

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Новосибирский государственный университет экономики и управления «НИНХ»**

**(ФГБОУ ВО «НГУЭУ», НГУЭУ)**

Кафедра прикладной информатики

**ОТЧЕТ О ПРОХОЖДЕНИИ**

**ПРОИЗВОДСТВЕННОЙ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ) ПРАКТИКИ**

Профессиональный модуль: ПМ.05 Проектирование и разработка информационных систем

Специальность СПО: 09.02.07 Информационные системы и программирование

Место прохождения практики: ИП Метленко Н.Ф. г. Новосибирск, ул. Мичурина 12

Выполнил

Студент гр. 11-ИС302 28.06.2025 И.Е. Метленко

номер группы дата И.О. Фамилия

Ответственное лицо

профильной организации

директор 28.06.2025 Н.Ф. Метленко

должность дата И.О. Фамилия

Руководитель практики

от образовательной организации

доцент 28.06.2025 С.Н. Терещенко

должность, категория, ученая степень, ученое звание дата И.О. Фамилия

Новосибирск 2025

|  |
| --- |
|  |
| **МИНОБРНАУКИ РОССИИ** |
| **федеральное государственное бюджетное образовательное учреждение** |
| **высшего образования** |
| **«Новосибирский государственный университет экономики и управления «НИНХ»** |
| **(ФГБОУ ВО «НГУЭУ», НГУЭУ)** |
|  |
| Кафедра прикладной информатики |

|  |
| --- |
| **ЗАДАНИЕ НА ПРАКТИКУ** |
|  |
| Вид практики: ПРОИЗВОДСТВЕННАЯ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)  Профессиональный модуль: ПМ.05 Проектирование и разработка информационных систем |
| Специальность СПО: 09.02.07 Информационные системы и программирование |
| Выдано студенту 2 курса 11-ИС302 группы |
| Метленко Иван Евгеньевич |
| Место прохождения производственной практики: г. Новосибирск, ул. Мичурина 12  Сроки прохождения практики: с 26.05.2025-28.06.2025г. |
| В период прохождения практики необходимо:  1.Перед началом работы ознакомиться с инструкциями профильной организации по правилам противопожарной безопасности, правилам охраны труда, техники безопасности, требованиями по соблюдению санитарно-эпидемиологических правил и гигиенических нормативов, планами эвакуации при возникновении пожара, местами расположения санитарно-бытовых помещений, медицинских кабинетов. Не приступать к работе при отсутствии информации о результатах аттестации условий труда на данном рабочем месте или при наличии информации о несоответствии параметров данного оборудования требованиям санитарных норм. |
| 2.Выполнять любые виды работ только по поручению и после соответствующего инструктажа ответственного лица профильной организации. |
| 3.Отражать этапы и виды выполненных работ в Дневнике прохождения практики. |
| 4.Составить, оформить и защитить Отчет о прохождении практики. |
| 5.Выполнить следующие этапы и виды работ: |
| |  |  |  | | --- | --- | --- | | № | Виды работ на практике | Количество часов на выполнение задания | | 1 | Подготовительный этап: | 4 | | 2 | Основной этап:  Осуществлять постановку задачи по обработке информации. Выполнять анализ предметной  области.  Использовать алгоритмы обработки информации для различных приложений. Работать с  инструментальными средствами обработки информации.  Осуществлять выбор модели построения информационной системы.  Осуществлять выбор модели и средства построения информационной системы и  программных средств  Осуществлять математическую и информационную постановку задач по обработке  информации.  Использовать алгоритмы обработки информации для различных приложений.  Создавать и управлять проектом по разработке приложения и формулировать его задачи.  Использовать языки структурного, объектно-ориентированного программирования и язык  сценариев для создания независимых программ.  Разрабатывать графический интерфейс приложения.  Использовать языки структурного, объектно-ориентированного программирования и язык  сценариев для создания независимых программ.  Решать прикладные вопросы программирования и языка сценариев для создания программ.  Проектировать и разрабатывать систему по заданным требованиям и спецификациям.  161  Разрабатывать графический интерфейс приложения.  Создавать проект по разработке приложения и формулировать его задачи. Использовать  методы тестирования в соответствии с техническим заданием.  Разрабатывать проектную документацию на эксплуатацию информационной системы.  Использовать стандарты при оформлении программной документации.  Использовать методы и критерии оценивания предметной области и методы определения  стратегии развития бизнес-процессов организации.  Решать прикладные вопросы интеллектуальных систем с использованием статических  экспертных систем, экспертных систем реального времени. | 154 | | 3 | Этап обработки и анализа информации:  Анализ полученной информации с целью выработки  рекомендаций по совершенствованию деятельности организации.  Обработка и систематизация практического материала необходимого для написания отчета. | 10 | | 4 | Этап подготовки отчета: | 12 | | 5 | Всего | 180 | |
| Задание на практику выдано  Дата «26» мая 2025 года. |
| Руководитель практики от образовательной организации: преподаватель СПО С.Н. Терещенко |
| Задание на практику получил. |
| Студент: И.Е. Метленко |
|  |
|  |



**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Новосибирский государственный университет экономики и управления «НИНХ»**

**(ФГБОУ ВО «НГУЭУ», НГУЭУ)**

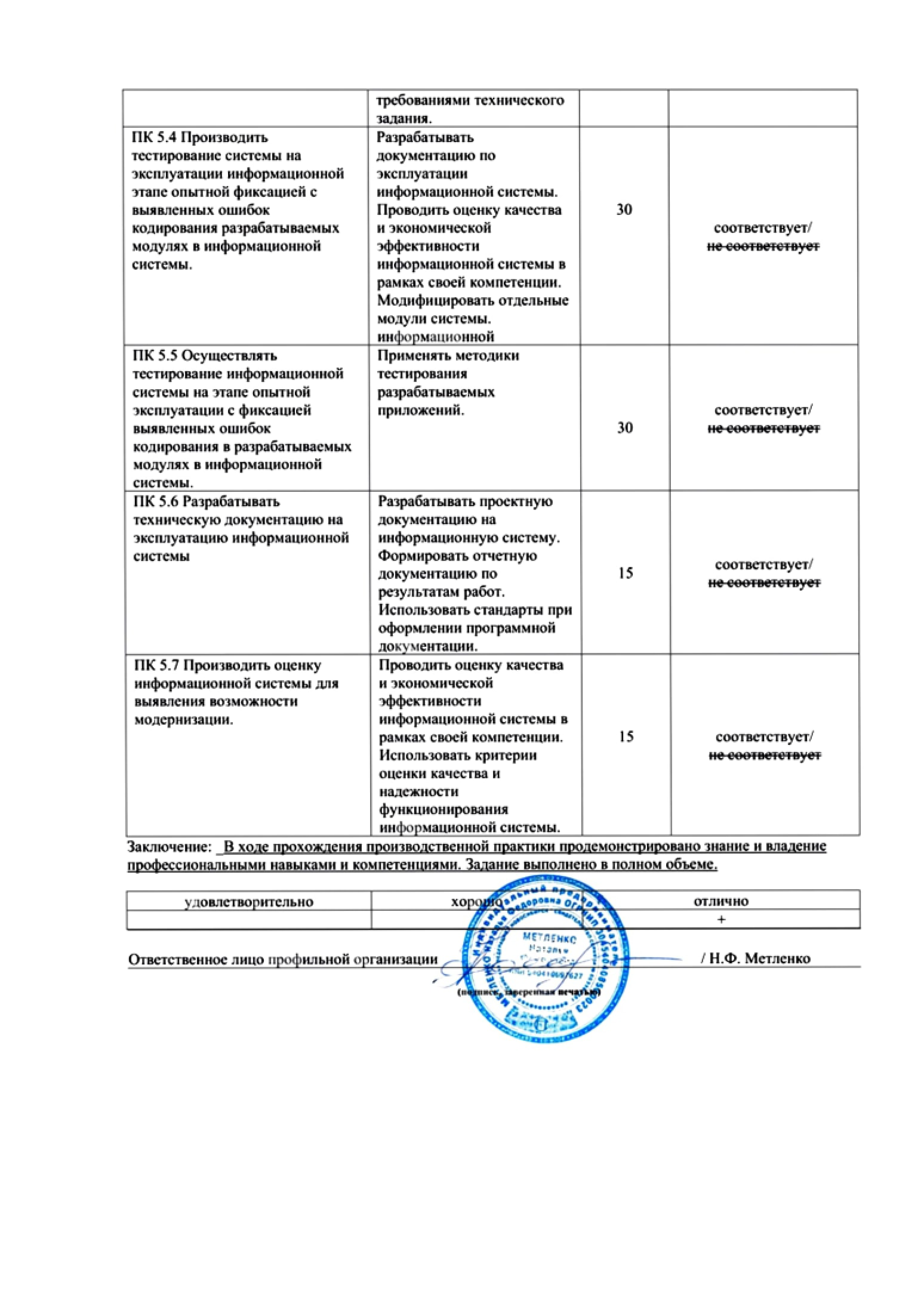
Кафедра прикладной информатики

|  |
| --- |
| **ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ** |
| Вид практики: ПРОИЗВОДСТВЕННАЯ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ) |
| Профессиональный модуль: ПМ.05 Проектирование и разработка информационных систем |
| Специальность СПО: 09.02.07 Информационные системы и программирование  Студент 2 курса 11-ИС302 группы Метленко Иван Евгеньевич |
| Место прохождения производственной (по профилю специальности) практики: г. Новосибирск, ул. Мичурина 12 |
|  |
| Сроки прохождения практики: с 26.05.2025-28.06.2025г.  Руководитель производственной (по профилю специальности) практики от  образовательной организации: |
| Преподаватель СПО Терещенко С.Н. |
|  |

|  |  |
| --- | --- |
| Дата/Период | Краткое описание выполненных работ |
| 1 | 2 |
| 26.05.2025 | Ознакомление с целями практики. Анализ требований к сайту-портфолио и целевой аудитории. |
| 27.05.2025 | Изучение библиотеки React, документации по JSX, компонентному подходу и виртуальному DOM. |
| 28.05.2025 | Настройка проекта: установка Vite, инициализация React-приложения, настройка репозитория Git. |
| 29.05.2025 | Проектирование структуры сайта: определение разделов, навигации и пользовательских сценариев. |
| 30.05.2025 | Разработка wireframe-макета сайта в Figma: создание базовой структуры и разметки экранов. |
| 31.05.2025 | Дизайн десктопной версии: выбор цветовой палитры, типографики, стилистики интерфейса. |
| 01.06.2025 | **Выходной** |
| 02.06.2025 | Создание финального прототипа в Figma: шапка, приветственный блок, раздел «Обо мне». |
| 03.06.2025 | Дизайн экранов технического стека и проектов. Подготовка мобильной версии прототипа. |
| 04.06.2025 | Верстка главной страницы в React: компоненты Header, Home, настройка базовой навигации. |
| 05.06.2025 | Импорт глобальных стилей, написание адаптивной верстки для главного экрана. |
| 06.06.2025 | Реализация страницы «Обо мне» и «Навыки»: создание компонентов карточек и сетки. |
| 07.06.2025 | Подключение логотипов технологий, оформление карточек с применением CSS Grid и Flexbox. |
| 08.06.2025 | **Выходной** |
| 09.06.2025 | Создание раздела «Проекты»: проектирование компонента ProjectCard и шаблонов данных. |
| 10.06.2025 | Добавление интерактивности карточкам проектов: hover-эффекты, динамическая генерация. |
| 11.06.2025 | Реализация маршрутизации через React Router, подключение всех страниц к навигации. |
| 12.06.2025 | Разработка страницы «Контакты»: создание формы с полями и базовой валидацией. |
| 13.06.2025 | Интеграция формы с EmailJS API: настройка шаблона письма, тестирование отправки. |
| 14.06.2025 | Оформление визуальной обратной связи: сообщения об успехе/ошибке, индикаторы загрузки. |
| 15.06.2025 | **Выходной** |
| 16.06.2025 | Оптимизация компонентов: использование React.memo, useCallback, настройка lazy loading. |
| 17.06.2025 | Тестирование адаптивности сайта на различных разрешениях, настройка медиа-запросов. |
| 18.06.2025 | Проверка кроссбраузерности: тесты в Chrome, Firefox, Safari, Edge. |
| 19.06.2025 | Повышение доступности: настройка aria-атрибутов, семантической структуры, alt-тексты. |
| 20.06.2025 | Подключение favicon, настройка SEO-мета-тегов, финальная сборка проекта. |
| 21.06.2025 | Проведение комплексного тестирования, устранение багов, финальная шлифовка UI. |
| 22.06.2025 | **Выходной** |
| 23.06.2025 | Подготовка презентационных материалов: скриншоты, описание компонентов, структура проекта. |
| 24.06.2025 | Оформление отчёта по практике: разделы по этапам работы, код, диаграммы, выводы. |
| 25.06.2025 | Корректура текста, компоновка приложения, проверка соответствия требованиям. |
| 26.06.2025 | Завершение оформления, выгрузка проекта на GitHub, подготовка к защите. |
| 27.06.2025 | Завершение оформления отчёта, корректура текста. Подготовка к защите и собеседованию. |
| 28.06.2025 | Собеседование по итогам прохождения практики |

