МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА И СОЗДАНИЕ КРОСС-ПЛАТФОРМЕННОГО ПРИЛОЖЕНИЯ ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ С ФУНКЦИОНАЛОМ СОЦИАЛЬНОЙ СЕТИ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Зыкова Кирилла Анатольевича

Научный руководитель
зав.каф., доцент, к. ф.-м. н. _____ С. В. Миронов

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2024

ВВЕДЕНИЕ

В современной мире люди ежеденевно взаимодействуют друг с другом в интернете: читают новости, делятся фотографиями и видео, общаются в текстовых чатах и видеоконференциях, а также поддерживают любимых авторов и общественные проекты. Социальные сети включают в себя обширный функционал для удовлетворения потребностей людей. Некоторым трудно представить повседневную рутину без взаимодействия с данными платформами. Социальные сети способствуют формированию цифровой общественности, где каждый может найти свое место и выразить себя.

Однако, с ростом популярности социальных сетей, увеличивается и их конкурентная борьба за внимание масс. Пользователи становятся всё более требовательными к интерфейсам и функционалу социальных платформ, ожидая интуитивно понятные и эффективные инструменты для общения и взаимодействия. Поэтому разработка отзывчевого пользовательского интерфейса приложений, отвечающих современным требованиям, становится приоритетной задачей для разработчиков. Также немаловажной частью подобных приложений является масштабируемость. Потребителям важно использовать веб-ресурсы на различных, им удобных, устройствах.

Объединившись в команду разработчиков, мы приняли участие в программе «Стартап как диплом». Было решено создать современное веб-приложение с функционалом социальной сети и сбором средств для социально значимых проектов. Главная идея нашей разработки — предоставить удобный ресурс для взаимодействия пользователей в сети для обмена информацией и продуктами творчества. Проект ориентирован на разработку платформы, которая обеспечит безотказное и безопасное взаимодействие пользователей.

Целью данного проекта является разработка современного пользовательского интерфейса для веб-приложения с функционалом социальной сети и поддержки социально-значимых инициатив, а также его портирование на ряд популярных устройств.

Чтобы достичь поставленную цель, необходимо реализовать следующие задачи:

1. Выбрать правильный подход для создания веб-приложения.
2. Разработать современный дизайн социальной сети.
3. Перенести получившиеся эскизы в веб-представление.

4. Добавить отзывчивые реакции на действия пользователя.
5. Масштабировать веб-приложение на различные устройства.

Структура и объем работы. Для решения поставленных задач выполнена выпускная квалификационная работа, включающая в себя введение, 2 основные главы, заключение, список использованных источников из 20 наименований и 4 приложения. Работа изложена на 57 страницах.

Первая глава имеет название «Анализ предметной области» и содержит основную информацию о доступных на текущий момент технологий разработки веб-приложений и обеспечения их непрерывной работы.

Вторая глава имеет название «Разработка программной части», данная глава содержит подробное описание процесса выполнения работы.

Выпускная квалификационная работа заканчивается заключением, списком использованных источников, а также приложения с кодом А-Г.

Основное содержание работы

Постановка задачи. Основной задачей проекта является создание веб-приложения для платной доставки контента с функционалом социальной сети, которое будет удобно как пользователям приложения, так и талантам, которые будут размещать продукты своего творчества на этой площадке. Ключевыми требованиями к разработке являются современный дизайн, удобство использования и безотказная работа на различных устройствах. **Описание продукта**. Продукт имеет название «Ghosty» и прямо приглашает ценителей творчества в гости к популярным авторам. Приложение будет представлено в виде веб-сайта с функционалом социальной сети, маркетплейса информационных товаров и площадки для сбора пожертвований. Пользователи смогут создавать новости, в которых будут делиться своим творчеством или собирать средства на реализацию масштабных идей.

Главный заработок площадки будет осуществляться с комиссии за доставку платного контента. Основной схемой для информационного обмена между пользователями будет служить гибкая система подписок, которую каждый автор сможет настроить под себя в зависимотсти от его целей и ценности продуктов творчества. Также будет осуществлена возможность сбора средств на нужды автора или социально значимые проекты.

