

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра Информатики и программирования

РАЗРАБОТКА ПЛАТФОРМЫ ЕДИНОГО РЕЗЮМЕ

КУРСОВАЯ РАБОТА

студента 1 курса 173 группы
направления 02.04.03 — Математическое обеспечение и администрирование
информационных систем
факультета КНиИТ
Кулакова Максима Сергеевича

Научный руководитель

к. э. н., доцент

Л. В. Кабанова

Заведующий кафедрой

к. ф.-м. н.

М. В. Огнева

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Анализ научной литературы	4
2 Анализ конкурентных платформ.....	9
2.1 Сервис поиска работы HeadHunter.ru	9
2.2 Сервис для поиска работы в IT-сфере Хабр Карьера.....	11
2.3 Международный сервис поиска работы Skipp.dev	14
2.4 Карьерный ресурс Icanchoose	16
2.5 Веб-сервис для хостинга ГитХаб	17
2.6 Блочный конструктор веб-сайтов Tilda	17
2.7 Онлайн-конструктор резюме Simpledoc	18
2.8 Онлайн-конструктор Enhancv	19
3 Основные аспекты разработки платформы единого резюме	23
3.1 Создание прототипа платформы резюме	23
3.2 Настройка рабочей среды	24
3.3 Реализация клиентской части платформы	25
3.4 Реализация серверной части платформы	29
3.5 Публикация платформы единого резюме	31
ЗАКЛЮЧЕНИЕ	33
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	34

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.

В настоящее время для поиска работы в интернете существует множество сервисов, при помощи которых работодатели могут как выкладывать свои вакансии, так и рассматривать входящие предложения от соискателей. Некоторые из платформ для поиска работы предоставляют возможность создавать резюме напрямую в личном кабинете соискателя.

Из-за того, что существует огромное количество сервисов для поиска работы, кому-то из работодателей не всегда будет удобна та или иная платформа, и своё предпочтение он отдаст “третьей”. Соискателю становится труднее ориентироваться в сайтах для поиска нужной вакансии, особенно если резюме на них висит уже долгое время, а навыки человека успели обновиться, что требует уже его дальнейшее обновление. Держать под контролем несколько резюме на нескольких сайтах становится всё сложнее, что становится проблемой для активного соискателя.

Целью научно-исследовательской работы является разработка платформы единого резюме с привязкой и последующим его обновлением на различных сервисах поиска работы (таких как GitHub, Хабр Карьера, hh.ru).

Задачами работы являются следующие пункты:

1. Обзор научной литературы (в том числе научно-технической) по теме «Разработка платформы единого резюме»;
2. Рассмотрение и анализ существующих платформ для создания резюме;
3. Формулировка собственных методов разработки единой платформы резюме;
4. Реализация клиентской и серверной частей платформы;
5. 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.

Автор также подчёркивает, что резюме является «визитной карточкой» для работодателя, и в зависимости от того, как оно составлено, будет зависеть решение о приглашении соискателя на собеседование. В статье раскрывается современное понятие «резюме», а также отмечается, что постепенное развитие и становление резюме определило его типологическое разнообразие. Так, традиционными типами резюме принято считать хронологическое (Chronological CV), функциональное (Functional CV) и комбинированное (Combination CV). После обзора на каждый из типов резюме автор рекомендует придерживаться следующим пунктам при составлении документа:

1. Личные сведения в начале резюме (имя, адрес, контактные телефоны, адрес электронной почты);
2. Краткое описание себя от первого или третьего лица в виде небольшого параграфа;
3. Список ключевых навыков, умений соискателя, а также занимаемых ранее должностей;

4. Опыт работы, профессиональные успехи, поставленные цели;
5. Образование, курсы, стажировки, академические степени, квалификации, членство в профессиональных организациях;
6. Дата рождения, пол, наличие водительского удостоверения;
7. Хобби и интересы.

Также, опираясь на зарубежные исследования, автор упоминает ряд рекомендаций по составлению резюме таким образом, чтобы работодатель с наибольшей вероятностью выбрал именно Вас на желаемую Вами вакансию [1].

Что касается сохранения единого резюме для вакансий, однообразие может конфликтовать с желанием соискателя попробовать себя в альтернативной сфере. Так автор Е.А.Шинкаренко в статье «Резюме как элемент практик поиска работы молодежью» в своей статье описывает эволюцию популярных вакансий, по которым чаще откликаются молодые люди возрастом 18-25 лет. Несмотря на смену интересов современной молодежи, в последнее время работодатели ставят более строгие критерии на свои вакансии, вследствие чего будущий соискатель должен будет обладать (в большинстве случаев) уровнем образования не ниже незаконченного высшего. В своём выводе автор статьи подчеркивает, что стремление молодых соискателей в профессиональные сферы при составлении нескольких резюме можно рассматривать как стратегию адаптации к разнообразию вакансий на рынке труда, применение образовательных компетенций к разным типам трудовой деятельности, разному содержанию труда, но то же время это делает рынок труда и профессий очень дифференцированным, что порождает стремление попробовать себя в разных его сегментах [2]. Так для выпускника высшего образовательного учреждения одно из резюме может быть попыткой реализовать диплом, но при этом другое резюме может быть составлено абсолютно по другой траектории развития (например, творческая специальность, связанная с хобби, вместо технической). Это можно рассматривать как желание занять максимально выгодные позиции, так и с точки зрения снижения рисков в условиях трансформации рынка труда и появления новых профессиональных сфер [3].

Рассматривая реализацию технической части будущего продукта, становится необходимым изучить вопрос хранения кода на сторонних сервисах, а также принципы его написания в качестве самостоятельного веб-сервиса. В статье С.А. Резина «Что должен знать начинающий Node.js разработчик» отмечается,

что наиболее популярный фреймворк Node.js обладает такой репутацией благодаря своей гибкости, лёгкости и высокой производительности, автор предлагает воспользоваться веб-фреймворками, которые обрабатывают запросы, пришедшие по протоколу HTTP, для работы по созданию собственного веб-сервиса [4]. Однако в статье автора также подчёркивается, что знания кодовой базы Node.js не будет достаточным для запуска веб-приложения, для чего необходимо дополнительно изучить возможности реализации интерфейсной части (при помощи JavaScript, HTML и CSS), хранения динамических данных непосредственно в базе данных и развертывания проекта на облачной платформе, например, Netlify, с последующим его запуском [5].

В статье В.М. Ефремова «Разработка веб-сайта с использованием Bootstrap» автор описывает преимущества библиотеки Bootstrap, подтверждая их на примере процесса создания веб-сайта и отмечая критерии качества, такие как адаптивность, кроссбраузерность и быстрота загрузки. Библиотеку Bootstrap создавали для того, чтобы уменьшить недостатки блочного отображения и свести использование альтернативных путей к минимально возможному. Также автор статьи демонстрирует преимущества использования данной библиотеки для веб-разработки, такие как особое внимание на работу с мобильными устройствами и большими возможностями адаптации сайта, кроссбраузерность, большой пакет внутренних CSS-классов, встроенные JS-функции, а также наличие различной документации на русском и английском языках [6]. Из недостатков применения Bootstrap автор указывает на размеры файлов в библиотеке, что влияет на скорость работы сайта, и блочный способ отображения, который не является гибким для воплощения веб-дизайнерских идей [7].

Для более комфортного взаимодействия сервера с клиентом в статье П.А. Безрука «Разработка системы распределенного мониторинга компьютерной сети на основе rest API» подчёркивается возможность использования REST API в качестве удобного архитектурного стиля [8]. В статье упоминается принцип работы REST API, а также рассматриваются решения для пользовательского интерфейса системы распределенного мониторинга, такие как SPA (Single Page Application) и фреймворк AngularJS, известный своим обширным функционалом [9] [10].

В статье М.А. Гарина, Д.К. Егоровой и С.Ф. Сайфетдинова «Создание Workflow аналитической платформы KNIME для анализа данных на примере

вакансий сайта HeadHunter» рассматривается процесс разработки кроссплатформенного приложения, написанного на языке Java с открытым исходным кодом для анализа данных. Работа приложения рассматривается на примере некоторых типов вакансий, размещенных на сервисе HeadHunter, с пошаговым наблюдением всех этапов и описания процессов и инструментов, применяемых на данном шаге [11]. Авторы статьи отмечают наличие открытого API на используемом ими сервисе в качестве примера [12].