|  |
| --- |
|  |
| Студент И.Е. Метленко |
|  |
|  |
|  |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **МИНОБРНАУКИ РОССИИ** | | | |
| **федеральное государственное бюджетное образовательное учреждение** | | | |
| **высшего образования** | | | |
| **«Новосибирский государственный университет экономики и управления «НИНХ»** | | | |
| **(ФГБОУ ВО «НГУЭУ», НГУЭУ)** | | | |
|  | | | |
| Кафедра прикладной информатики | | | |
|  | | | |
| **АТТЕСТАЦИОННЫЙ ЛИСТ** | | | |
| **ПО ПРОИЗВОДСТВЕННОЙ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ) ПРАКТИКЕ** | | | |
| Метленко Иван Евгеньевич | | | |
| обучающийся на 2 курсе 11-ИС302 группы | | | |
| по специальности СПО: 09.02.07 Информационные системы и программирование | | | |
| прошел производственную практику по профессиональному модулю:  09.02.07 ПМ.05 Проектирование и разработка информационных систем в объеме 180 часов с 26.05.2025-28.06.2025г в организации: ИП Метленко Н.Ф. г. Новосибирск, ул. Мичурина 12ФГБОУ ВО «НГУЭУ», кафедра прикладной информатики, г. Новосибирск, ул. Каменская, 52/1, каб.5-406 | | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | В период прохождения практики студент соблюдал принятые в организации правила внутреннего трудового распорядка; активно участвовал в процессах деятельности организации.  Содержание и объем работ, указанные студентом в дневнике практике, полностью соответствуют фактически выполненным | | | | | | За время прохождения производственной практики студент продемонстрировал освоение (развитие) общих компетенций на уровне: | | | | | | Код  общей компетенции | Содержание общей компетенции | Уровень освоения общей компетенции | | | | низкий | средний | высокий | | ОК 1. | Выбирать способы решения задач  профессиональной деятельности,  применительно к различным контекстам |  |  | + | | ОК 2. | Осуществлять поиск, анализ и  интерпретацию информации, необходимой  для выполнения задач профессиональной  деятельности |  |  | + | | ОК 3. | Планировать и реализовывать собственное  профессиональное и личностное развитие |  |  | + | | ОК 4. | Работать в коллективе и команде,  эффективно взаимодействовать с коллегами,  руководством, клиентами. |  |  | + | | ОК 5. | Осуществлять устную и письменную  коммуникацию на государственном языке с  учетом особенностей социального и  культурного контекста. |  |  | + | | ОК 6. | Проявлять гражданско-патриотическую  позицию, демонстрировать осознанное |  |  | + | | ОК 7. | Содействовать сохранению окружающей  среды, ресурсосбережению, эффективно  действовать в чрезвычайных ситуациях. |  |  | + | | ОК 8. | Использовать средства физической  культуры для сохранения и укрепления  здоровья в процессе профессиональной  деятельности и поддержания необходимого  уровня физической подготовленности |  |  | + | | ОК 9. | Использовать информационные технологии  в профессиональной деятельности. |  |  | + | | ОК 10. | Пользоваться профессиональной  документацией на государственном и  иностранном языке. |  |  | + | | ОК 11. | Планировать предпринимательскую  деятельность в профессиональной сфере |  |  | + |   Оценка освоения профессиональных компетенций по основному виду деятельности: Разработка дизайна веб-приложений | | | |
| Профессиональные компетенции | Виды работ, выполненные во время производственной (по профилю специальности) практики | Объем работ, часов | Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика |
| ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему. | Анализировать предметную область. Использовать инструментальные средства обработки информации. Обеспечивать сбор данных для анализа использования и функционирования информационной системы. Определять состав оборудования и программных средств разработки информационной системы. Выполнять работы предпроектной стадии. | 30 | соответствует/  ~~не соответствует~~ |
| ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика. | Разрабатывать проектную документацию на информационную систему. | 30 | соответствует/  ~~не соответствует~~ |
| ПК 5.3. Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием. | Управлять процессом разработки приложений с использованием инструментальных средств. Модифицировать отдельные модули информационной системы. Программировать в соответствии с требованиями технического задания. | 30 | соответствует/  ~~не соответствует~~ |



**СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ 9](#_Toc201896864)

[ВВЕДЕНИЕ 10](#_Toc201896865)

[1 Выбор технологии и обоснование 11](#_Toc201896866)

[1.1 Особенности библиотеки React и её преимущества 11](#_Toc201896867)

[1.2 Выбор React и CSS как основных технологий реализации 15](#_Toc201896868)

[2 Разработка портфолио веб-разработчика 9](#_Toc201896869)

[2.1 Определение целей проекта и целевой аудитории 9](#_Toc201896870)

[2.2 Разработка прототипа в Figma 10](#_Toc201896871)

[2.3 Обзор основных страниц сайта 11](#_Toc201896872)

[3 Реализация проекта на React 19](#_Toc201896873)

[3.1 Архитектура проекта и структура компонентов 19](#_Toc201896874)

[3.2 Реализация компонентов и взаимодействие с API 20](#_Toc201896875)

[3.3 Тестирование и демонстрация функциональности 26](#_Toc201896876)

[ЗАКЛЮЧЕНИЕ 39](#_Toc201896877)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 40](#_Toc201896878)

[ПРИЛОЖЕНИЕ A 42](#_Toc201896879)

[ПРИЛОЖЕНИЕ Б 45](#_Toc201896880)

[ПРИЛОЖЕНИЕ В 46](#_Toc201896881)

**ВВЕДЕНИЕ**

Современный рынок труда требует от начинающих специалистов не только наличия профессиональных навыков, но и умения презентовать свои работы. Одним из эффективных способов демонстрации компетенций веб-разработчика является личное портфолио в виде сайта. Такой ресурс позволяет наглядно представить собственные проекты, рассказать о себе и привлечь внимание потенциальных заказчиков или работодателей.

Производственная практика была направлена на формирование и развитие профессиональных компетенций в области проектирования и разработки веб-проектов. В рамках практики мною был реализован индивидуальный проект — сайт-портфолио веб-разработчика. Проект включал в себя все основные этапы жизненного цикла веб-приложения: от формирования идеи и прототипирования до программной реализации, тестирования и документирования.

В качестве инструмента проектирования был использован сервис Figma, позволяющий создавать макеты интерфейсов с учетом адаптивности под различные разрешения экранов. Для реализации сайта был выбран фреймворк React — современная библиотека для создания пользовательских интерфейсов на языке JavaScript, а стилизация осуществлялась с помощью обычного CSS.

1. Выбор технологии и обоснование

# Особенности библиотеки React и её преимущества

React представляет собой одну из наиболее влиятельных и популярных JavaScript-библиотек современной веб-разработки. Созданная командой Facebook в 2013 году, эта библиотека с открытым исходным кодом трансформировала подходы к созданию пользовательских интерфейсов и продолжает оставаться доминирующей силой в экосистеме фронтенд-разработки.

React кардинально отличается от традиционных подходов к веб-разработке своим декларативным характером. В то время как императивное программирование требует детального описания каждого шага для достижения результата, декларативный подход React позволяет разработчикам сосредоточиться на описании конечного состояния интерфейса. Библиотека самостоятельно определяет необходимые изменения и обновляет интерфейс соответствующим образом. Это значительно упрощает процесс разработки, поскольку программисту достаточно один раз подробно описать, как должен выглядеть и вести себя интерфейс в различных состояниях, не фокусируясь на способе достижения этого результата.

Центральной концепцией React являются компоненты — независимые, переиспользуемые блоки кода, каждый из которых представляет отдельную часть пользовательского интерфейса. Каждый компонент инкапсулирует свою логику, состояние и представление, что делает их полностью самостоятельными элементами. Такая архитектура обеспечивает высокую модульность кода и позволяет легко переиспользовать компоненты в различных частях приложения или даже в других проектах.

Виртуальный DOM представляет ещё одну ключевую особенность React. Это легковесная копия реального DOM-дерева, которая хранится в памяти и позволяет React эффективно отслеживать изменения. Когда происходят изменения в данных или состоянии компонентов, React сначала обновляет виртуальный DOM, затем сравнивает его с предыдущим состоянием и вычисляет минимальный набор операций, необходимых для обновления реального DOM. Этот процесс называется reconciliation и обеспечивает высокую производительность приложений, особенно при работе с частыми изменениями интерфейса.

JSX — специальный XML-подобный синтаксис, который является неотъемлемой частью React. JSX позволяет писать HTML-разметку непосредственно в JavaScript-коде, что делает компоненты более читаемыми и удобными для разработки. Несмотря на внешнее сходство с HTML, JSX представляет собой расширение синтаксиса JavaScript, которое во время компиляции транслируется в обычные вызовы функций React.createElement.

Преимущества React. Производительность React обеспечивается не только виртуальным DOM, но и эффективными алгоритмами обновления. Библиотека способна группировать множественные обновления и применять их одновременно, что значительно снижает нагрузку на браузер. Система управления состоянием React гарантирует повторное отображение только тех компонентов, которые действительно нуждаются в обновлении, что существенно экономит вычислительные ресурсы.

Гибкость React проявляется в том, что это библиотека, а не полноценный фреймворк. React не навязывает жёстких правил организации проекта и позволяет разработчикам самостоятельно выбирать дополнительные инструменты для роутинга, управления состоянием, серверного рендеринга и других задач. Эта свобода выбора делает React универсальным инструментом, подходящим для проектов различного масштаба и сложности.

Экосистема React поражает своим богатством и разнообразием. Вокруг библиотеки сформировалось множество дополнительных инструментов, библиотек и фреймворков, которые расширяют её возможности и упрощают решение типовых задач. React может использоваться не только для веб-разработки, но и для создания мобильных приложений через React Native, настольных приложений и даже VR-приложений.

Сообщество React представляет одно из самых активных и многочисленных в мире веб-разработки. По данным статистики Stack Overflow 2024 года, около 39.5% разработчиков используют React, что делает его второй по популярности веб-технологией после Node.js. Большое сообщество означает обширную документацию, множество обучающих материалов, готовых решений и активную поддержку.

Простота изучения и использования React достигается благодаря чёткой концептуальной модели и хорошо структурированной документации. Новички могут быстро освоить основы библиотеки и начать создавать полноценные приложения. Декларативный подход и компонентная архитектура делают код более предсказуемым и понятным, что упрощает как разработку, так и отладку.

Масштабируемость React проявляется в способности библиотеки эффективно работать как в небольших проектах, так и в крупных корпоративных приложениях. Компонентная архитектура позволяет легко добавлять новую функциональность без нарушения существующего кода, а модульная структура облегчает работу больших команд разработчиков над одним проектом.

Недостатки и ограничения. Несмотря на многочисленные преимущества, React имеет определённые недостатки. Порог входа для новичков может показаться средним по сложности. Изучение React требует не только понимания синтаксиса библиотеки, но и освоения новой парадигмы мышления, включая JSX, концепции состояния и жизненного цикла компонентов. Особенно сложным может оказаться освоение JSX для разработчиков, привыкших к разделению HTML и JavaScript.

Быстрое развитие React и частые обновления могут создавать проблемы совместимости версий. При обновлении на новую версию библиотеки могут возникнуть конфликты с существующим кодом, что требует дополнительных усилий по поддержке проектов. Это особенно критично для крупных приложений, где миграция на новую версию может потребовать значительных временных и ресурсных затрат.

Управление состоянием в сложных приложениях представляет ещё одну потенциальную сложность. Хотя React предоставляет встроенные механизмы управления состоянием, в больших проектах часто требуются дополнительные библиотеки типа Redux или MobX. Освоение и правильная реализация этих инструментов может потребовать дополнительного времени и усилий от разработчиков.

Статистика использования React впечатляет своими масштабами. По данным 2024 года, React используется на более чем 1.3 миллионах веб-сайтов, а еженедельное количество загрузок пакета React превышает 20 миллионов, что в несколько раз больше, чем у ближайших конкурентов Angular и Vue.js. GitHub-репозиторий React имеет более 228,000 звёзд и 46,000 форков, что свидетельствует об активном участии сообщества в развитии библиотеки.

React применяется в различных сферах деятельности, от электронной коммерции и финансов до образования и развлечений. Крупнейшие технологические компании, включая Facebook, Netflix, Instagram и многие другие, используют React для создания своих продуктов. Подавляющее большинство разработчиков применяют React в профессиональной деятельности, создавая одностраничные приложения, статические сайты, системы дизайна и гибридные приложения.

React продолжает активно развиваться, и выход версии React 19 в 2024 году принёс множество улучшений в области производительности и удобства использования. Новые возможности, включая React Compiler и улучшенную работу с формами, направлены на достижение высокого пользовательского опыта при сохранении простоты разработки.

Таким образом, React представляет собой зрелую, мощную и широко поддерживаемую библиотеку, которая продолжает оставаться одним из лучших выборов для современной веб-разработки. Её декларативный подход, компонентная архитектура, виртуальный DOM и богатая экосистема делают React превосходным инструментом для создания современных, производительных и масштабируемых пользовательских интерфейсов.

# Выбор React и CSS как основных технологий реализации

React и CSS часто выбираются как базовые технологии для создания современных веб-приложений благодаря их универсальности, популярности и поддержке со стороны сообщества.

Здесь в таблице 1 представлены популярные библиотеки и технологии для реализации пользовательских интерфейсов, в которой React и CSS получают преимущество по суммарным баллам.

Таблица 1 — Сравнительная таблица популярных библиотек и технологий для реализации пользовательских интерфейсов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Технология / Библиотека | Производительность | Масштабируемость | Простота обучения | Экосистема и поддержка | Гибкость и кастомизация | Стилизация | Общий балл |
| **1** | 2 | 3 | 4 | 5 | 6 | 7 | **8** |
| **React + CSS (CSS Modules / CSS-in-JS)** | 9 | 10 | 8 | 10 | 10 | 9 | **56** |
| Angular (TypeScript, встроенный CSS) | 8 | 9 | 6 | 9 | 7 | 7 | 46 |
| Vue.js + CSS | 8 | 8 | 9 | 8 | 8 | 8 | 49 |
| Material UI (React, CSS-in-JS) | 8 | 9 | 7 | 10 | 8 | 8 | 50 |
| Chakra UI (React, Emotion CSS-in-JS) | 7 | 8 | 8 | 8 | 9 | 7 | 47 |
| Tailwind CSS (Utility-first CSS) | 9 | 9 | 7 | 7 | 8 | 10 | 50 |

Таким образом, React в сочетании с CSS (особенно с современными подходами CSS Modules или CSS-in-JS) выигрывает по сумме ключевых параметров, обеспечивая высокую производительность, масштабируемость, гибкость и удобство разработки, что делает их оптимальным выбором для реализации современных веб-приложений.

1. Разработка портфолио веб-разработчика

# Определение целей проекта и целевой аудитории

Основной целью разработки сайта-портфолио является создание современного и функционального веб-ресурса, позволяющего представить себя как веб-разработчика, продемонстрировать выполненные проекты, навыки и используемые технологии. Согласно техническому заданию из Приложение А, сайт должен быть визуально привлекательным, удобным для навигации, а также адаптивным — корректно отображающимся на различных устройствах и экранах.

Задачи, решаемые в рамках проекта:

* создание структуры сайта с логичной навигацией;
* разработка прототипа и пользовательского интерфейса в Figma;
* реализация сайта с использованием фреймворка React и обычного CSS;
* обеспечение адаптивной верстки для десктопной и мобильной версий;
* размещение информации о себе, проектах и контактных данных.

Проект также преследует образовательную цель — закрепление на практике знаний и навыков, полученных в процессе обучения, и развитие компетенций в сфере веб-разработки.

Целевая аудитория сайта — это потенциальные работодатели, заказчики, а также коллеги по профессиональному сообществу, которым может быть интересен опыт и уровень подготовки разработчика.