Анализ существующих решений. Одним из конкурентов на российском рынке является веб-сайт `Sponsr.ru`. Это платформа для монетизации контента через подписки, имеет несколько значительных недостатков, которые могут повлиять на её пользователей. Также существует зарубежная компания `Patreon`. Это популярная платформа для создателей контента. Компания рассчитана на мировой рынок, поэтому не так отзывчиво относится к пользателям из Российской Федерации, поэтому будет оказывать косвенную конкеренцию нашему проекту. Также `Patreon` имеет несколько недостатков, которые могут оказать влияние как на самих создателей, так и на их аудиторию.

Обзор инструментальных средств.

HTTP (Hypertext Transfer Protocol) — это протокол передачи данных в сети Интернет. Он используется для обмена различными видами информации между веб-серверами и клиентами, такими как веб-браузеры. Протокол HTTP разработан с учетом простоты, что облегчает взаимодействие между клиентом и сервером.

HTML (HyperText Markup Language) — это язык разметки, используемый для создания структуры веб-страниц. С его помощью определяются элементы и их взаимосвязи на странице, такие как заголовки, параграфы, списки, изображения и ссылки. HTML использует теги для обозначения различных элементов, каждый из которых имеет свою функцию и свойства.

CSS (Cascading Style Sheets) — это язык таблиц стилей, который используется для оформления веб-страниц. С помощью CSS можно определить внешний вид HTML-элементов, такие как цвет текста, шрифт, размеры, отступы, рамки, и многое другое. CSS позволяет создавать красивый и структурированный дизайн, разделяя структуру и стиль веб-страницы. CSS правила могут применяться к элементам непосредственно в HTML-коде или быть описаны в отдельных файлах стилей.

Sass (Syntactically Awesome Style Sheets) — это один из самых популярных CSS препроцессоров. Он добавляет функциональность к стандартному CSS, позволяя использовать переменные, вложенные правила, миксины, функции и многое другое.

Figma была выбрана для создания макета визуальной составляющей пользовательского интерфейса и имеет множество достоинств, о которых рассказано в тексте ВКР.

В процессе разработки интерфейса веб-приложения использовался следующий стек технологий. Для создания и прототипирования дизайна был использован инструмент Figma, который позволил эффективно планировать и визуализировать интерфейс. Основу разметки веб-страниц составлял HTML, язык разметки гипертекста, который обеспечивает структуру веб-страниц. Стилизация была реализована с помощью CSS, каскадных таблиц стилей, и мощного фреймворка Tailwind CSS, обеспечивающего удобную и быструю работу со стилями благодаря утилитарному подходу. Управление зависимостями и сборкой проекта осуществлялось с использованием пакетных менеджеров Yarn и npm, которые помогают управлять библиотеками и инструментами, необходимыми для разработки. Взаимодействия с сервером были реализованы с помощью Node.js, среды выполнения, позволяющей запускать JavaScript на сервере. Для создания настольного приложения использовался фреймворк Electron, что позволило разрабатывать кросс-платформенное приложение, работающее на различных операционных системах.

Реализация работы приложения.

Разработка пользовательской части веб-приложения с функционалом социальной сети включает в себя несколько ключевых этапов: проектирование, создание дизайна, установка и настройка окружения для разработки, веб-верстка страниц приложения и конфигурирование файлов настройки адаптивности.

Проектирование.

В ходе работы над проектированием веб-приложения были составлены функциональные требования к будущему продукту, а такде был разработан ряд диаграмм: диаграммы случаев использования, диаграммы активности.

Диаграмма случаев использования предоставляет четкое понимание функциональных возможностей системы и взаимодействия пользователей с системой, что помогает в дальнейшей разработке, тестировании и поддержке веб-приложения.

Диаграмма активности отображает поток работы в системе, моделируя процессы и действия, выполняемые системой. Она позволяет визуализировать последовательность действий и условия перехода между ними, что помогает понять бизнес-логику и рабочие процессы внутри системы.

Создание дизайна.

Следующий этап разработки приложения – процесс создания пользовательского интерфейса (UI) для веб-приложения платной доставки контента с функционалом социальной сети с использованием инструмента для прототипирования и дизайна —- Figma. В работе описаны этапы реализации пользовательского интерфейса.

1. Исследование и анализ требований.
2. Создание wireframes (каркасов).
3. Разработка визуального дизайна.
4. Прототипирование.
5. Переход дизайна в разработку.
6. Тестирование и итерации.