Для хранения кода проекта часто приходится прибегать к специализирующимся на этом сервисах, о чём авторы Н.А. Грузин и А.А. Голубничий в статье «Обзор и сравнение хостингов для git-репозиторий: Bitbucket, Github и Gitlab» рассказывают в виде сравнения трёх основных хостингов: BitBucket, GitHub и GitLab. В статье упоминается, что использование репозиторий позволяет отследить ошибки, последние изменения и позволяет управлять релизами, списками рассылки и проектной документацией [13]. В качестве веб-сервиса хостинга авторы статей рассматривают в первую очередь Bitbucket, преимущества использования которым заключается в наличии бесплатных тарифных планов аккаунта, который поддерживает неограниченное количество частных репозиторий, а также поддержка таких функций, как проверки слияния, поиск кода, Git Large File Storage, дополнения и интеграции, использование REST API для создания сторонних приложений, которые могут использовать любой язык разработки, сниппеты и умное зеркалирование [14]. В случае с GitHub авторы статьи отмечают, что основной функционал доступен пользователям бесплатно, пока как продвинутые профессиональные и корпоративные являются коммерческими. Несмотря на это, подчеркивается, что основной целью GitHub является облегчение контроля версий и аспектов отслеживания проблем при разработке программного обеспечения. В качестве преимущественных функций GitHub авторы отмечают документацию, включая автоматически визуализированные файлы, README в различных форматах файлов, подобных Markdown, GitHub Actions, различные диаграммы с участниками, коммитами, частотой кода, возможность подписать кого-то на уведомления, упомянув их, возможность использования эмодзи, программа просмотра PDF-документов, а также оповещения системы безопасности об известных распространенных уязвимостях и уязвимостях в разных пакетах [15]. Рассматривая, инструмент GitLab, авторы статьи отмечают его развитие от решения для управления исходным кодом

до комплексного решения разработки программного обеспечения. В качестве ключевых различий между тремя платформами авторы статьи выделяют виды поддержки репозитория с открытым исходным кодом в платформе, импорт самих репозитория, дистрибуцию проектов, а также приводят сравнительную таблицу хостингов для git-репозитория [16].

2 Анализ конкурентных платформ

Существуют решения, предлагающие создание резюме на своей платформе с различными плюсами и минусами. Рассмотрим самые популярные и востребованные решения.

2.1 Сервис поиска работы HeadHunter.ru

HeadHunter является одним из крупнейших сервисов по поиску работы и сотрудников в России и по всему миру. Каждый месяц на сайте обрабатывается свыше сотни тысяч вакансий, и ещё большее количество людей имеют возможность найти работу мечты [17].

Со стороны соискателей алгоритм отклика на вакансию выглядит следующим образом:

1. Зайти в свой аккаунт на hh.ru;
2. Найти кнопку «Создать резюме»;
3. Заполнить пункт с контактными данными. Структура пункта представлена на рисунке 2.1.

Контактные данные

Имя	<input type="text" value="Имя"/>
Фамилия	<input type="text" value="Фамилия"/>
Мобильный телефон	<input type="text" value="+7"/>
Город проживания	<input type="text" value=""/>

Рисунок 2.1 – Структура пункта контактных данных

4. Заполнить пункты основной информации. Структура пункта представлена на рисунке 2.2.
5. Указать желаемую специальность и заработную плату. Структура пункта представлена на рисунке 2.3.
6. Указать опыт работы (при его наличии), а также навыки, которые предлагаются пользователю в качестве отдельных ключевых слов;

Основная информация


Дата рождения	<table><tr><td>День</td><td>Месяц ▼</td><td>Год</td></tr></table>	День	Месяц ▼	Год
День	Месяц ▼	Год		
Пол	<p><input type="radio"/> Мужской</p> <p><input type="radio"/> Женский</p>			
Гражданство	<div><div></div><div>Россия </div></div>			
Опыт работы	<p><input type="radio"/> Есть опыт работы</p> <p><input type="radio"/> Нет опыта работы</p>			

Рисунок 2.2 – Структура пункта основной информации

Специальность

Желаемая должность	<div></div>
Зарплата	<div><div></div><div>руб. ▼</div><div>на руки</div></div>

Рисунок 2.3 – Структура пункта специальности

7. Указать уровень образования, место его получения и года выпуска (либо «по настоящее время» для школьников или студентов);
8. Указать владение языками и его уровень (для иностранных). Структура пункта представлена на рисунке 2.4.

Владение языками

Родной язык	<div>Русский ▼</div>						
Иностранные языки	<table><tr><td>Английский ▼</td><td>C2 — В совершенстве ▼</td><td>×</td></tr><tr><td>Немецкий ▼</td><td>B1 — Средний ▼</td><td>×</td></tr></table> <p>Указать ещё один язык</p>	Английский ▼	C2 — В совершенстве ▼	×	Немецкий ▼	B1 — Средний ▼	×
Английский ▼	C2 — В совершенстве ▼	×					
Немецкий ▼	B1 — Средний ▼	×					

Рисунок 2.4 – Структура пункта владения языками

9. Пункт «другой важной информации» содержит в себе сведения о готовности к переезду, желаемой занятости, графика работы, наличии автомобиля

и водительских прав, а также категорий в них. Для иностранных граждан присутствует пункт «разрешения на работу».

После публикации резюме его может быть не видно большинству работодателей, если некоторые из пунктов являются незаполненными. Сам алгоритм составления резюме не является сложным, а возможность откликнуться на вакансию часто подразумевает прикрепление сопроводительного письма помимо самой «визитки» соискателя. Дополнительно сервис hh.ru предлагает услуги экспертов для составления грамотного резюме за небольшую плату (от 3 до 8 тысяч рублей в зависимости от разновидности услуги).

2.2 Сервис для поиска работы в IT-сфере Хабр Карьера

Являясь одним из популярных в России коллективным IT-блогом, Хабр смог развить не только форум для программистов, но и отдельные сервисы, которые связаны с помощью начинающих и опытных разработчиков, тестировщиков, дизайнеров и прочих информационных вакансий. Одним из подобных сервисов для поддержки начинающих и опытных IT-специалистов является Хабр Карьера [18].

В отличие от hh.ru, Хабр Карьера публикует вакансии исключительно связанные с IT сферой. На сервисе предлагаются вакансии как небольших компаний, так и компаний-гигантов (например, Яндекс, Авито, Mail.ru). Для своих коллег и знакомых на Хабр Карьера пользователю предоставляется возможность оставить профессиональную рекомендацию.

Составление резюме на сайте начинает свой путь с процесса авторизации на сервисе. Это можно сделать как при помощи стандартной регистрации с подтверждением почты, так и через сервисы, доступные в России (Вконтакте, Google Account).

Составить своё резюме Хабр Карьера предлагает сразу же после авторизации, причём существует возможность импортировать резюме с сервиса hh.ru. Данное окно представлено на рисунке 2.5.

Для ручного ввода или создания первого резюме пользователю потребуется 3 шага:

1. На первом указывается фамилия и имя, пол и дата рождения, а также основная цель регистрации на Хабр Карьера. Структура пункта представлена на рисунке 2.6.

В той же вкладке (если выбрана роль соискателя) необходимо выбрать

Быстрое заполнение с hh

Мы автоматически подгрузим нужную информацию о вас из вашего резюме на hh.ru. Проверьте, что оно составлено на русском или английском языке и не скрыто настройками приватности

Ссылка на резюме

Импортировать

Рисунок 2.5 – Предложение импорта резюме с hh.ru

Давайте знакомиться!

Имя *

Фамилия *

Пол

☒ Мужской ☐ Женский

Дата рождения

Основная цель *

Выберите вашу цель на Хабр Карьере. Если со временем она изменится – не страшно! Сейчас это поможет нам понять, какой мини-курс в письмах вам отправить: для соискателей или для эйчаров и рекрутеров.

