76
77
78                 </section>
79             );
80 }

```

Итоговый вид страницы обновления навыков представлен на рисунке 2.1

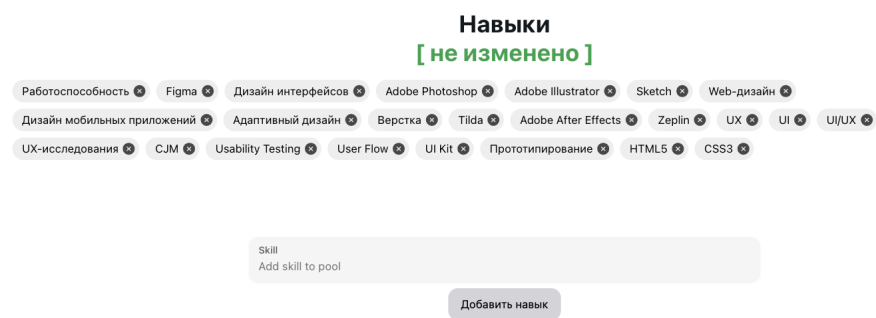


Рисунок 2.6 – Страница обновления навыков на платформе hh.ru

ЗАКЛЮЧЕНИЕ

В результате проведения исследовательской работы были приобретены навыки анализа качества и эффективности научной литературы в области разработки сервисов с автоматическим обновлением данных, достигнут навык анализа конкурентных платформ для создания резюме и сформулирован собственный метод разработки единой платформы резюме, тем самым было достигнуто полное выполнение поставленных задач.

В качестве объектов анализа научной литературы выступили статьи по темам составления резюме, их анализа со стороны социологии, а также статьи, в которых рассматриваются инструменты для веб-разработки. С учётом проведённого анализа научной литературы были составлены основные требования для разработки будущей платформы как с технической стороны, так и со стороны гуманитарно-социальных наук.

Анализ конкурентных платформ позволил выявить слабые стороны существующих сервисов, исправление которых возможно реализовать в разработке собственной единой платформы резюме при условии его дальнейшего масштабирования. [2]

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Гриднева, М. А.* Особенности работы с персоналом в it-сфере / М. А. Гриднева, М. Ю. Федосова // *Телескоп*. — 2021. — № 1. — С. 106–112.
- 2 *Архипов, Л. В.* Концепция применения контентно-адресуемых систем хранения для контроля версий / Л. В. Архипов // *Инновации в науке*. — 2015. — № 12 (49). — С. 42–46.

ПРИЛОЖЕНИЕ А
Код Компонента

Да