В первую очередь сайт ориентирован на:

* HR-специалистов и рекрутеров IT-компаний, осуществляющих поиск кандидатов на позиции junior- или trainee-разработчиков;
* заказчиков, которым необходимы услуги по созданию сайтов, разработке интерфейсов и пользовательских решений;
* преподавателей и наставников, оценивающих уровень проектной деятельности;
* других веб-разработчиков, заинтересованных в обмене опытом и вдохновении.

Сайт должен быть интуитивно понятным, содержать только важную и структурированную информацию, быстро загружаться и быть доступным с разных устройств и браузеров.

# Разработка прототипа в Figma

Проектирование интерфейса сайта-портфолио началось с создания wireframe-макета, в котором была обозначена общая структура будущего сайта: шапка с логотипом и навигацией, блок приветствия, технический стек и проекты. Это позволило определить логическую последовательность элементов на странице и обеспечить удобную навигацию для пользователя.

После утверждения структуры был разработан финальный прототип в Figma для двух разрешений — десктопной версии (1920×1080) и мобильной (400×728). Макет Figma можно увидеть, перейдя по ссылке в Приложение Б. При проектировании учитывались принципы современного UI/UX-дизайна: лаконичный стиль, единое цветовое оформление, читаемая типографика и логичная компоновка блоков.

Визуальный стиль был выдержан в светло-зелёной цветовой гамме с акцентами оранжевого цвета. В качестве шрифтов использовались простые и современные гарнитуры, обеспечивающие хорошую читаемость. Все интерфейсные элементы — кнопки, карточки проектов, форма — оформлены в едином стиле с закруглениями и тенями для придания объёма.

В макете на рисунке 1 и в Приложение Б были предусмотрены четыре основных экранов:

* Desktop 1: информация «Обо мне»;
* Desktop 2: блок с техническим стеком — логотипы используемых технологий;
* Desktop 3: галерея проектов — карточки с названиями и описанием;
* Desktop 4: форма обратной связи и контактный e-mail.

Каждый из этих экранов адаптирован под мобильные устройства: изменена компоновка блоков, уменьшены размеры шрифтов, реализовано выпадающее меню.

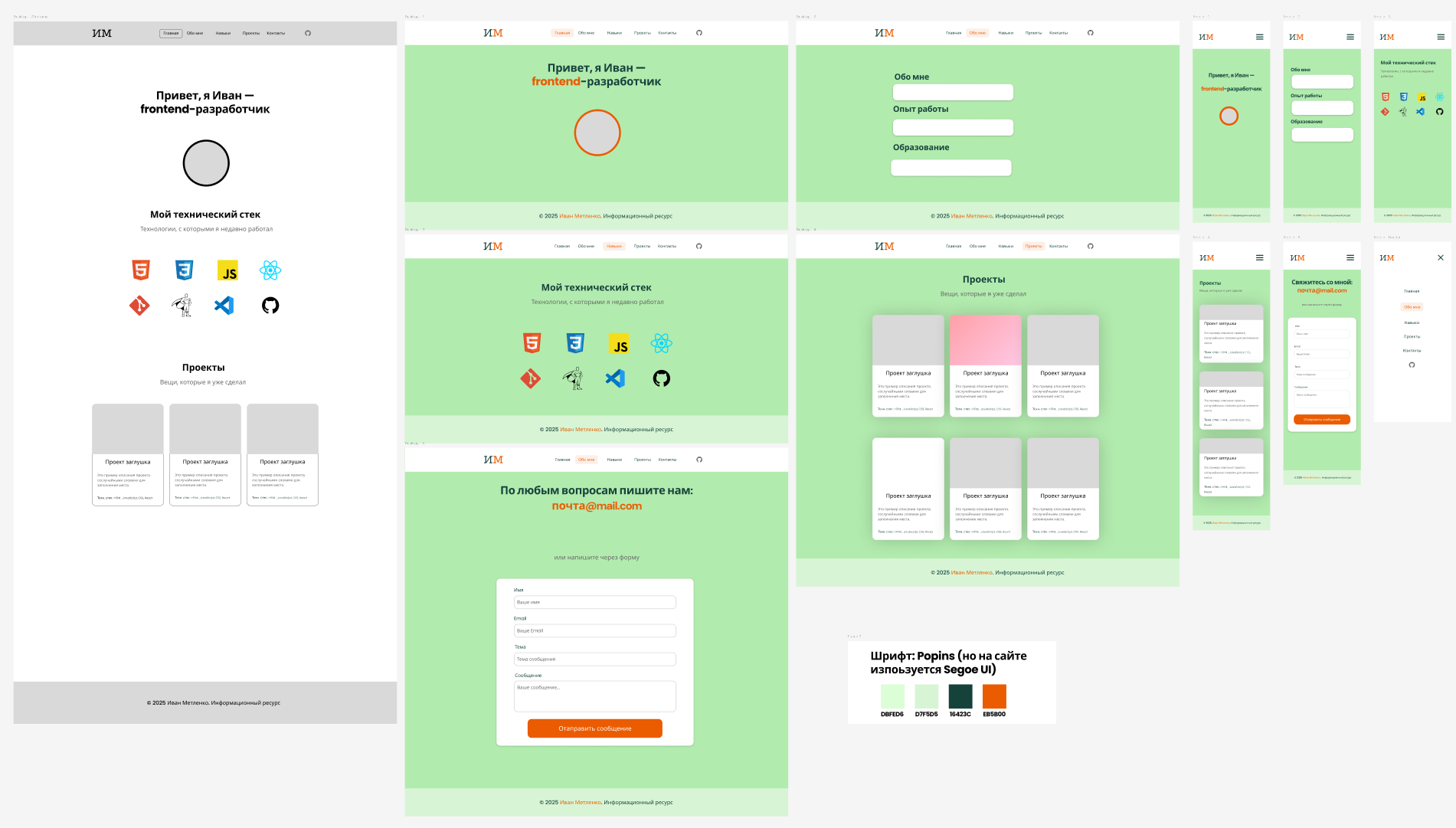


Рисунок 1 – Все макеты Figma

Готовый макет был экспортирован и использован в качестве основного источника при вёрстке сайта на React. Макет позволил сократить время разработки и обеспечить соответствие финального продукта изначально задуманной концепции.

# Обзор основных страниц сайта

Сайт-портфолио состоит из четырёх ключевых разделов, реализованных в виде отдельных экранов и логически связанных навигацией:

На главной странице рисунка 2 размещено приветствие и краткое описание — кто такой автор сайта и чем он занимается. Также здесь расположен логотип и верхнее навигационное меню, ссылающееся на другие разделы. Центральным элементом является аватар и фраза «Привет, я Иван — frontend-разработчик».

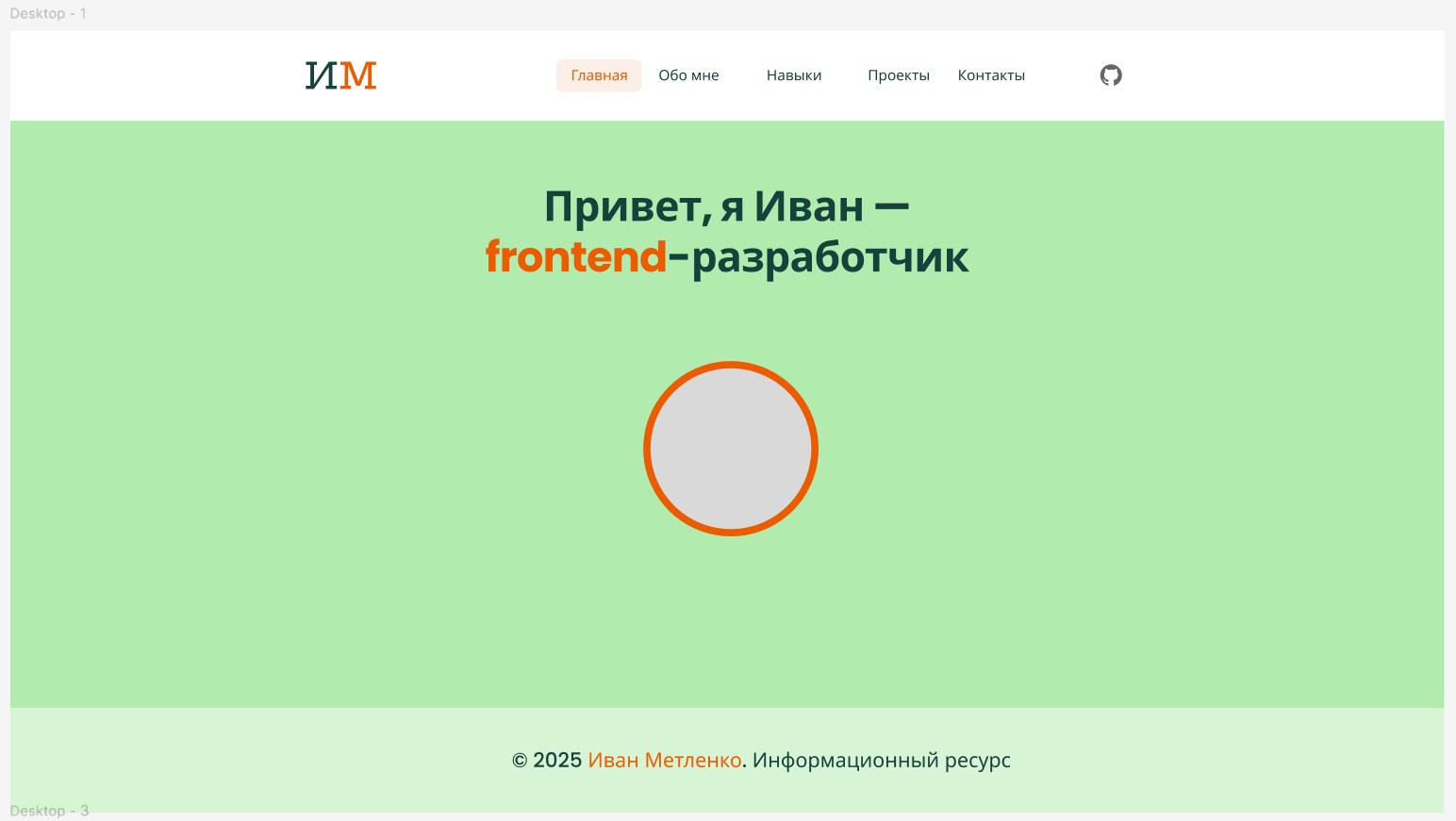


Рисунок 2 – Главная страница в Figma

В этом разделе на рисунке 3 размещена базовая информация об авторе: опыт работы, навыки и образование. Представление информации организовано в виде трёх карточек с заголовками и блоками для текста. Такой формат удобен для быстрого восприятия и хорошо адаптируется под мобильные устройства.

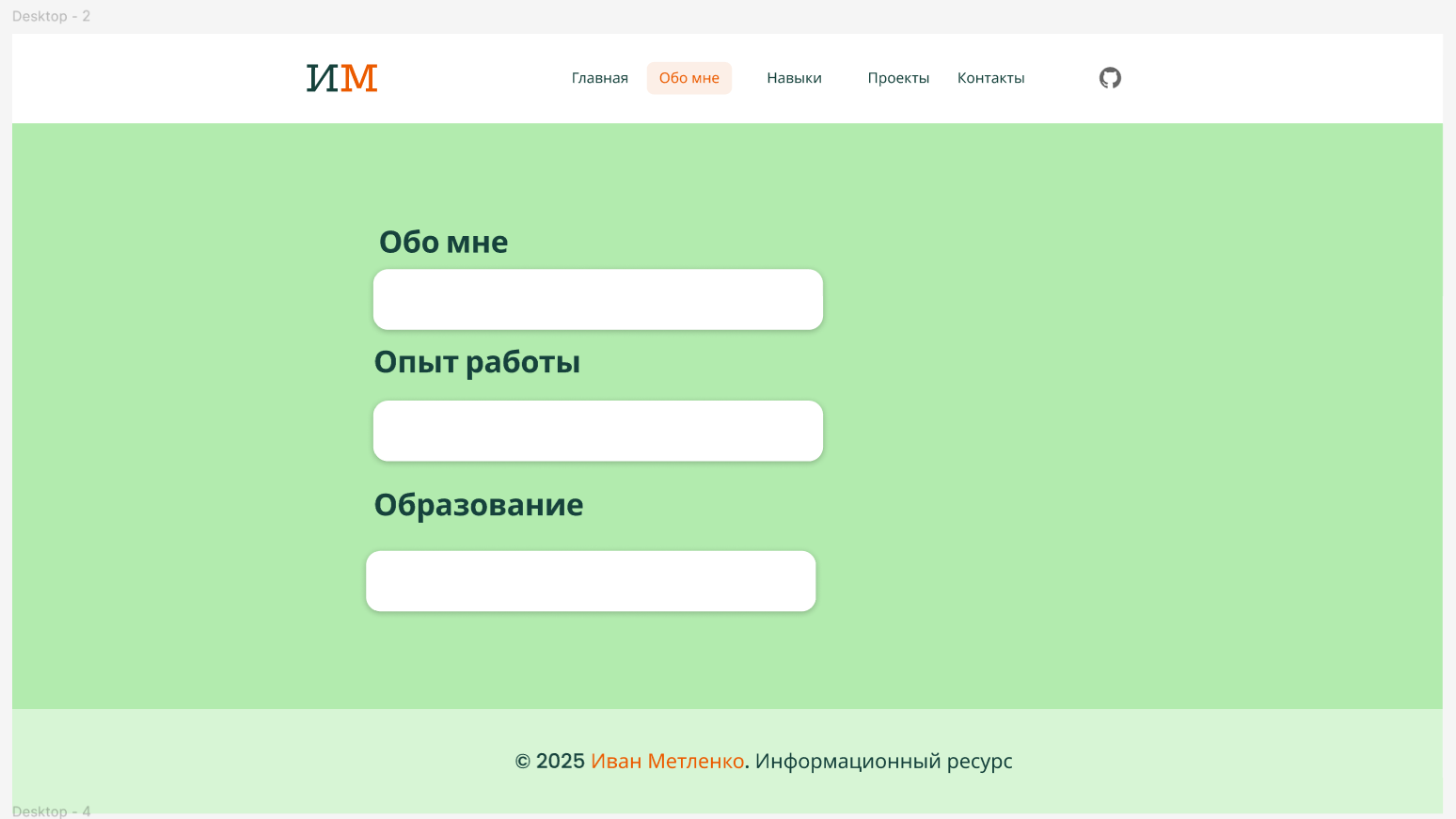


Рисунок 3 – Страница «Обо мне» в Figma

Отдельный блок посвящён технологиям на рисунке 4. Здесь представлены логотипы таких инструментов и языков, как HTML5, CSS3, JavaScript, React, Git, VS Code и других. Это позволяет потенциальному заказчику или работодателю сразу увидеть, с какими технологиями знаком разработчик.