☒ Я ищу работу ☐ Я нанимаю сотрудников

Рисунок 2.6 – Первый шаг регистрации

основную специализацию и отдельный профиль, а также квалификация (Intern, Junior, Middle, Senior, Lead). В отличие от hh.ru и в связи с ограниченной сферой деятельности, все специализации представлены наглядно и распределены по категориям для удобства выбора. Структура пункта представлена на рисунке 2.7.

Дополнительно указываются профессиональные навыки, которым владеет соискатель. Список таких навыков очень обширен и позволяет гибко найти нужные профессиональные «скиллы». Часть из них предлагается уже на старте как «Самые популярные». Структура пункта представлена на рисунке 2.8.

- Вторая вкладка будет содержать в себе контактную информацию, что также важно при составлении резюме, ведь одних данных об имени и фамилии будет недостаточно для связи с соискателем. По содержанию оно аналогично сервису hh.ru, например, указать город проживания и пункт о готовности к переезду или удаленной работе. Однако для связи соискатель может оставить не только номер телефона или ссылку-портфолио,

Выберите одну или две специализации, чтобы мы могли рекомендовать вам вакансии, а вас — эйчарам и рекрутерам. **Что означает каждая специализация.**

в, вначале укажите самые важные. Точный стек вакансии и курсы.

Рисунок 2.7 – Выбор специализации и квалификации

Профессиональные навыки *

Выберите до десяти своих профессиональных навыков, вначале укажите самые важные. Точный стек навыков поможет нам подбирать для вас подходящие вакансии и курсы.

Введите навык

Git

JavaScript

HTML

SQL

CSS

Python

Linux

PostgreSQL

Управление проектами

PHP

MySQL

ООП

React

Java

Docker

Управление людьми

Адаптивная верстка

Подбор специалистов

C#

Веб-разработка

Adobe Photoshop

TypeScript

Node.js

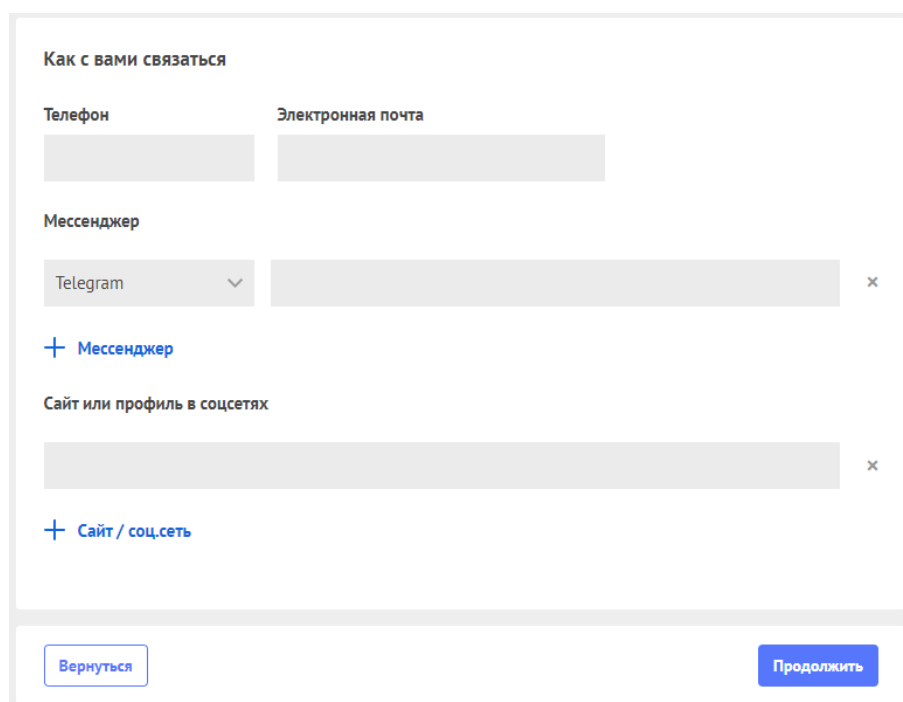
Английский язык

JQuery

Рисунок 2.8 – Выбор профессиональных навыков

но и свой логин в мессенджерах, например, в Telegram. Структура пункта представлена на рисунке 2.9.

3. Последняя вкладка содержит в себе пункты, связанные с опытом работы.



Как с вами связаться

Телефон

Электронная почта

Мессенджер

Telegram

+ Мессенджер

Сайт или профиль в соцсетях

+ Сайт / соц.сеть

Вернуться

Продолжить

Рисунок 2.9 – Выбор контактных данных

По статистике, большинство работодателей присматриваются к кандидатам, у которых за спиной есть даже самый незначительный, но указанный в резюме опыт. Однако на Хабр Карьера отсутствует пункт, связанный с «фрилансом», что мешает заполнению опыта работы в данной сфере, но его можно добавить самостоятельно.

2.3 Международный сервис поиска работы Skipp.dev

Skipp.dev является международным сервисом как для поиска работы, так и для найма работников со стороны работодателей. Различие между предыдущими сервисами отличается не только в плане дизайна, но и порядка заполнения пунктов. Для того, чтобы начать поиск желаемой вакансии, необходимо также пройти авторизацию, и после этого поочередно заполнить пункты специализации и навыков:

1. Необходимо верифицироваться по номеру мобильного телефона. Этот пункт необходим для того, чтобы проверить подлинность аккаунта, а также для дальнейшей авторизации на сервисе при помощи смс-кода.
2. На первом шаге соискателю предлагается выбрать свои навыки. Структура пункта представлена на рисунке 2.10.
3. После выбора навыков необходимо указать уровень навыков, которые были выбраны на предыдущей странице, что является нестандартным для

Choose your skills

This will match you with the most suitable projects

SPECIALIZATION

QA	DevOps	Team Leader	Backend Developer
Frontend Developer	Product Manager	Mobile Developer	
AR/VR Developer	Project Manager	No-Coder	Designer
Account Manager	Analyst	System Architect	
Data Scientist	Blockchain		

STACK

Figma	Graphic Design	Marketing Design	Animation	3D Design
Other	UX/UI Design	Web Design	Mobile Design	Animation
2D/3D Animation	Branding			

Next

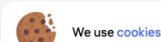


Рисунок 2.10 – Выбор навыков

отечественных сервисов. Структура пункта представлена на рисунке 2.11.

← SETTING UP PROFILE: STEP 2 OF 6

How many years experience do you have?

STACK	LEVEL	LOOKING FOR A JOB
Figma	4 Years	<input checked="" type="checkbox"/>
UX/UI Design	4 Years	<input checked="" type="checkbox"/>
Web Design	4 Years	<input checked="" type="checkbox"/>
Mobile Design	4 Years	<input checked="" type="checkbox"/>

Next



Рисунок 2.11 – Выбор уровня навыков

4. Заполнение опыта работы не сильно отличается от рассмотренных нами предыдущих платформ, но имеет свои преимущества. Например, для более подробного описания должностных обязанностей на предыдущих местах работы пользователю предлагаются различные виды деятельности, также начиная с самых популярных на сайте. Также к опыту работы есть возможность приложить изображения в качестве портфолио. Структура пункта представлена на рисунке 2.12.
5. Последний пункт содержит в себе форму заполнения контактной информации, начиная от ФИО до номера телефона и e-mail.