Ниже будут приведены примеры прототипов веб-страниц, которые перешли в раздел официального дизайна приложения (см. рисунки 1, 2).

Установка необходимых компонентов.

Для эффективной работы над проектами веб-верстки необходимо правильно настроить среду разработки. На данном этапе будут рассмотрены ос-

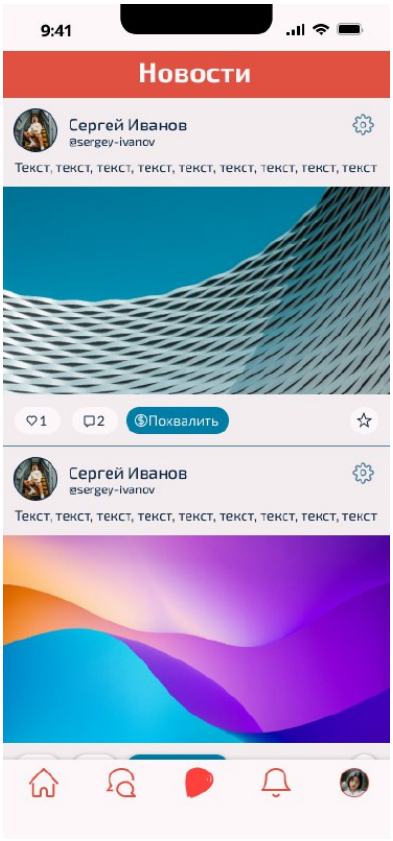

Рисунок 1 – Главная страница ресурса с контентом

новные шаги по установке необходимых инструментов, таких как пакетные менеджеры, Node.js, `TailwindCSS` а также необходимые расширения для среды разработки, которые позволят оперативно отслеживать изменения кода и нагляднее вноить в него правки.

1) Установка среды разработки. Первым шагом в настройке среды разработки является установка текстового редактора или интегрированной среды разработки (IDE). Наиболее популярные редакторы для веб-разработки включают Visual Studio Code, Sublime Text и Atom. Visual Studio Code (VS Code) является одним из самых популярных и предлагает множество расширений, облегчающих работу.
2) Установка Node.js. Node.js позволяет запускать JavaScript-код вне браузера и часто используется для создания серверной части приложений, а также для управления инструментами разработки.
3) Установка пакетных менеджеров. Пакетные менеджеры облегчают процесс установки и обновления библиотек и инструментов, используемых в разработке. Помимо npm, существует еще один популярный пакетный менеджер — Yarn.

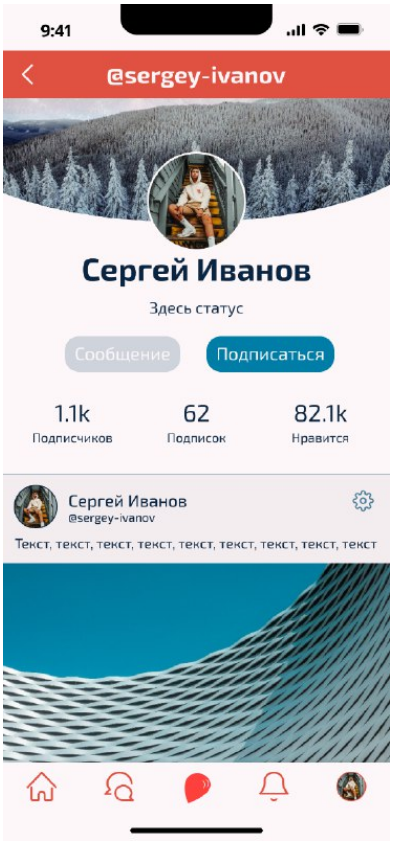

Рисунок 2 – Профиль пользователя

4) Создать файл PostCSS для управления плагинами.

```
touch postcss.config.js
```

```
module.exports = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  }
}
```

5) Настроить CSS-файл для использования Tailwind CSS и создать файл `src/styles.css` и добавить в него следующие строки:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

6) Настроить процесс сборки CSS, добавив соответствующие скрипты в package.json.

```
"scripts": {
  "build:css": "postcss src/styles.css -o public/styles.css",
  "watch:css": "postcss src/styles.css -o public/styles.css --watch"
}
```

Адаптиная верстка.