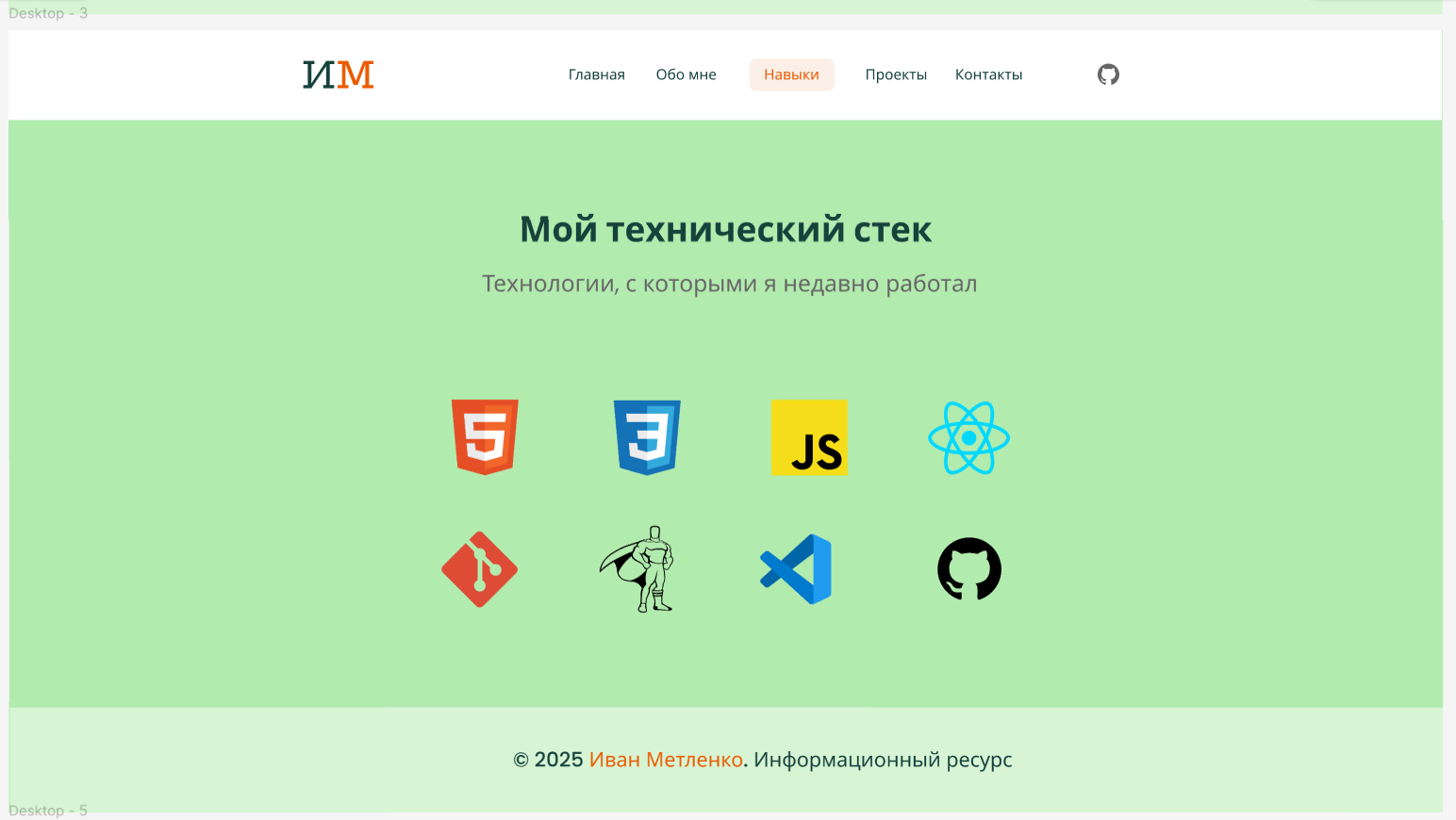


Рисунок 4 – Страница с техническим стеком в Figma

Карточки проектов расположены в сетке и содержат название, краткое описание и ссылку, показано на рисунке 5. В дизайне использованы акцентные цвета и лёгкие тени для выделения элементов. Хотя некоторые проекты ещё находятся в разработке, их оформление уже позволяет составить общее впечатление о структуре и стиле будущих работ.

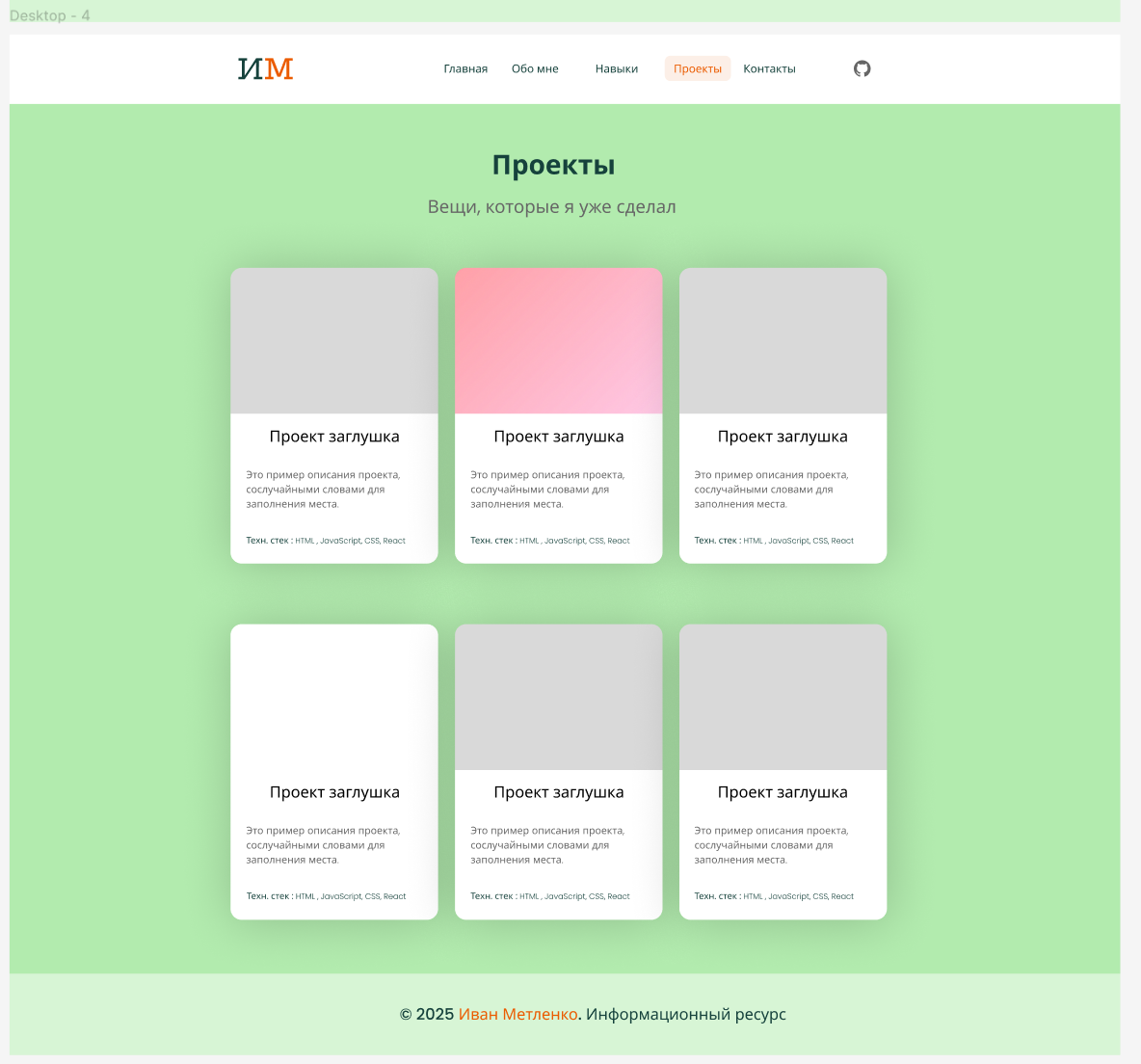


Рисунок 5 – Страница проектов в Figma

Здесь, на рисунке 6, размещена контактная информация (e-mail) и форма обратной связи, через которую можно отправить сообщение. Форма включает поля для имени, адреса электронной почты, темы и текста сообщения. Также добавлена кнопка отправки с визуальной обратной связью при наведении.

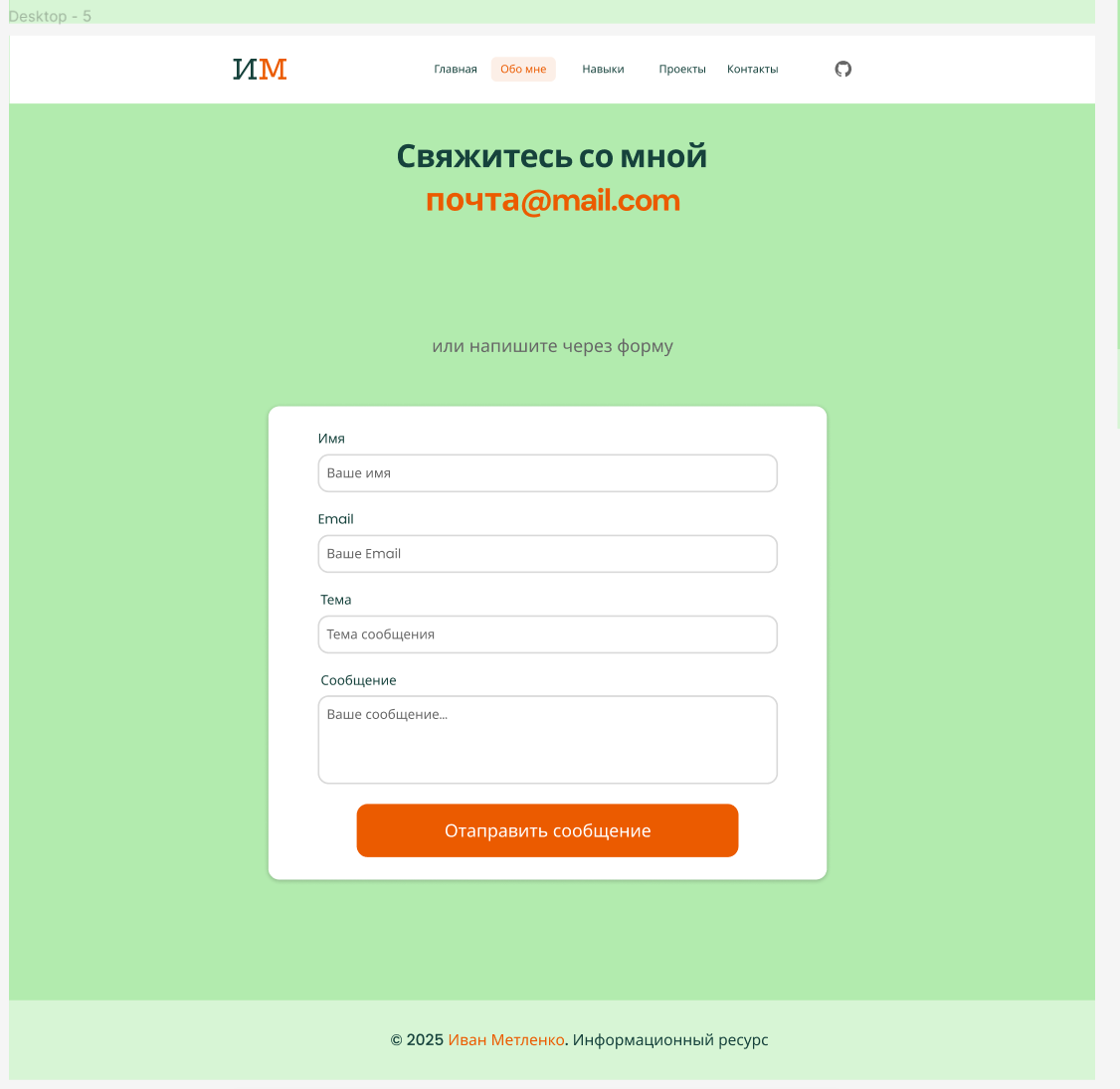


Рисунок 6 – Контактная страница в Figma

Каждая страница сайта выполнена в адаптивном формате: мобильные версии отображают ту же информацию, но в упрощённой вертикальной структуре и с выпадающим меню на рисунке 7.

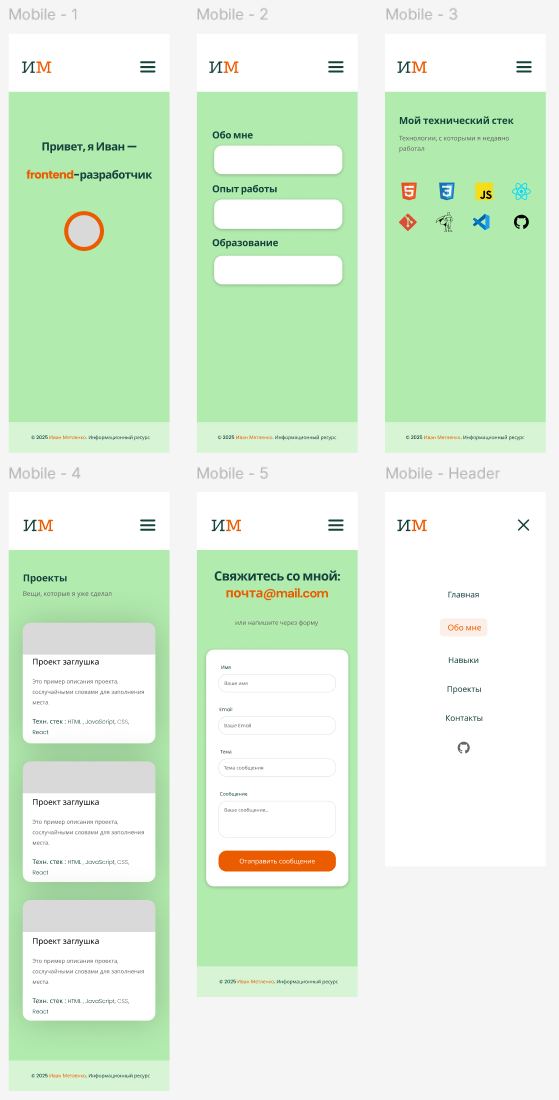


Рисунок 7 – Мобильная версия в Figma

Таким образом, структура сайта была спроектирована таким образом, чтобы пользователь мог легко получить всю необходимую информацию о разработчике: кто он, с какими технологиями работает, какие проекты реализовал и как с ним можно связаться. Благодаря лаконичному дизайну, акцентной цветовой палитре и чёткой визуальной иерархии, каждая страница сайта выполняет свою функцию и поддерживает общее цельное восприятие ресурса. Это делает сайт удобным и информативным как для потенциальных заказчиков, так и для HR-специалистов, заинтересованных в оценке профессионального уровня разработчика.