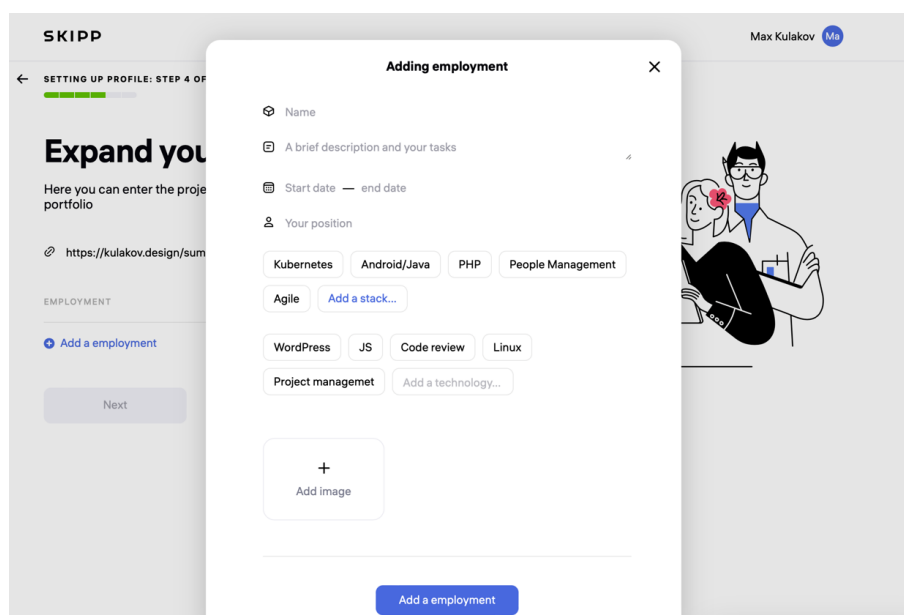


Рисунок 2.12 – Выбор опыта работы

Из положительных сторон данного сервиса можно выделить приятный внешний вид и неторопливое пошаговое заполнение всех пунктов. Акцент делается на опыте работы, чтобы сконцентрировать внимание пользователя на том, как будет лучше преподнести себя будущим работодателям. Однако сервис не захватывает резюме с других платформ, что делает эту платформу абстрагированной от других себе подобных.

2.4 Карьерный ресурс Icanchoose

ICanChoose рекомендует себя в качестве карьерного ресурса нового формата. Из новизны, предпологаемо, сервис предлагает помощь в поиске работы, а также предлагает карьерные советы.

Авторизация на сервисе доступна при помощи ВКонтакте, а так же с email. При создании резюме сразу предлагается импорт из сервиса hh.ru, представленный на рисунке 2.13.

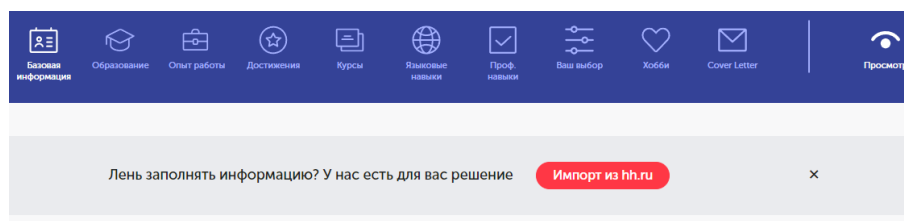


Рисунок 2.13 – Возможность импорта резюме

Процесс составления резюме происходит так же, как на вышерассмотренных сервисах, но с добавлением пунктов о хобби, достижениях и сопроводи-

тельное письмо. Также на сервисе существует возможность предварительного просмотра готового резюме, которое работает только при полном заполнении основной информации.

2.5 Веб-сервис для хостинга ГитХаб

ГитХаб зарекомендовал себя крупнейшим сервисом для хостинга с возможностью их совместной разработки, но, благодаря своим инструментам, на платформе возможно создание резюме. На аккаунте начинающего разработчика такое решение будет являться хорошим продолжением стратегии его развития, так как при рассмотрении профиля в качестве портфолио работодателю будут видны все разработанные проекты, на основе чего высока вероятность получить приглашение самому разработчику.

Для начала работы с составлением резюме пользователю необходимо создать новый репозиторий с названием, которое будет повторять «юзернейм» на GitHub. Сервис подчеркнёт его в качестве уникального и захватит его нужным образом.

Вся информация по резюме будет находиться в файле README.md. Другими словами, всё написанное и отформатированное будет видно на странице в GitHub и будет служить в дальнейшем красочной визиткой для тех, кто будет просматривать профиль разработчика [19].

Удобство написания резюме в GitHub подкрепляется отсутствием шаблона и полной свободой мысли, однако из-за отсутствия точных критериев содержания информации необходимо для себя составлять структуру будущей визитки, чтобы она выглядело не только гармонично, но и понятно.

2.6 Блочный конструктор веб-сайтов Tilda

Tilda изначально является конструктором, позволяющим на внутренних шаблонах создать любой сайт, который удовлетворяет задаче, в том числе резюме. Для последнего на сайте есть готовые шаблоны, которые упрощают работу с сервисом, в частности для начинающих пользователей [20]. Визуально это представлено на рисунке 2.14.

Составленный из блоков шаблон также можно будет использовать в качестве собственного лендинга.

Профиль на Тильде также является портфолио, что позволяет работодателю, желающему создать сайт на данном сервисе, найти необходимого ему



Шаблон сайта с резюме

Стильно расскажите потенциальным работодателям о себе. Разработайте сайт с вашим резюме без знания дизайна и навыков HTML-верстки на [конструкторе Tilda Publishing](#)

Создать сайт

Просмотр шаблона

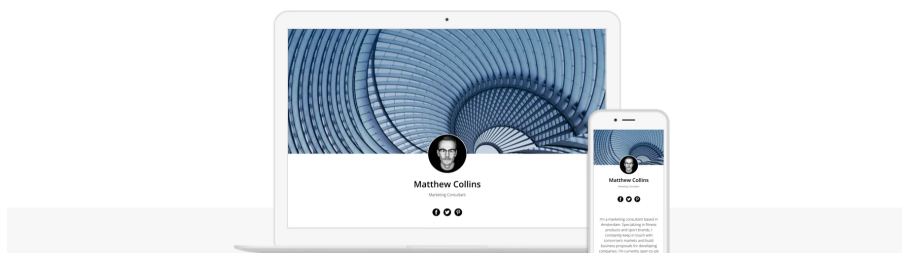


Рисунок 2.14 – Предложение использования шаблона

разработчика при помощи системы заказов на Tilda Express.

Резюме на Tilda Express представляет из себя небольшую страницу с размещенными на них проектами, разработанными на Tilda, а также окошком с заполнением заявки на обратную связь при наличии желания заказать у данного разработчика собственный сайт. Пример резюме представлен на рисунке 2.15.

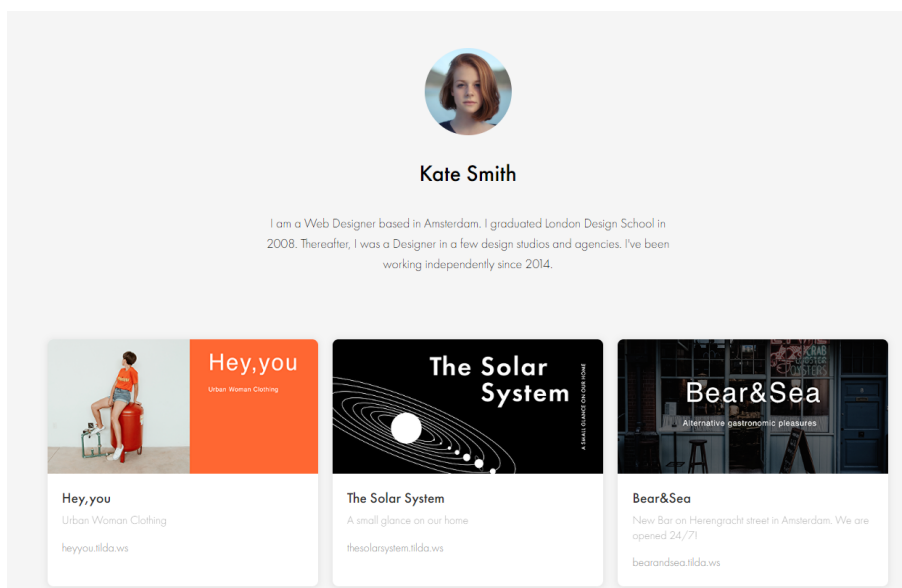


Рисунок 2.15 – Пример резюме на платформе Tilda

2.7 Онлайн-конструктор резюме Simpledoc

SimpleDoc является сервисом исключительно по составлению резюме. На сайте предлагается составление резюме в трёх основных этапах: заполнение формы, выбор дизайна для резюме и его скачивание или отправка по e-mail, представленных на рисунке 2.16.