После выполнения настройки компонентов следовал этап верстки всех веб-страниц этого проекта. Разработка веб-страницы представляла собой создание двух файлов, в первом находилась индивидуальная разметка для страницы – все поля и элементы, а второй файл содержал стили для этой страницы, а также алгоритмы для полей, которые нуждались в обработке зачений пользователей, например алгоритмы ввода электронной почты или личной информации.

В качетсве примера выполнения работы по созданию веб-представления проекта ниже будут представлены скриншоты страниц веб-приложения.(см. рисунки 3, 4).

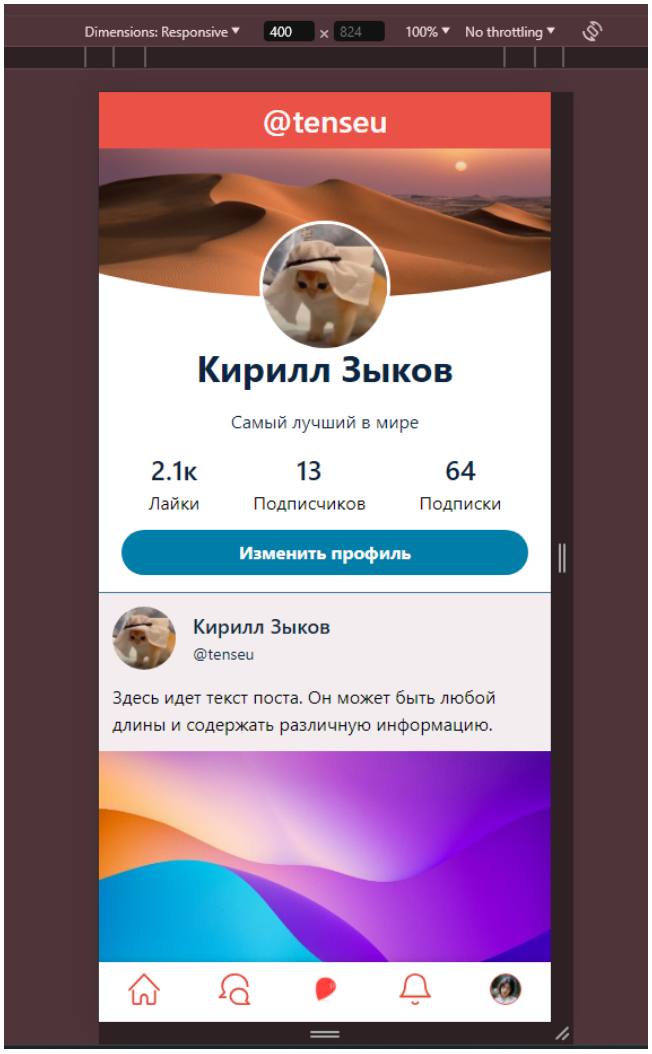

Рисунок 3 – Профиль пользователя в браузере

Масштабирование проекта.

Процесс портирования веб-приложения в десктопное приложение с использованием Electron включает в себя несколько ключевых шагов. Вот подробное описание каждого из этих шагов:

1) Установка Electron.