1. Реализация проекта на React

# Архитектура проекта и структура компонентов

Для реализации сайта-портфолио был выбран фреймворк React, который обеспечивает компонентный подход к разработке интерфейсов и гибкую архитектуру проекта. В качестве инструмента сборки и локального сервера использован Vite — быстрый и современный инструмент, обеспечивающий мгновенную перезагрузку при разработке и минимальный размер финального бандла. Все стили написаны вручную с использованием обычного CSS, без сторонних UI-фреймворков.

Общую структуру проекта и весь исходный код можно посмотреть на GitHab перейдя по ссылке в Приложение В. Исходный код размещён в папке src/, которая содержит точку входа, главный компонент, глобальные стили и два основных каталога: components/ и pages/:

* main.jsx — точка входа в приложение, где подключается App.jsx;
* App.jsx — главный компонент, содержащий логику роутинга;
* styles.css — глобальный CSS-файл;
* index.html — HTML-шаблон для сборки через Vite.

В приложении используется библиотека React Router версии 7. Маршруты настроены в App.jsx с использованием компонентов на рисунке 8:

* <HashRouter> — обёртка для всей маршрутизации;
* <Routes> и <Route> — определяют соответствие URL-адресов компонентам страниц.

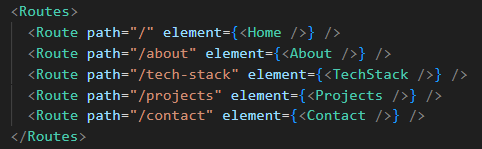


Рисунок 8 – Пример настроенного роутинга в App.jsx

Это позволяет организовать навигацию без перезагрузки страницы, улучшая производительность и пользовательский опыт.

Структура компонентов.

Каталог components/ содержит переиспользуемые UI-элементы:

* Header.jsx — навигационная панель;
* Footer.jsx — нижний колонтитул;
* ProjectCard.jsx — карточка проекта, реализованная как динамический компонент с передачей данных через props. Компонент обёрнут в React.memo для предотвращения лишних перерисовок;
* ContactForm.jsx — форма обратной связи с полями и кнопкой отправки;
* LoadingSpinner.jsx — заготовка компонента-индикатора загрузки, предназначенная для будущей динамической подгрузки данных.

Каталог pages/ хранит страницы сайта, каждая из которых представляет собой отдельный компонент:

* Home.jsx — блок приветствия и описание;
* About.jsx — сведения об авторе;
* TechStack.jsx — используемые технологии;
* Projects.jsx — список проектов с карточками;
* Contact.jsx — форма и контакты.

Такая модульная архитектура обеспечивает масштабируемость, простоту поддержки и возможность повторного использования компонентов в других проектах.

# Реализация компонентов и взаимодействие с API

В процессе разработки портфолио были созданы функциональные компоненты с использованием современных возможностей React. Все компоненты построены на хуках (hooks), что обеспечивает чистый и понятный код без использования классовых компонентов.

Компонент Header является одним из ключевых элементов интерфейса. Он содержит логотип, основную навигацию и адаптивное мобильное меню. Для управления состоянием открытия/закрытия мобильного меню используется хук useState. На рисунке 9 показана структура компонента Header с основными элементами навигации.



Рисунок 9 - Структура навигационного компонента Header

Особенность реализации заключается в том, что навигация автоматически адаптируется под размер экрана. На больших экранах отображается горизонтальное меню, а на мобильных устройствах — кнопка-гамбургер, которая открывает полноэкранное меню. Переключение между режимами реализовано через CSS-медиазапросы и управляемые классы.

Компонент ProjectCard представляет собой переиспользуемую карточку для отображения информации о проектах. Он принимает данные через props и отображает название, описание, изображение и теги проекта. Для оптимизации производительности компонент обёрнут в React.memo, что предотвращает ненужные перерисовки при неизменных данных.

Для управления локальным состоянием компонентов используются хуки useState и useCallback. В компоненте Projects данные о проектах хранятся в массиве и обрабатываются с помощью хука useMemo для оптимизации производительности. Это предотвращает пересоздание массива при каждом рендере компонента.

Компонент ContactForm демонстрирует работу с формами в React. Для каждого поля формы создано отдельное состояние, которое обновляется при вводе пользователя. На рисунке 10 показана схема работы формы обратной связи с управлением состоянием.

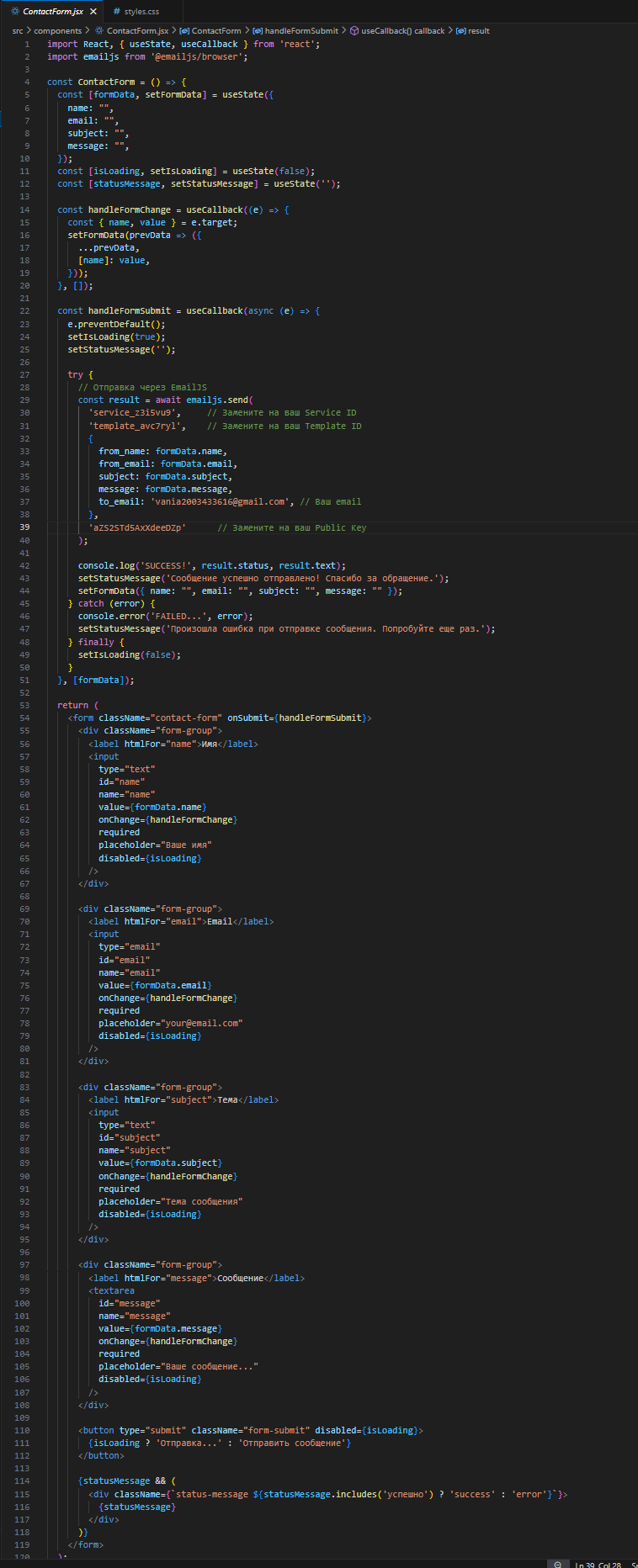


Рисунок 10 - Схема управления состоянием в форме обратной связи

Обработка изменений в форме выполняется через единый обработчик handleFormChange, который определяет, какое поле изменилось, и обновляет соответствующее значение в состоянии. Такой подход упрощает код и делает его более читаемым.

Для реализации функции отправки сообщений через форму обратной связи используется сервис EmailJS. Этот сервис позволяет отправлять электронные письма напрямую из браузера без создания собственного серверного кода.

Интеграция выполнена через официальную библиотеку @emailjs/browser. В компоненте ContactForm создана асинхронная функция handleFormSubmit, которая обрабатывает отправку формы. При отправке формы данные передаются в EmailJS API, который формирует и отправляет письмо на указанный адрес.

Процесс отправки включает несколько этапов: валидацию данных на стороне браузера, отправку запроса к API EmailJS, обработку ответа и отображение результата пользователю. Во время отправки интерфейс блокируется, чтобы предотвратить повторную отправку, а пользователю показывается индикатор загрузки.

В форме обратной связи реализована система обработки ошибок и отображения статусных сообщений. Если отправка сообщения прошла успешно, пользователь видит сообщение об успехе, а поля формы очищаются. В случае ошибки отображается соответствующее предупреждение.

Для улучшения пользовательского опыта добавлены состояния загрузки. Во время отправки формы кнопка отправки становится неактивной и меняет свой текст на "Отправка...". Поля формы также блокируются для ввода, что предотвращает случайные изменения данных во время обработки.

Валидация данных выполняется на двух уровнях: встроенная HTML5-валидация через атрибуты required и type, а также дополнительная проверка в JavaScript-коде. Это обеспечивает надёжную проверку данных перед отправкой.

Для оптимизации производительности приложения применено несколько техник. Использование React.memo для компонента ProjectCard предотвращает ненужные перерисовки карточек проектов. Хук useMemo в компоненте Projects кэширует массив проектов, что исключает его пересоздание при каждом рендере.

Обработчики событий в компонентах обёрнуты в useCallback, что предотвращает создание новых функций при каждом рендере. На рисунке 11 показана схема применённых оптимизаций в различных компонентах.



Рисунок 11 - Схема оптимизаций производительности в компонентах

Также используется ленивая загрузка (lazy loading) для компонентов страниц через React.lazy. Это позволяет загружать код страниц только при переходе на них, что уменьшает размер первоначально загружаемого пакета.

Все компоненты адаптированы для работы на различных устройствах. Стили написаны с использованием CSS-медиазапросов, которые изменяют внешний вид компонентов в зависимости от размера экрана. Основные точки останова установлены для мобильных устройств (до 768px) и планшетов (до 1024px).

Компонент Header демонстрирует адаптивный дизайн: на больших экранах показывается полная навигация, а на мобильных — компактное меню-гамбургер. Сетка проектов в компоненте Projects автоматически адаптируется под размер экрана, изменяя количество колонок.

Все интерактивные элементы имеют hover-эффекты и плавные переходы, что улучшает визуальную обратную связь с пользователем. Цветовая схема и типографика поддерживают единый стиль throughout всего приложения.

# Тестирование и демонстрация функциональности

После завершения разработки портфолио было проведено всестороннее тестирование функциональности и визуального представления сайта на различных устройствах и в разных браузерах. Главная страница портфолио представляет собой лаконичный интерфейс с центрированным контентом, где пользователь сразу видит основную информацию о разработчике. На рисунке 12 представлен внешний вид главной страницы в десктопной версии, демонстрирующий современный дизайн с градиентным фоном и четкой типографикой.

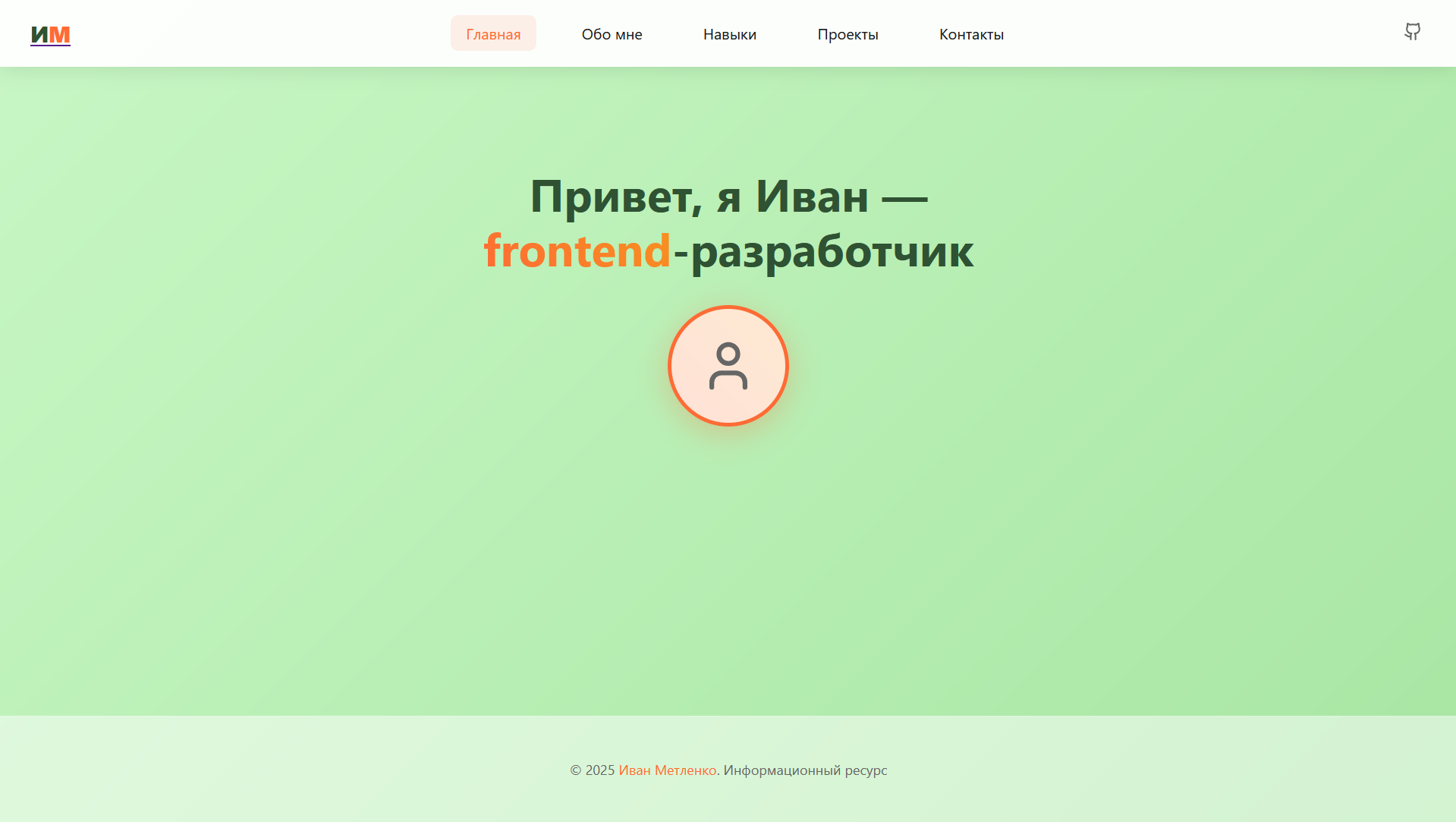


Рисунок 12 - Главная страница портфолио в десктопной версии

Навигационная система работает корректно во всех разделах сайта, обеспечивая плавные переходы между страницами благодаря React Router. Активный пункт меню визуально выделяется оранжевым цветом, что соответствует общей цветовой схеме проекта. Логотип в левом верхнем углу представляет собой стилизованные инициалы разработчика и служит ссылкой на главную страницу.