Составьте эффективное резюме

Для этого достаточно выполнить три простых действия:

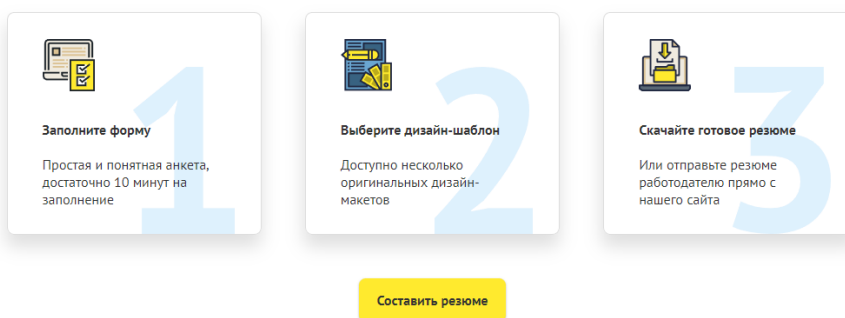


Рисунок 2.16 – Этапы создания резюме

Сам процесс заполнения формы занимает около 10ти минут и содержит в себе стандартные для резюме пункты:

1. Основная информация;
2. Личная информация;
3. Опыт работы;
4. Образование;
5. Курсы и тренинги;
6. Иностранные языки и компьютерные навыки;
7. Дополнительная информация.

После заполнения пунктов сервис предлагает выбрать один из четырёх шаблонов для дальнейшего скачивания, представленных на рисунке 2.17.

После составления резюме по шаблону пользователь сможет скачать его и редактировать в течение месяца после разовой оплаты на самом сервисе. Сервис также не позволяет импортировать резюме с других сервисов.

2.8 Онлайн-конструктор Enhancv


Enhancv также является сервисом для составления резюме, но в отличие от SimpleDoc имеет приятный дизайн, представленный на рисунке 2.18 уже со стартовой страницы:

При начале работы нам помогает «виртуальный» помощник, показанный на рисунке 2.19, с которым мы заполняем пункты по очереди:

1. Имя Фамилия
2. Наличие опыта в создании резюме (для возможности его импортировать)
3. Готовые шаблоны для будущего резюме.

Заполните поля выделенные желтым маркером

Консервативный Классический **Современный** Прогрессивный



Фамилия Имя Отчество

Желаемая зарплата

не указано

✓ Полная занятость

✓ Полный день

Контакты

не указано

не указано

Компьютерные навыки

✓ Печать, сканирование, копирование документов

✓ Интернет

✓ Электронная почта

Личная информация

Гражданство: не указано

Образование: Высшее

Дата рождения: 8 января 2005 (18 лет)

Пол: Мужской

Семейное положение: Холост

Опыт работы

не указано

не указано

январь 2022 - январь 2023 (1 год)

Образование

не указано

не указано

2023, очная форма обучения

Дополнительная информация

Наличие водительских прав (категории): B, BE

Личные качества:

Отсутствие вредных привычек, энергичность, инициативность, самостоятельность, ответственность, коммуникабельность, быстрая обучаемость

Рисунок 2.17 – Вариант шаблона резюме

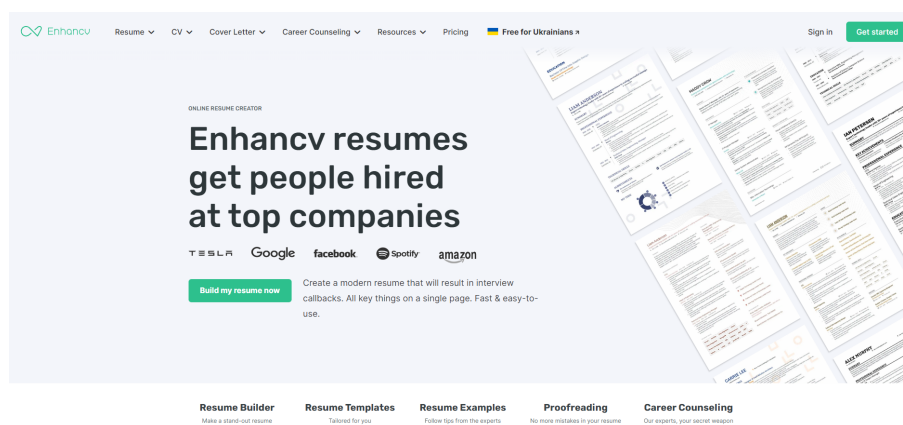


Рисунок 2.18 – Внешний вид сервиса Enhancv

После чего предлагается отредактировать информацию в резюме напрямую в одном из шаблонов, показанном на рисунке 2.20:

На выбор предлагается множество инструментов для комфортной работы с шаблоном: от возможности редактирования шрифта до редактируемой инфо-

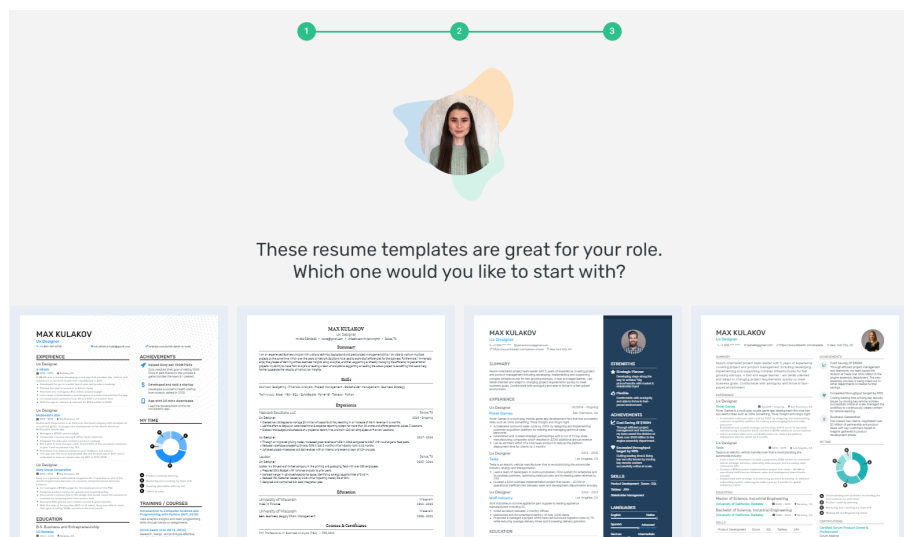


Рисунок 2.19 – Помощник в составлении резюме

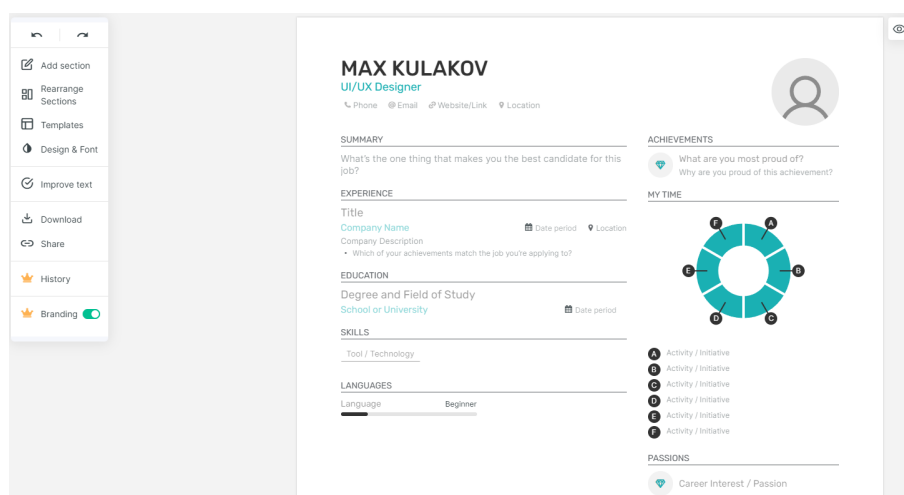


Рисунок 2.20 – Внешний вид шаблона резюме

графики. Также является платным сервисом.