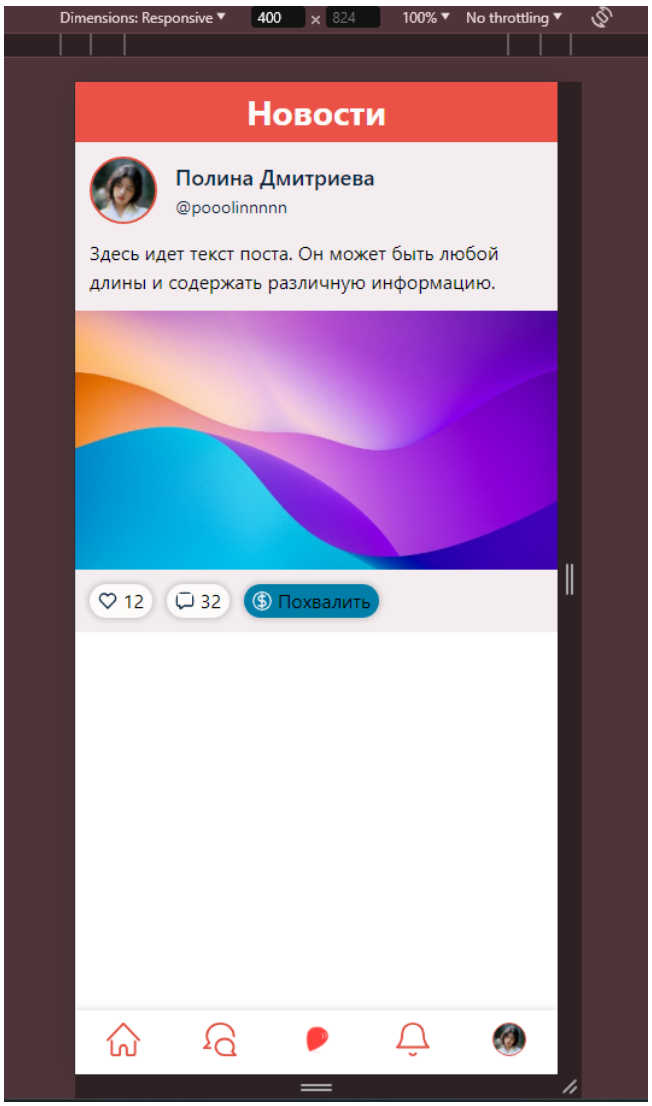

Рисунок 4 – Новостная запись в браузере

```
npm install electron --save-dev
```

Эта команда добавит Electron в зависимости проекта.

2) Создание файла конфигурации package.json.

Пример package.json:

```
{
  "name": "my-electron-app",
  "version": "1.0.0",
  "main": "main.js",
  "scripts": {
    "start": "electron ."
  },
  "devDependencies": {
    "electron": "^11.0.0"
  }
}
```

3) Настройка основного процесса (main process).

   В Electron основным процессом управляет файл JavaScript, например, main.js, который отвечает за создание окон и обработку событий на уровне приложения.

4) Веб-приложение обычно состоит из HTML, CSS и JavaScript файлов. Эти файлы необходимо перенести в директорию проекта Electron. Файл index.html будет загружен в окно, создаваемое в main.js. В html-код необходимо добавить `<script src="renderer.js"></script>` корректного отображения веб-страницы.

5) Для упаковки итогового приложения справляется `Electron-Packeger`. это инструмент командной строки, который позволяет упаковывать приложение Electron в исполняемые файлы для различных операционных систем (Windows, macOS, Linux). Этот инструмент автоматизирует процесс создания исполняемых файлов, включая все необходимые зависимости и ресурсы, такие как иконки и манифесты, чтобы приложение можно было запустить на целевой платформе без необходимости установки Node.js или Electron.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта мы разработали современное веб-приложение, которое сочетает в себе функционал социальной сети и платформу для поддержки социально значимых проектов. Данный проект стал результатом участия в программе «Стартап как диплом», где наша команда поставила перед собой амбициозную цель – создать удобный, интуитивно понятный и масштабируемый ресурс для взаимодействия пользователей и обмена информацией.

Для достижения поставленной цели мы решили ряд ключевых задач. Во-первых, был выбран подход, который позволяет максимально эффективно создать веб-приложение, соответствующее современным требованиям. Во-вторых, был разработан и реализован современный дизайн интерфейса социальной сети, который обеспечивает удобство и привлекательность для пользователей. В-третьих, мы перенесли дизайн в веб-представление, создав функциональный и отзывчивый интерфейс, который легко адаптируется под различные устройства.

Особое внимание было уделено отзывчивым реакциям на действия пользователей, что значительно улучшило пользовательский опыт. Масштабируемость веб-приложения обеспечила его доступность на различных устройствах, от настольных компьютеров до мобильных гаджетов, что особенно важно в современном мире, где пользователи взаимодействуют с интернетом через множество различных платформ.

В результате, наш проект достиг всех поставленных целей и задач, представив современное, безопасное и удобное веб-приложение, способное удовлетворить потребности широкой аудитории. Данный ресурс не только способствует эффективному взаимодействию пользователей и обмену информацией, но и поддерживает социально значимые инициативы, предоставляя платформу для их реализации и продвижения.