Страница "Обо мне" содержит структурированную информацию о разработчике, разделенную на логические блоки с использованием карточек с эффектом размытия фона. На рисунке 13 показано оформление информационных блоков на странице "Обо мне", где каждый раздел имеет собственную область с полупрозрачным фоном и четкими границами.

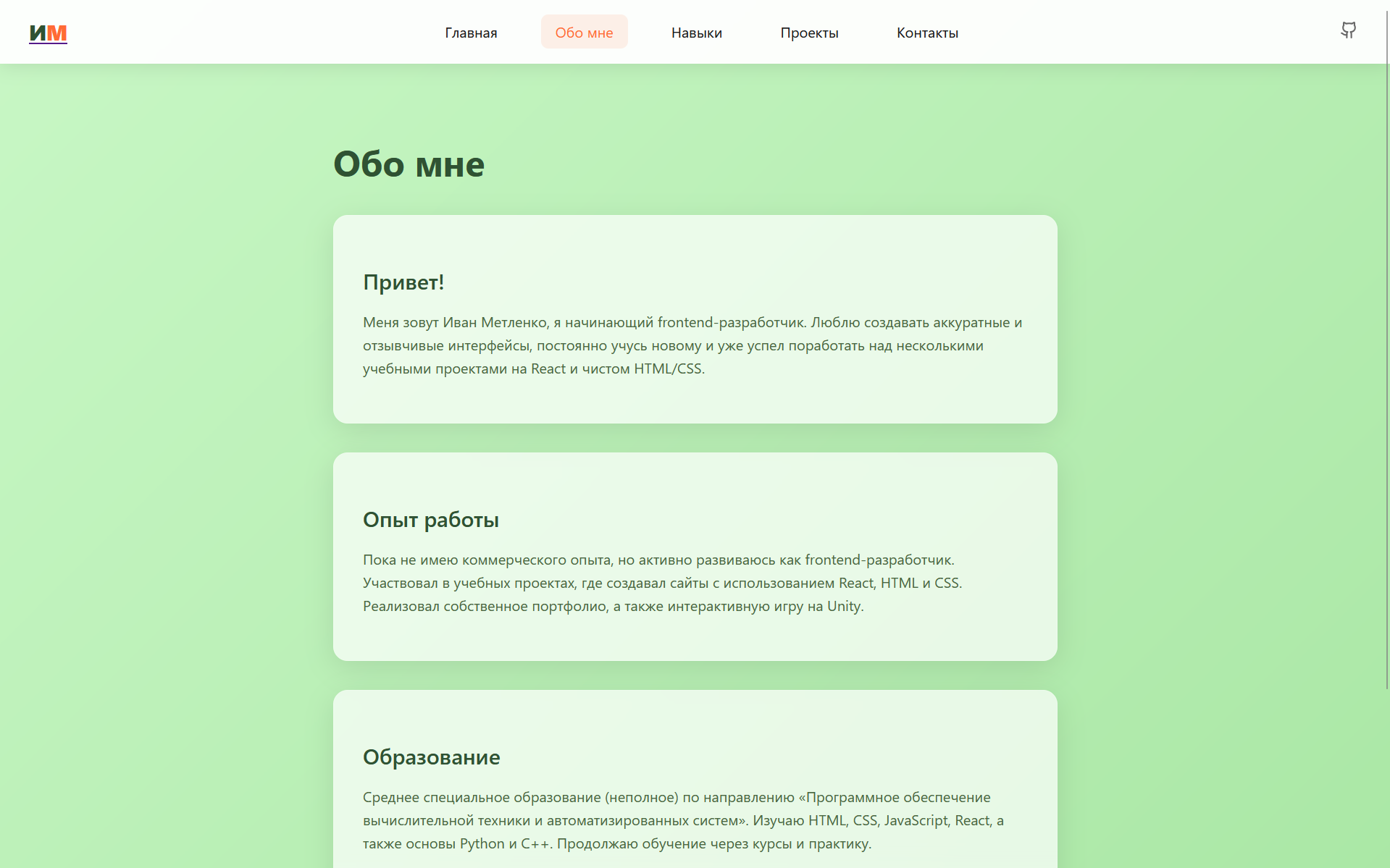


Рисунок 13 - Страница "Обо мне" с информационными блоками

Особое внимание при тестировании было уделено проверке адаптивного поведения сайта на мобильных устройствах. Мобильная версия главной страницы сохраняет все ключевые элементы десктопной версии, но с оптимизированными размерами и расположением элементов. На рисунке 14 представлена мобильная версия главной страницы, демонстрирующая корректное масштабирование контента и читаемость текста.

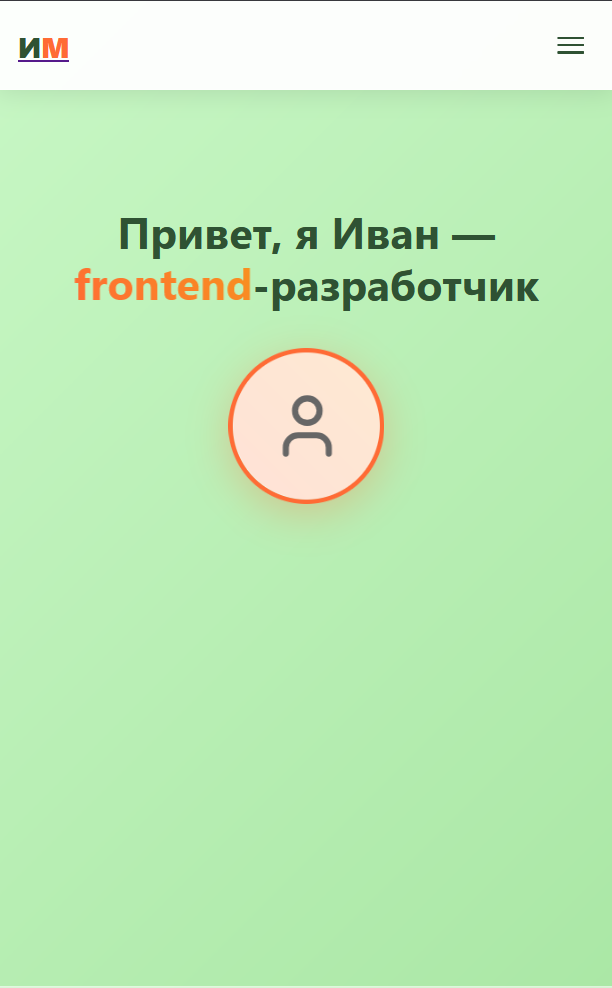


Рисунок 14 - Мобильная версия главной страницы

Навигационное меню на мобильных устройствах автоматически трансформируется в бургер-меню, которое при активации разворачивается в полноэкранное overlay-меню. Анимация открытия и закрытия меню выполнена плавно с использованием CSS-переходов, что создает современный пользовательский опыт. На рисунке 15 показано активное состояние мобильного меню с полным списком навигационных элементов.

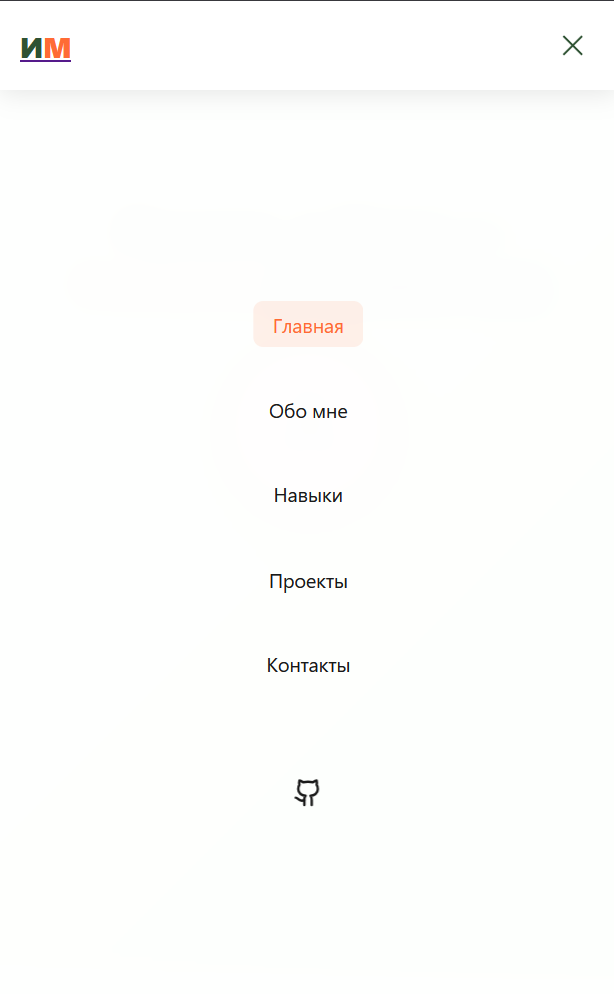


Рисунок 15 - Активное мобильное меню в полноэкранном режиме

Страница "Навыки" представляет технический стек навыков в виде аккуратно оформленных карточек, сгруппированных по сетке. Каждый элемент — отдельная технология, такая как HTML5, CSS3, JavaScript, React, Git, VS Code, GitHub и Figma. Карточки выполнены с закруглёнными краями и тенями, а при наведении плавно смещаются вверх, создавая эффект движения и подчёркивая интерактивность интерфейса. На рисунке 16 показан внешний вид страницы в десктопной версии.

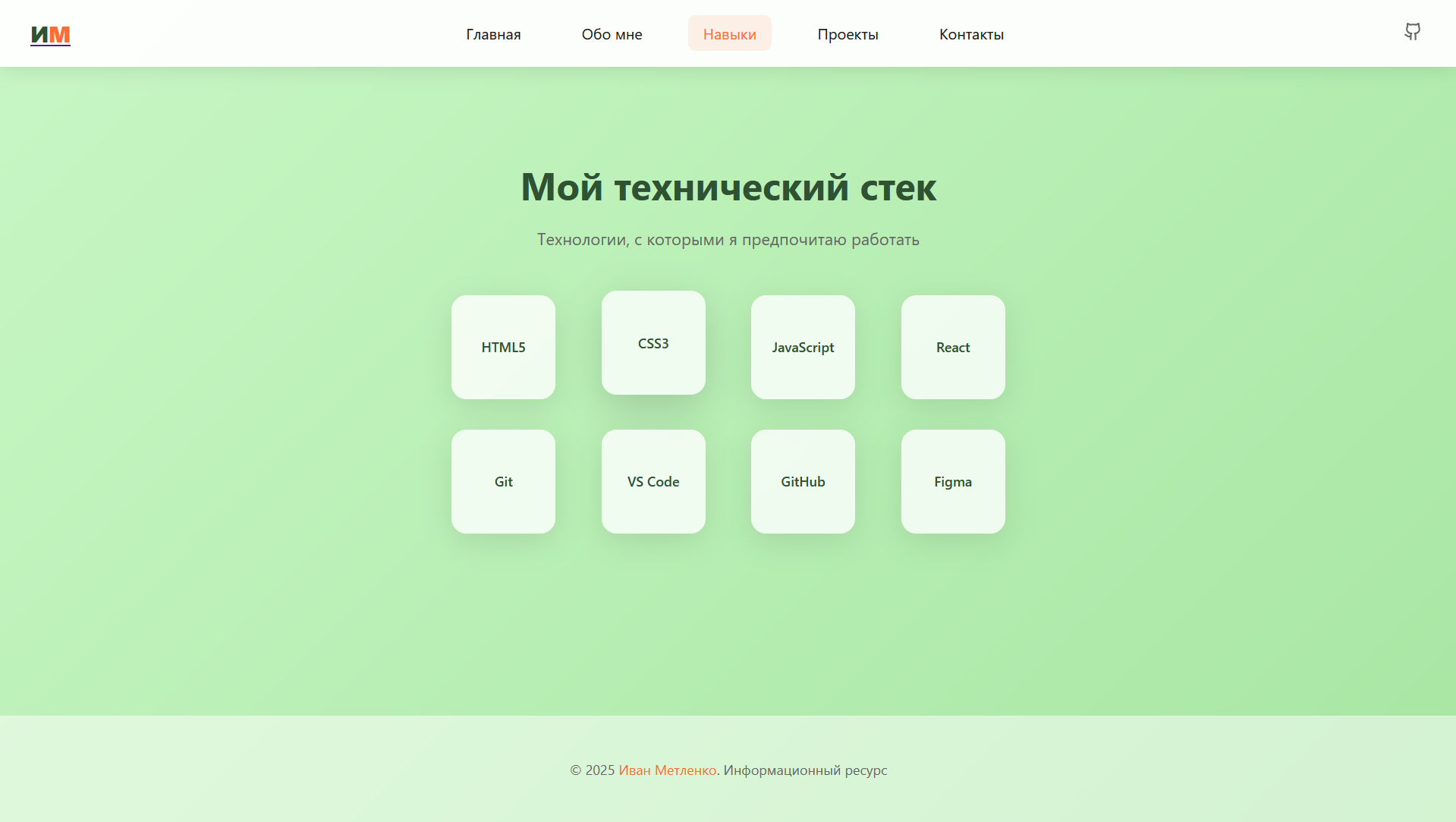


Рисунок 16 — Страница "Навыки" с интерактивными элементами стека

Страница проектов демонстрирует сетку карточек с информацией о выполненных работах. Каждая карточка содержит изображение проекта или градиентную заглушку, название, описание и список используемых технологий. На рисунке 17 представлен общий вид страницы проектов с адаптивной сеткой карточек, которая автоматически подстраивается под ширину экрана.