На основе просмотренных конкурентных платформ можно вынести их основные недостатки:

1. Не все сервисы предлагают импорт резюме из других платформ;
2. Сервисы, работающие исключительно на составление резюме, берут деньги за свои услуги;
3. Некоторые сервисы при указании опыта работы требуют официальное название компании или должности, даже если они не зарегистрированы, и не всегда удается указать нужное название;
4. В дополнение к пункту 3, не везде возможно указание опыта «свободных заказов» вне сервисов по типу «Фриланс.ру»
5. Для русскоязычного сегмента сервисов поиска работы основным является

именно HeadHunter, импорт резюме с которого предлагается лишь в части платформ.

3 Основные аспекты разработки платформы единого резюме

С учётом собранных данных при анализе научной литературы и конкурентных сервисов для разработки собственной платформы единого резюме необходимо придерживаться следующим пунктам:

1. Реализовать сбор информации о текущих резюме с уже существующих сервисов;
2. Предоставить возможность объединения, замены, дополнения или удаления смежных или отсутствующих пунктов данных;
3. Создание единого бланка с отредактированными или созданными полями, с возможностью хранения их на личном пространстве;

Для составления базового и необходимого на начальное время функционала, в качестве развития платформы единого резюме, следует реализовать следующее:

1. Удобный пользовательский интерфейс, соответствующий современным web-стандартам и позволяющий пользоваться функционалом сервиса на любом устройстве, поддерживающим работу с браузерами;
2. Возможность создания пользователем персональной страницы на основе шаблонов, для возможности доступа к актуальному резюме при переходе по ссылке сервиса;
3. Возможность переноса резюме из стороннего ресурса в шаблон платформы;
4. Публикацию серверного проекта на хостинг, поддерживающий динамическое создание и обновление страниц согласно подготовленным данным.

3.1 Создание прототипа платформы резюме

Прототипирование – разработка интерактивной модели приложения, симулирующее коммуникацию пользователя с интерфейсом и созданное для тестирования базового функционала. Данный этап необходим чтобы удобно и корректно выстроить логику взаимодействия с продуктом и достижения поставленной задачи.

Для создания прототипа использовано приложение Figma, так как оно является бесплатным (платная версия отличается от бесплатной количеством единовременных редакторов проекта и невозможностью организовать команду) и полностью подходит для реализации данного этапа разработки. Были

изображены экраны бланка, страницы ввода и свободного редактора контента, показанные на рисунке 3.1.

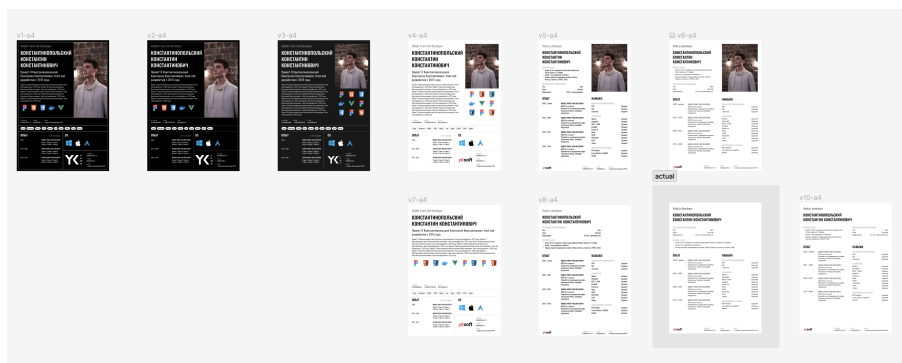


Рисунок 3.1 – Внешний вид прототипов платформы

Далее идёт проверка прототипа на реальных людях, проведённая коридорным тестированием – перед случайными людьми ставится задача взаимодействовать с интерфейсом приложения, тогда как разработчик убеждается в корректности задуманного поведения и находит ошибки. Такие тесты помогают максимально быстро получить обратную связь, потратив минимальное количество времени на поиск респондентов. Результатом тестирования стало подтверждение правильности разработанного прототипа, поэтому можно перейти к увеличению детализации экранов и написанию кода. Важным аспектом при разработке прототипа было сохранение адаптивности страницы, чтобы контент вписывался в различные разрешения экрана и кроме того мог быть адаптирован под печатную версию в формате А4.

3.2 Настройка рабочей среды

Создание приложения происходило на компьютере под управлением Mac OS, что позволяет использовать Terminal – средство взаимодействия с системой посредством командной строки при помощи bash-команд. В качестве среды разработки был выбран Visual Studio Code – легковесный текстовый редактор для кроссплатформенной разработки.

С помощью терминала установим пакетный менеджер Homebrew, который необходим для загрузки и запуска компонентов системы. Сделать это можно с помощью команды:

```
1 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/  
↪ HEAD/install.sh)"
```


Для работы с библиотеками разрабатываемой платформы необходимо выполнить установку Node.js – среды запуска Java Script приложений и Watchman – утилиты для контроля изменений файлов:

```
1 brew install node
2 brew install watchman
```

Инициализируем проект, создав стартовое рабочее приложение:

```
1 mkdir cv-editor
2 cd cv-editor
3 npm init
```

После перехода в папку проекта и инициации проекта необходимо добавить требуемые зависимости, такие как parcel, необходимый для итоговой сборки и работы в режиме сервера и editor.js, необходимый для использования возможностей динамического ввода и сохранения информации пользователя:

```
1 yarn add --dev parcel
2 yarn add @editorjs/editorjs
```

3.3 Реализация клиентской части платформы

Входным файлом проекта назначим index.html. Опишем его стандартную структуру и внутрь тега head добавим ссылки на файл стилей main.css и файл шрифта Mona-Sans.woff2 и стили библиотеки Bootstrap:

```
1 <title>CV Editor</title>
2 <link rel="stylesheet" href="css/main.css">
3 <link rel="preload" href="/src/font/Mona-Sans.woff2" as="font"
  ↪ type="font/woff2">
4 <link href="bootstrap.min.css" rel="stylesheet">
```

В том же файле добавим блок вызова редактора, кнопку сохранения введённых данных и после в конец файла перед закрытием тега body подключим файл скриптов проекта index.js и ссылку на файл скриптов библиотеки стилей:

```
1 <div id="editorjs"></div>
2 <button type="button" id="saveButton">Save</button>
3 <script src="/index.js" type="module"></script>
4 <script src="bootstrap.bundle.min.js"></script>
```

После краткого описания языка разметки нам необходимо реализовать функционал редактора. Для этого в файле `index.js` импортируем саму библиотеку и вспомогательные модули для отображения и работы с заголовками и стилями. Для описания самого редактора создадим новый объект редактора, включающий в себя вышеописанные библиотеки:

```
1 import EditorJS from '@editorjs/editorjs';
2 import Header from '@editorjs/header';
3 import List from '@editorjs/list';
4
5 const editor = new EditorJS({
6   holder: 'editorjs',
7   tools: {
8     header: {class: Header, inlineToolbar: ['link']},
9     list: {class: List, inlineToolbar: true},
10  },
11 });
```

Для использования введенной пользователем информации из блока редактора реализуем функцию получения данных по нажатию кнопки сохранения. Для этого создадим обработчик события, находящего элемент на странице по идентификационному значению и через него вызовем метод сохранения. Чтобы избежать падения сервера опишем обработчик исключений, возвращающий данные в случае успешного выполнения и выводящий ошибку в случае возможного возникновения проблем:

```
1 document.getElementById("saveButton").onclick = function() {
2   editor.save().then((outputData) => {
3     console.log('Article data: ', outputData)
4   }).catch((error) => {
5     console.log('Saving failed: ', error)
6   });
7 };
```

Создадим заголовок страницы, добавим классы для корректного позиционирования кнопок на странице с помощью библиотеки Bootstrap. Для этого создадим тег с классом контейнера и внутренними отступами с помощью классов `container` и `p-4`. Добавим краткую подсказку для запуска проекта на время разработки и стилизуем кнопку: `btn btn-outline-primary`. Итоговый вид и функционал редактора показан на рисунке 3.2.