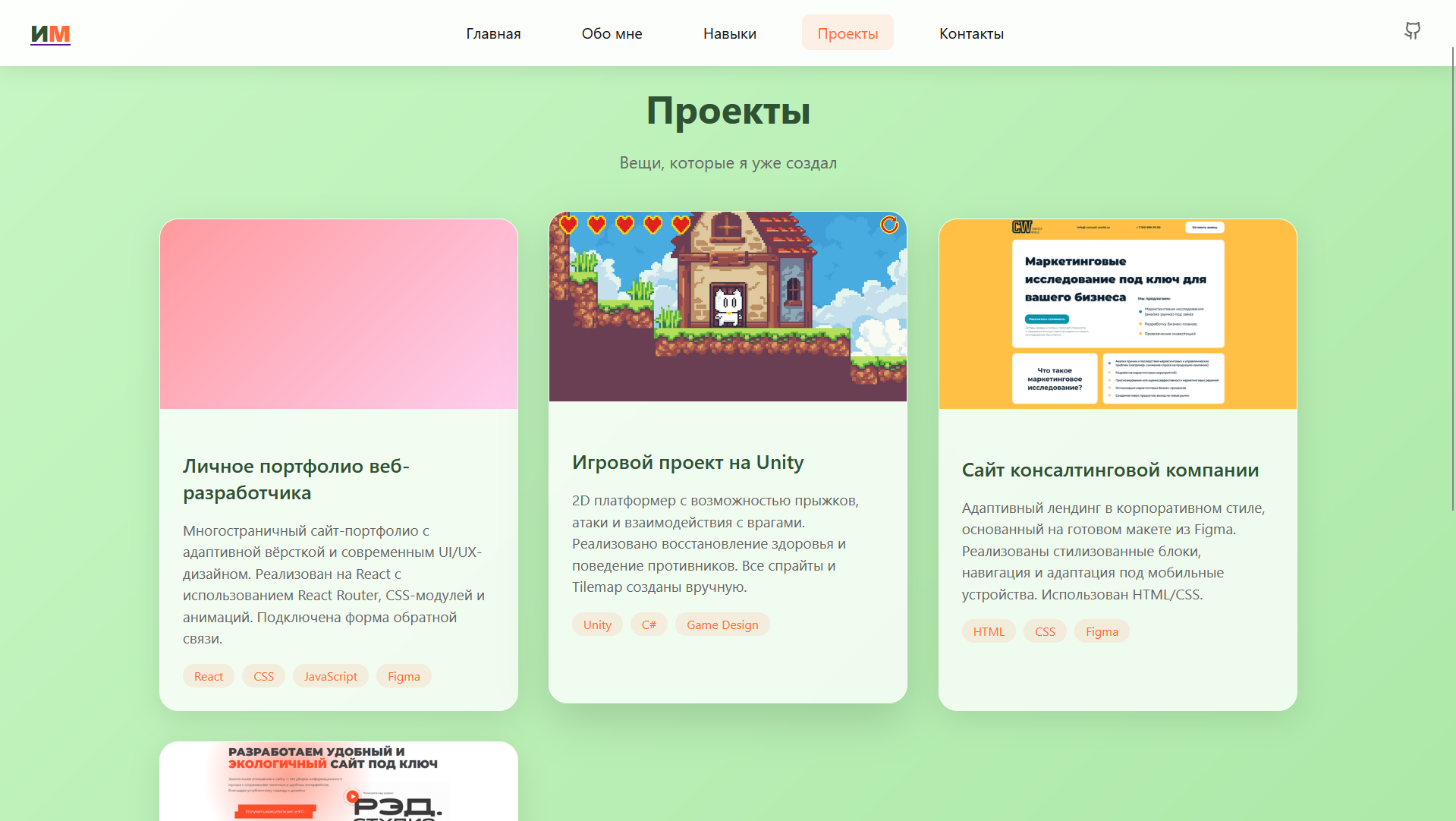


Рисунок 17 - Страница проектов с сеткой карточек

Карточки проектов имеют интерактивные эффекты при наведении курсора, включая плавное поднятие элемента и усиление тени, что создает ощущение глубины интерфейса. Теги технологий внутри каждой карточки стилизованы в соответствии с общей дизайн-системой и помогают быстро идентифицировать используемый стек технологий.

В мобильной версии сетка проектов трансформируется в одноколоночный layout, где карточки располагаются вертикально друг под другом. Это обеспечивает оптимальную читаемость контента и удобство навигации на устройствах с небольшим экраном.

Страница контактов включает в себя как прямую ссылку на электронную почту, так и интерактивную форму обратной связи. На рисунке 18 показан внешний вид контактной формы с полями для ввода имени, email, темы сообщения и текста обращения.

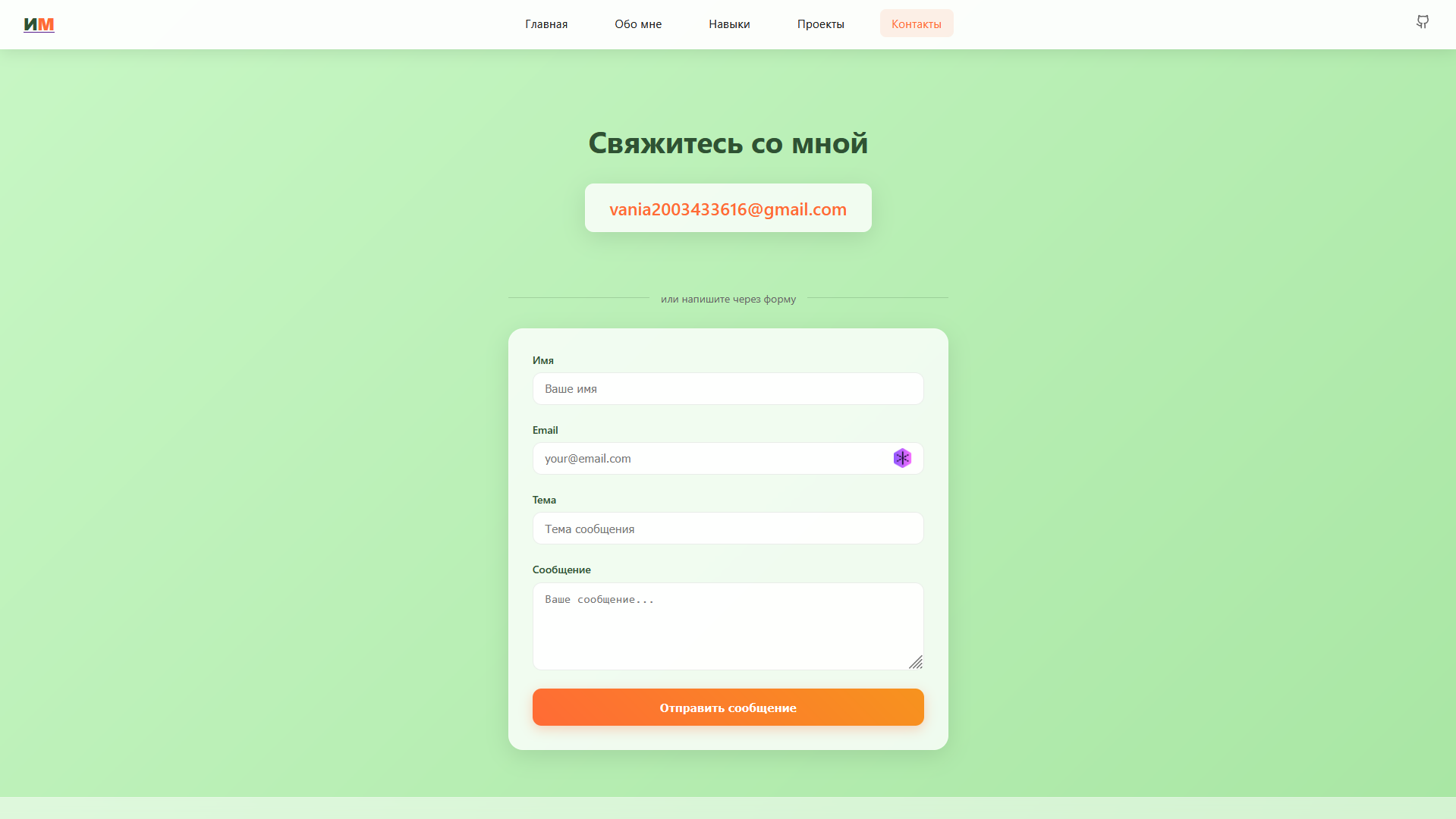


Рисунок 18 - Форма обратной связи на странице контактов

Тестирование функциональности формы проводилось в несколько этапов. Первоначально была проверена HTML5 валидация полей, которая корректно срабатывает при попытке отправки формы с незаполненными обязательными полями или некорректным форматом email-адреса. Визуальная индикация ошибок валидации соответствует общему стилю интерфейса.

Процесс отправки сообщения сопровождается изменением текста кнопки на "Отправка..." и блокировкой всех полей формы для предотвращения повторной отправки. На рисунке 19 демонстрируется состояние формы во время отправки сообщения с заблокированными элементами управления.

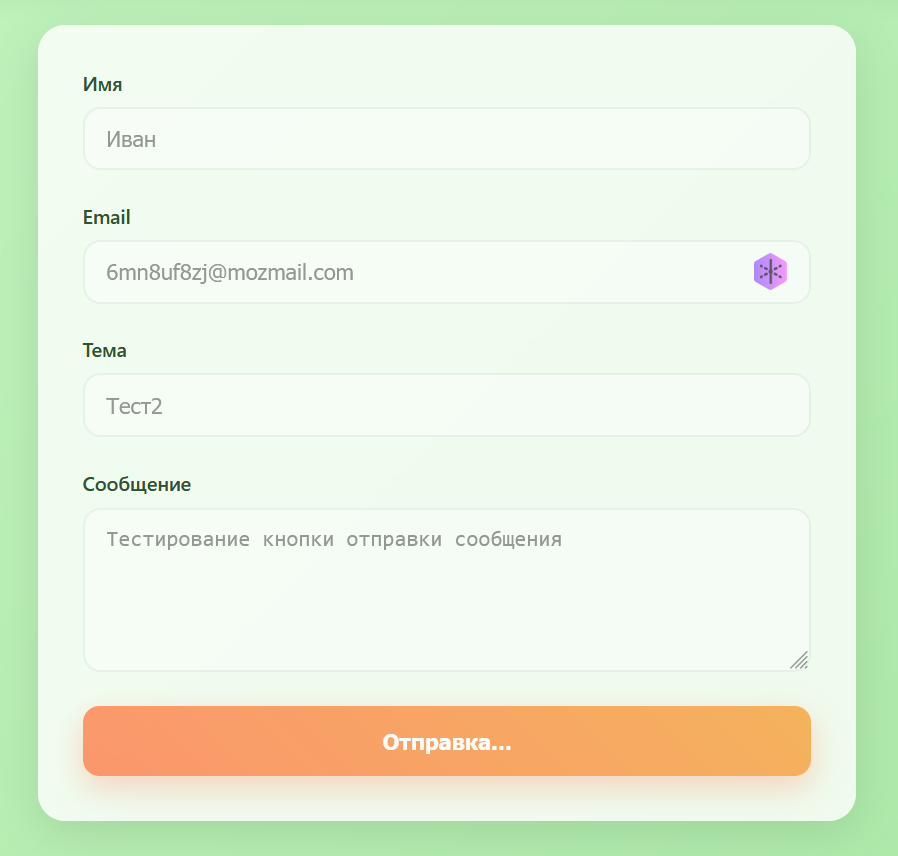


Рисунок 19 - Форма в состоянии отправки сообщения

Интеграция с сервисом EmailJS была протестирована путем отправки реальных сообщений через форму обратной связи. Конфигурация сервиса включает настройку шаблона письма, который автоматически формируется на основе данных, введенных пользователем в форму. Успешная отправка сообщения сопровождается появлением зеленого уведомления с текстом подтверждения.

В случае возникновения ошибки при отправке, например, при отсутствии интернет-соединения или проблемах с API сервиса, пользователю отображается красное уведомление с просьбой повторить попытку. На рисунке 20 и 21 показаны примеры успешного и неуспешного статусных сообщений формы.