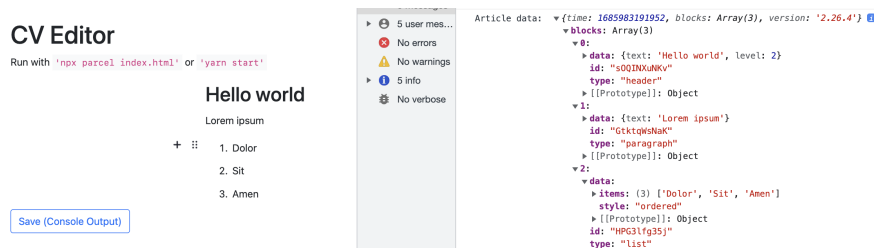


Рисунок 3.2 – Внешний вид свободного редактора данных

Реализуем вкладки для шаблона с двумя экранами: заполненным резюме и полями ввода. Для этого внутрь контейнера добавим список из двух значений, добавим им атрибуты вкладок и обозначим первый таб за активный, чтобы при заходе на страницу платформы бланк загружался сразу.

```

1 <div class="container p-4">
2   <ul class="nav nav-tabs" id="myTab" role="tablist">
3     <li class="nav-item" role="presentation">
4       <button class="nav-link active" id="home-tab" data-bs-toggle="tab"
        ↳ data-bs-target="#home" type="button" role="tab"
        ↳ aria-controls="home" aria-selected="true">CV Template</button>

```

Согласно дизайну сверстаем шаблон и предусмотрим адаптивное отображение страницы, чтобы сайт корректно показывался как на смартфонах, так и на планшетах и компьютерах. Для этого необходимо правильно описать стили в классах тегов – от наименьшего или стандартного разрешения к самому большому.

```

1 <h5 class="h5 text-uppercase text-secondary mt-5">
2 <strong>Персональная информация</strong></h5>
3 <div class="row">
4   <div class="col-6 col-lg-3">
5     Пол
6   </div>
7   <div class="col-6 col-lg-3 text-end">
8     Муж
9   </div>
10 </div>
11 <div class="row">
12   <div class="col-6 col-lg-3">
13     Язык
14   </div>
15   <div class="col-6 col-lg-3 text-end">
16     Русский

```

```

17     </div>
18 </div>

```

Дополнительно для удобства вёрстки воспользуемся сеткой, представляющей из себя вертикальные направляющие, делящие html-страницу на равные части относительно центра страницы с равными отступами от краёв экрана и равным отступом между колонками. Так на компьютерах с разрешением 1920 пикселей боковой отступ будет составлять 365 пикселей, размер рабочего контейнера – 1190 пикселей, всего будет 12 колонок с расстоянием между ними в 30 пикселей. Помимо этого внутренние контейнеры также адаптивно делятся в своих пропорциях, образуя сетку в сетке. Итоговый вид шаблона и первой вкладки показан на рисунке 3.3.

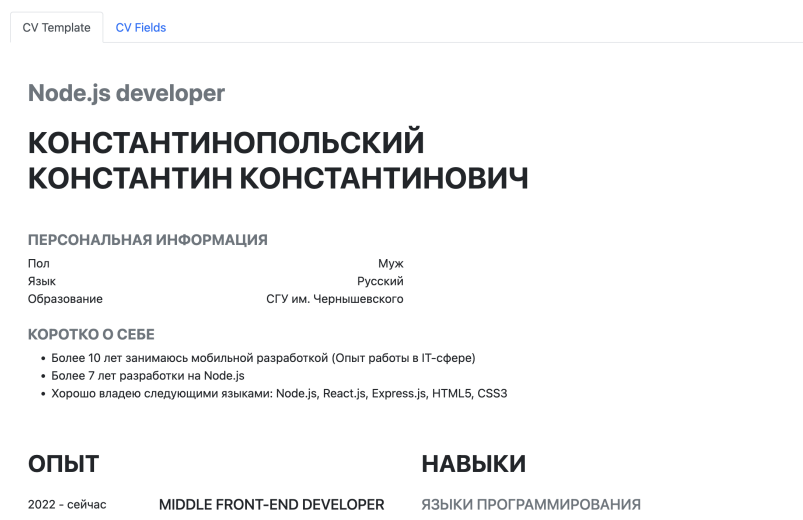


Рисунок 3.3 – Отображение вкладки переключения и шаблона

После вёрстки бланка резюме необходимо реализовать ручные поля ввода, из которых будет приходить информация для страницы. Разобьём их по категориям и представим пользователю в доступном виде, необходимом для последовательного заполнения. Для этого воспользуемся оформлением стилей из библиотеки Bootstrap и реализуем однострочные, многострочные поля ввода и селекторы, в которых изначально предложены варианты выбора данных и стандартное значение.

```

1 <h3 class="h3 pb-2">Personal information</h3>
2   <div class="row g-4 pb-4">
3     <div class="col-md">
4       <div class="form-floating">
5         <select class="form-select" id="floatingSelectGrid">

```

```

6         <option selected>Select</option>
7         <option value="1">Male</option>
8         <option value="2">Female</option>
9     </select>
10    <label for="floatingSelectGrid">Gender</label>
11</div>
12</div>

```

Результатом написания данного блока является вкладка с множеством полей ввода, показанных на рисунке 3.4.

Рисунок 3.4 – Отображение вкладки ввода данных

Структура хранения данных о пользователе представляет из себя json-файл, содержащий все описанные выше поля, вроде имени, даты рождения, пола и так далее. Данный выбор формата универсален в рамках нашего приложения и позволяет взаимодействовать с информацией быстрым и удобным способом.

```

1  "last_name": "Константинопольский",
2  "first_name": "Константин",
3  "middle_name": "none",
4  "age": 33,
5  "birth_date": "1991-04-30",
6  "gender": "male",

```

3.4 Реализация серверной части платформы

После написания клиентской части приложения необходимо реализовать часть, отвечающую за перенос резюме из сторонних проектов. Для решения поставленной задачи воспользуемся парсером html-страницы на основе библиотеки Puppeteer. Иницилируем новый проект аналогично тому, что было в реализации клиентской части и добавим библиотеку для скраппинга:

```

1 mkdir parser-cv-editor
2 cd parser-cv-editor
3 npm init
4 yarn add puppeteer

```

Работа парсера заключается в создании экземпляра браузера и выполнении заранее прописанных действий. Для этого создадим файл `browser.js` и опишем логику работы компонента. Так как браузер будет открываться в терминальном состоянии – его визуальное представление необходимо только для отладки работы и во время публикации параметр можно выключить:

```

1 const puppeteer = require('puppeteer');
2 async function startBrowser(){
3     let browser;
4     try {
5         console.log("Opening the browser");
6         browser = await puppeteer.launch({
7             headless: true,
8             args: ["--disable-setuid-sandbox"],
9             'ignoreHTTPSErrors': true
10        });
11    } catch (err) {
12        console.log("Could not create a browser instance => : ", err);
13    }
14    return browser;
15 }
16 module.exports = {
17     startBrowser
18 };

```

За управлением экземпляра браузера отвечает `pageController.js`. В его задачи входит проход по указанной странице и сохранение полученных данных в хранилище в формате `json`. Для этого подключается с помощью `require` в Node JS подключается работа с файловой системой, после чего инициализируется объект браузера и через асинхронную функцию происходит считывание страницы с её последующей записью. Дополнительно реализуем обработчик исключений чтобы работа серверной части не нарушалась в случае возникновения ошибок как в момент перехода на страницу, так и в момент записи информации:

```

1 const fs = require('fs');
2 async function scrapeAll(browserInstance){

```

```

3     let browser;
4     try{
5         browser = await browserInstance;
6         let scrapedData = {};
7         scrapedData = await pageScraper.scrapers(browser);
8         fs.writeFile("data.json", JSON.stringify(scrapedData), 'utf8',
↪     function(err) {
9             if(err) {return console.log(err);}
10            console.log("Done");
11        });
12    } catch(err){console.log("Error in ", err);}
13 }
14 module.exports = (browserInstance) => scrapeAll(browserInstance)

```

После описания вспомогательных файлов серверной части платформы необходимо реализовать точку входа в приложение. Для этого в `index.js` подключим вышеописанные компоненты, создадим экземпляр браузера и запустим работу контроллера с данным объектом:

```

1 const browserObject = require('./browser');
2 const scraperController = require('./pageController');
3 let browserInstance = browserObject.startBrowser();
4 scraperController(browserInstance)

```

3.5 Публикация платформы единого резюме

Перед публикацией необходимо изменить файл настроек проекта, указав точку входа и конфигурацию скрипта в файле `package.json`:

```

1 "default": "index.html",
2 "scripts": "start": "parcel ./index.html"

```

Так как платформа работает в режиме сервера – публикация в виде статичной страницы не представляется возможным, поэтому воспользуемся хостингом Netlify, позволяющим реализовать данную потребность. Первоначально необходимо загрузить свой код в систему контроля версий GitHub, после чего связать аккаунт с аккаунтом хостинга, синхронизировать репозитории и настроить параметры публикации. В итоговом виде платформа соберётся в оптимизированный для получения пользователем вид, закешируется в папке `dist` и будет выдаваться браузеру при обращении по ссылке. Настройки публикации сайта в Netlify представлены на рисунке 3.5.