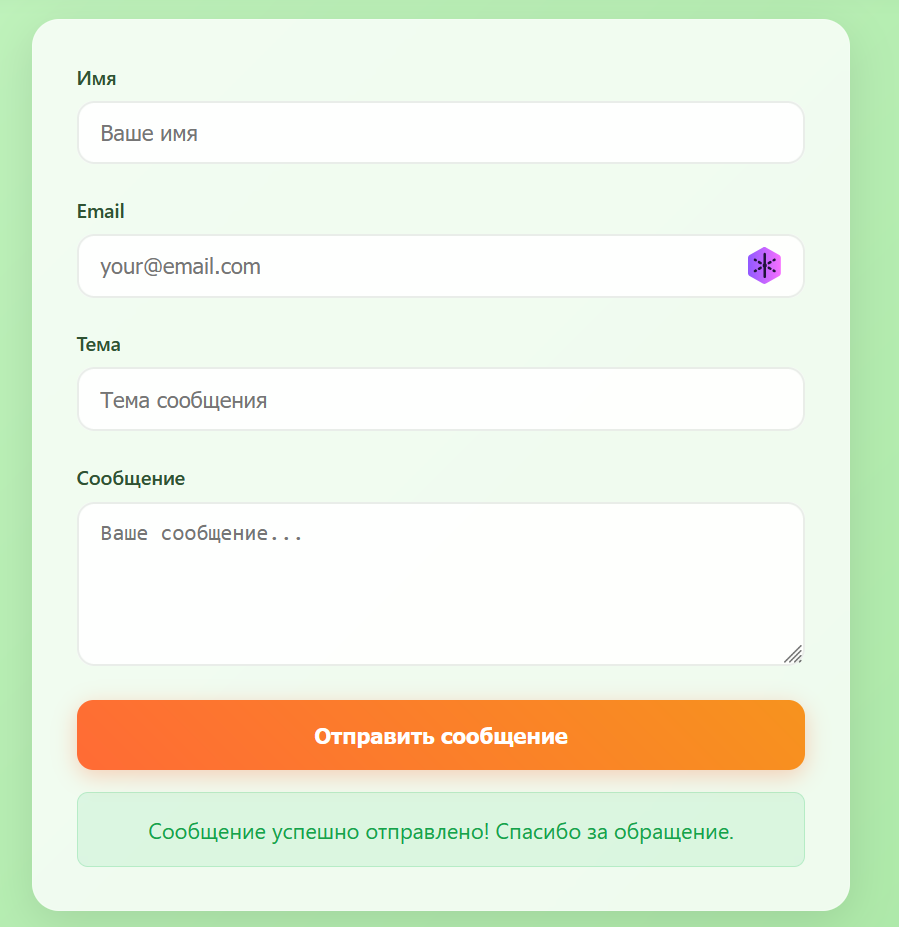


Рисунок 20 - Статусные сообщения формы успешной обратной связи

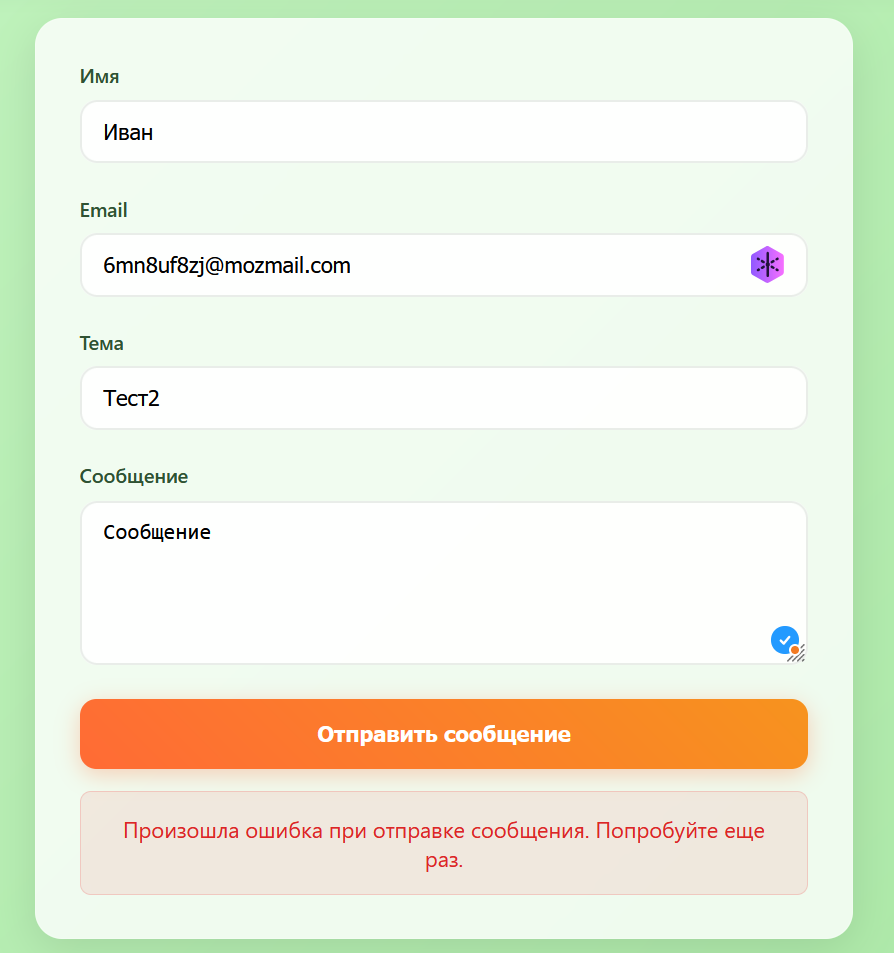


Рисунок 21 - Статусные сообщения формы неуспешной обратной связи

Тестирование показало, что время отправки сообщения составляет в среднем 2-3 секунды, что является приемлемым для пользовательского опыта. После успешной отправки все поля формы автоматически очищаются, что позволяет пользователю отправить новое сообщение без необходимости ручной очистки полей.

Портфолио было протестировано в основных современных браузерах, включая Chrome, Firefox, Safari и Edge. Все функциональные элементы работают корректно во всех тестируемых браузерах, включая CSS-анимации, переходы и JavaScript-функциональность. Особое внимание было уделено проверке поддержки CSS Grid и Flexbox, которые широко используются в layout системе проекта.

Тестирование на различных разрешениях экрана показало корректную работу адаптивного дизайна от 320px (мобильные устройства) до 1920px и выше (широкоформатные мониторы). Контрольные точки (breakpoints) в медиа-запросах обеспечивают плавное масштабирование интерфейса без нарушения визуальной целостности.

Анализ производительности сайта был проведен с использованием инструментов разработчика браузера и онлайн-сервисов для аудита веб-страниц. Ленивая загрузка компонентов через React.lazy показала значительное улучшение времени первоначальной загрузки страницы. Размер JavaScript-бандла для первоначальной загрузки составил менее 100KB в сжатом виде.

Использование оптимизированных изображений и CSS-градиентов вместо тяжелых графических файлов способствовало быстрой загрузке страниц даже на медленных соединениях. Все интерактивные элементы интерфейса отвечают на действия пользователя без заметных задержек, что обеспечивает плавный пользовательский опыт.

В рамках тестирования была проверена базовая доступность (accessibility) сайта. Все интерактивные элементы доступны для навигации с клавиатуры, что важно для пользователей с ограниченными возможностями. Контрастность текста соответствует требованиям WCAG, обеспечивая читаемость контента для пользователей с нарушениями зрения.

Семантическая HTML-разметка с использованием соответствующих тегов (header, nav, main, section) обеспечивает правильную интерпретацию контента экранными читалками. Все изображения содержат альтернативный текст, а интерактивные элементы имеют понятные aria-label атрибуты.

Результаты проведенного тестирования подтвердили готовность портфолио к публикации и использованию в качестве профессионального представительства веб-разработчика в интернете. Все основные функции работают стабильно, интерфейс адаптируется под различные устройства, а пользовательский опыт остается комфортным во всех протестированных сценариях использования.

**ЗАКЛЮЧЕНИЕ**

В ходе прохождения производственной практики была достигнута основная цель — формирование и развитие профессиональных компетенций в области проектирования и разработки веб-проектов. Практика включала полный цикл разработки персонального сайта-портфолио: от выбора технологий до реализации и тестирования готового веб-приложения.

В качестве основных технологий были выбраны библиотека React и CSS, что обусловлено их гибкостью, популярностью и широкой поддержкой со стороны профессионального сообщества. Использование компонентного подхода React, декларативной логики отображения интерфейса, а также инструментов оптимизации (React.memo, useMemo, lazy loading) позволило создать производительное, масштабируемое и адаптивное веб-приложение.

Особое внимание было уделено пользовательскому интерфейсу: спроектирован прототип в Figma с учетом принципов UI/UX-дизайна, реализованы адаптивные страницы с современным визуальным стилем, обеспечена интерактивность всех элементов. Разработка охватывала как технические аспекты (настройка роутинга, оптимизация рендеринга, работа с EmailJS API), так и обеспечение высокого пользовательского опыта (адаптивность, отзывчивость, доступность интерфейса).

В результате практики были закреплены и углублены знания в области фронтенд-разработки, освоены современные технологии и инструменты, а также выработаны навыки проектной работы — от идеи до финального тестирования. Полученные результаты демонстрируют готовность к выполнению профессиональных задач в области веб-разработки и формируют базу для дальнейшего профессионального роста.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. React. Начало работы // Официальная документация React [Электронный ресурс]. – Режим доступа: <https://ru.legacy.reactjs.org/docs/getting-started.html> (дата обращения: 02.06.2025).
2. EmailJS. Официальная документация [Электронный ресурс]. – Режим доступа: <https://www.emailjs.com/docs/> (дата обращения: 03.06.2025).
3. Vite. Начало работы // Официальная документация Vite [Электронный ресурс]. – Режим доступа: <https://vitejs.ru/guide/> (дата обращения: 04.06.2025).
4. React Router. Руководство по React Router [Электронный ресурс]. – Режим доступа: <https://my-js.org/docs/guide/react-router> (дата обращения: 05.06.2025).
5. MDN Web Docs. Официальная документация по HTML, CSS, JavaScript [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/> (дата обращения: 06.06.2025).
6. React. Быстрый старт, 2-е изд. / Издательский дом «Питер». – СПб.: Питер, 2023. – 320 с. (дата обращения: 07.06.2025).
7. МакГрат М. JavaScript для начинающих. 6-е издание. – М.: Эксмо, 2023. – 384 с. (дата обращения: 08.06.2025).
8. Скотт А.Д., Пауэрс Ш. JavaScript. Рецепты для разработчиков. 3-е издание. – М.: Вильямс, 2022. – 512 с. (дата обращения: 09.06.2025).
9. Дронов В.А. JavaScript. 32 урока для начинающих. – М.: БХВ-Петербург, 2021. – 256 с. (дата обращения: 10.06.2025).
10. EmailJS: Прямая отправка email с сайта через JavaScript: методы и код [Электронный ресурс]. – Режим доступа: <https://sky.pro/wiki/javascript/pryamaya-otpravka-email-s-sayta-cherez-java-script-metody-i-kod/> (дата обращения: 11.06.2025).
11. React. Официальная документация (англ.) [Электронный ресурс]. – Режим доступа: <https://react.dev/> (дата обращения: 12.06.2025).
12. Vite. Официальная документация (англ.) [Электронный ресурс]. – Режим доступа: <https://vitejs.dev/> (дата обращения: 13.06.2025).
13. React Router. Официальная документация (англ.) [Электронный ресурс]. – Режим доступа: <https://reactrouter.com/> (дата обращения: 14.06.2025).
14. Mozilla Developer Network. MDN Web Docs (англ.) [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/> (дата обращения: 15.06.2025).
15. Флэнаган Д. JavaScript. Подробное руководство. 7-е издание. – СПб.: Питер, 2021. – 1088 с. (дата обращения: 16.06.2025).
16. Бэнкс А. React и Redux. Разработка сложных интерфейсов. – М.: ДМК Пресс, 2022. – 400 с. (дата обращения: 17.06.2025).
17. Стаффорд В. CSS. Карманный справочник. – М.: О’Рейли, 2021. – 256 с. (дата обращения: 18.06.2025).
18. Фримен Э., Робсон Э. HTML и CSS. Разработка и дизайн веб-сайтов. – М.: Эксмо, 2023. – 512 с. (дата обращения: 19.06.2025).
19. Крокфорд Д. JavaScript: сильные стороны. – М.: Символ-Плюс, 2020. – 176 с. (дата обращения: 20.06.2025).
20. Стефанов С. JavaScript. Шаблоны. – М.: Вильямс, 2021. – 320 с. (дата обращения: 21.06.2025).

ПРИЛОЖЕНИЕ A

**Техническое задание**

Проект: Разработка сайта-портфолио frontend-разработчика на React.

Цель: создать современный адаптивный сайт, демонстрирующий профессиональные навыки, реализованные проекты и обеспечивающий удобную коммуникацию с потенциальными работодателями и заказчиками.

**Описание целевой аудитории**

Сайт должен быть ориентирован на пользователей, интересующихся профессиональной деятельностью начинающего веб-разработчика. В первую очередь это HR-специалисты и рекрутеры, которые ищут кандидатов на должности начального уровня. Также сайт будет полезен заказчикам, рассматривающим возможность сотрудничества в области создания пользовательских интерфейсов, веб-сайтов и одностраничных приложений. Кроме того, ресурс может быть интересен преподавателям, оценивающим практическую подготовку студента, и другим разработчикам, стремящимся к обмену опытом и взаимодействию внутри профессионального сообщества.

**Функциональные требования**

Структура сайта должна включать в себя несколько разделов: главную страницу с приветствием и кратким описанием, страницу с информацией о разработчике, блок с техническим стеком, галерею проектов, а также раздел с контактами и формой обратной связи. Навигация между этими разделами осуществляется без перезагрузки страницы, с использованием возможностей React Router.

Сайт должен иметь адаптивный дизайн, корректно отображаться на различных устройствах — от мобильных телефонов до широкоформатных мониторов — и обеспечивать интуитивную навигацию как через стандартное верхнее меню, так и через выпадающее бургер-меню в мобильной версии. Визуальное оформление должно быть выдержано в едином стиле, а взаимодействие с элементами сопровождается анимациями и эффектами наведения, что делает пользовательский опыт более приятным и современным.

Форма обратной связи должно предоставлять пользователю возможность отправить сообщение, заполнив стандартные поля (имя, e-mail, тема, текст сообщения). Отправка данных осуществляется через сервис EmailJS. Должна быть предусмотрена валидация данных как на уровне HTML, так и с помощью JavaScript. После отправки пользователю должно отображатся сообщение о статусе: успешная отправка или возникшая ошибка, при этом форма временно блокируется во избежание повторных запросов.

**Нефункциональные требования**

Сайт должен обладать высокой производительностью, что обеспечивается за счет применения техники ленивой загрузки страниц, использования оптимизированных изображений и минимизации размера начального JavaScript-бандла. Компоненты приложения должны быть построены с использованием хуков React и оптимизированы через React.memo, useMemo, useCallback, что позволит избежать лишних перерисовок и ускорить отклик интерфейса.

Необходимо обеспечить кроссбраузерную совместимость. Сайт должен стабильно работать во всех современных браузерах, включая Chrome, Firefox, Safari и Microsoft Edge. Используемые технологии верстки — CSS Grid и Flexbox — должны быть реализованы с учетом различий в поддержке между браузерами.

Особое внимание должно уделятся доступности. Пользовательский интерфейс должен соответствовать основным принципам WCAG: быть доступным для навигации с клавиатуры, иметь достаточный контраст текста и включать альтернативные подписи для изображений и aria-атрибуты для интерактивных элементов. Это обеспечит использование сайта людьми с ограниченными возможностями.

Безопасность взаимодействия также важна. Все данные, вводимые пользователем, проходят валидацию, что предотвращает возможные XSS-атаки. Отправка сообщений через форму защищена от повторных запросов, а в случае сбоев предоставляется соответствующая визуальная обратная связь.

Сайт должен разрабатываться с использованием библиотеки React, инструментальной среды Vite, и оформляется вручную с помощью CSS. Прототип пользовательского интерфейса должен разрабатыватся в Figma и служит основой для финальной вёрстки. Хранение и контроль версий проекта должен осуществлятся с использованием Git.

ПРИЛОЖЕНИЕ Б

Ссылка на макет в Figma:

<https://www.figma.com/design/O7IqX1Z6mDgbMB1bLVQLMf/Portfolio?node-id=0-1&p=f&t=kvjIrOK0IzsRhADj-0>

ПРИЛОЖЕНИЕ В

Ссылка на GitHub проекта:

<https://github.com/Aidem101/Portfolio>