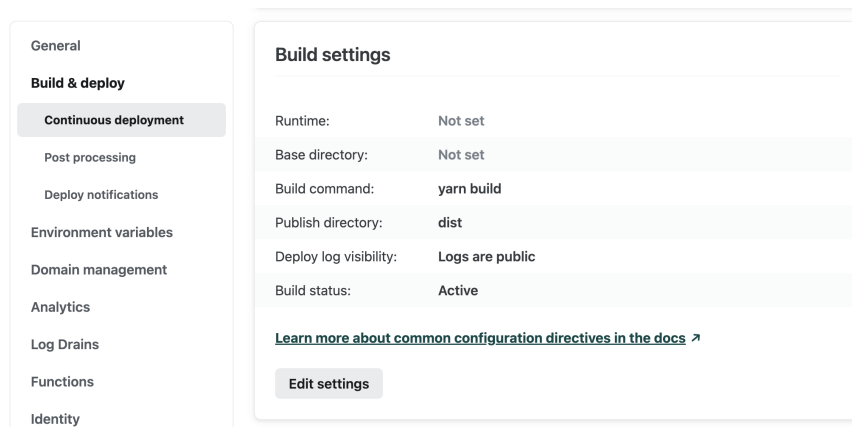


Рисунок 3.5 – Настройка публикации сайта

После реализации кода мы получили готовый сервис, в котором написаны все задуманные на данном этапе функции. Итоговый вид главной страницы проекта показан на рисунке 3.6.

CV Editor

CV Template CV Fields

Node.js developer

КОНСТАНТИНОПОЛЬСКИЙ КОНСТАНТИН КОНСТАНТИНОВИЧ

ПЕРСОНАЛЬНАЯ ИНФОРМАЦИЯ

Пол	Муж
Язык	Русский
Образование	СГУ им. Чернышевского

КОРОТКО О СЕБЕ

- Более 10 лет занимаюсь мобильной разработкой (Опыт работы в IT-сфере)
- Более 7 лет разработки на Node.js
- Хорошо владею следующими языками: Node.js, React.js, Express.js, HTML5, CSS3

ОПЫТ		НАВЫКИ	
2022 - сейчас	MIDDLE FRONT-END DEVELOPER	ЯЗЫКИ ПРОГРАММИРОВАНИЯ	
	ООО Рога и копыта Разработчик в медицинской сфере Технологии: Node.js, MongoDB, integrations	PHP SQL Javascript	Средний Средний Средний
2021 - 2022	MIDDLE FRONT-END DEVELOPER	ТЕХНОЛОГИИ	
		Node.js	Средний

Рисунок 3.6 – Итоговый вид страницы проекта

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, на основании которых был разработан функционал собственной единой платформы резюме, включающей в себя серверную и клиентскую части приложения с последующей публикацией проекта на серверном хостинге для корректного взаимодействия с возможностями платформы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Косолапова, Т. В. Типологические особенности современного резюме на английском языке / Т. В. Косолапова // *Поволжский педагогический вестник*. — 2020. — Т. 8, № 2 (27). — С. 30–35.
- 2 Шинкаренко, Е. А. Резюме как элемент практик поиска работы молодежью / Е. А. Шинкаренко // *Вестник Нижегородского университета им. Н. И. Лобачевского. Серия: Социальные науки*. — 2022. — № 2. — С. 135–140.
- 3 Шинкаренко, Е. А. Сценарии поиска работы студентами и выпускниками социоэкономических специальностей / Е. А. Шинкаренко // *Вестник Пермского национального исследовательского политехнического университета. Социально-экономические науки*. — 2021. — № 2. — С. 35–50.
- 4 Разин, С. А. Что должен знать начинающий Node.js разработчик / С. А. Разин // *E-Scio*. — 2020. — Т. 9, № 48.
- 5 Холмс, С. Стек MEAN. Mongo, Express, Angular, Node / С. Холмс; Под ред. Н. Гринчик. — Санкт-Петербург: Питер, 2017.
- 6 Прохоренок, Н. А. Bootstrap и CSS-препроцессор Sass. Самое необходимое / Н. А. Прохоренок. — Санкт-Петербург: БХВ, 2021.
- 7 Ефремов, В. М. Разработка веб-сайта с использованием Bootstrap / В. М. Ефремов // *Инновационная наука*. — 2020. — № 6. — С. 40–42.
- 8 Безрук, П. А. Разработка системы распределенного мониторинга компьютерной сети на основе Rest API / П. А. Безрук // *Актуальные проблемы авиации и космонавтики*. — 2017. — Т. 2, № 13. — С. 94–95.
- 9 Миковски, М. Разработка одностраничных веб-приложений / М. Миковски, П. Джош; Под ред. А. А. Слинкин. — Москва: ДМК Пресс, 2018.
- 10 Файн, Я. Angular и TypeScript. Сайтостроение для профессионалов / Я. Файн, А. Моисеев. — Санкт-Петербург: Питер, 2022.
- 11 Гарин, М. А. Создание Workflow аналитической платформы KNIME для анализа данных на примере вакансий сайта HeadHunter / М. А. Гарин, Д. К. Егорова, С. Ф. Сайфетдинов // *Огарёв-Online*. — 2021. — Т. 12, № 165. — С. 1–6.

- 12 API HeadHunter. Общая информация [Электронный ресурс]. — URL: <https://github.com/hhru/api/blob/master/docs/general.md> (дата обращения 22.12.2022). Загл. с экр. Яз. рус.
- 13 Грузин, Н. А. Обзор и сравнение хостингов для git-репозиторий: Bitbucket, Github и Gitlab / Н. А. Грузин, А. А. Голубничий // *E-Scio*. — 2021. — № 1 (52). — С. 588–596.
- 14 Архипов, Л. В. Концепция применения контентно-адресуемых систем хранения для контроля версий / Л. В. Архипов // *Инновации в науке*. — 2015. — № 12 (49). — С. 42–46.
- 15 Чакон, С. Git для профессионального программиста / С. Чакон, Б. Штрауб. — Санкт-Петербург: Питер, 2017.
- 16 Bitbucket vs Github vs Gitlab Infographics [Электронный ресурс]. — URL: <https://www.educba.com/bitbucket-vs-github-vs-gitlab/> (дата обращения 12.12.2022). Загл. с экр. Яз. англ.
- 17 О компании HeadHunter [Электронный ресурс]. — URL: <https://saratov.hh.ru/article/28> (дата обращения 15.12.2022). Загл. с экр. Яз. рус.
- 18 Гриднева, М. А. Особенности работы с персоналом в it-сфере / М. А. Гриднева, М. Ю. Федосова // *Телескоп*. — 2021. — № 1. — С. 106–112.
- 19 Пишем резюме на GitHub. Краткий гайд [Электронный ресурс]. — URL: <https://javarush.com/groups/posts/3424-pishem-rezjume-na-github> (дата обращения 20.12.2022). Загл. с экр. Яз. рус.
- 20 Создание сайта с резюме [Электронный ресурс]. — URL: <https://tilda.cc/ru/tpls/sozdanie-saita-rezume/> (дата обращения 22.12.2022). Загл. с экр. Яз. рус.