

## РАЗРАБОТКА ВЕБ ПРИЛОЖЕНИЯ "MASTERS.KZ" С ИСПОЛЬЗОВАНИЕМ АКТУАЛЬНЫХ ВЕБ ТЕХНОЛОГИЙ

**Барабанов Олег Сергеевич**

[Boleg748@gmail.com](mailto:Boleg748@gmail.com)

Студент 3 курса образовательной программы «Информационные системы»  
Алматинский Технологический Университет, г.Алматы, Республика Казахстан  
Научный руководитель, м.т.н., лектор – **Букенова И. Н.**

В современном мире веб-технологии играют решающую роль в развитии информационных систем и сервисов. Проект "Masters.kz" направлен на создание уникальной веб-приложения, который соединяет владельцев бизнеса из разных отраслей с их клиентами, предлагая широкий спектр услуг от деловых до личных. Анализируя текущее состояние и волатильность рынка, данное исследование направлено на количественную оценку волатильности и эффективности веб-приложения "Masters.kz", а также его вспомогательных функций. Используя передовые технологические достижения, проект направлен на достижение беспрецедентного уровня комфорта, надежности и эффективности для пользователей.

Современный рынок характеризуется увеличением спроса на онлайн-сервисы, предоставляющие услуги в сфере образования, ремонта, красоты и многих других. Веб-приложение может удовлетворить этот спрос, предлагая удобный и эффективный способ подключения всех.

Прогресс в области веб-технологий, таких как Progressive Web Apps (PWA), React, Vue.js, Angular, Node.js и других, позволяет создавать более быстрые, надежные и интерактивные веб-приложения. Исследование и применение этих технологий даст возможность реализовать передовые функции и улучшить пользовательский опыт.

В условиях цифровизации экономики бизнесам необходимо адаптироваться к изменяющимся условиям рынка, предлагая свои услуги в онлайн-формате. Разработка веб-приложения позволит компаниям и частным предпринимателям расширить свою клиентскую базу и увеличить доходы через цифровую платформу [1].

В "Masters.kz" техническая составляющая проекта заключается в том, что он играет ключевую роль, обособляя возможности, гибкость и возможность масштабирования. Необходимо обратить внимание на такие технологические достижения, как администрирование на основе Docker, администрирование на основе Kubernetes, а также

применение микросервисной архитектуры для отслеживания гибкости и невидимости компонентов системы.

Безопасность данных и транзакций занимает центральное место в разработке веб-приложения. В этом аспекте будут использоваться передовые методы шифрования, аутентификации и авторизации, такие как OAuth 2.0 и JSON Web Tokens (JWT), для защиты пользовательских данных и предотвращения неавторизованного доступа к системе. Кроме того, будет проведена регулярная оценка уязвимостей и аудит безопасности для выявления и устранения потенциальных рисков.

Интеграция и взаимодействие различных сервисов и компонентов приложения требует применения надежных и эффективных решений для обмена данными и синхронизации состояний. В этом контексте будут использоваться такие технологии, как RESTful API и GraphQL, для обеспечения гибкости взаимодействия между фронтендом и бэкендом, а также между различными микросервисами. Для обработки реального времени и обеспечения мгновенного общения между пользователями будет использоваться WebSocket.

Кроме того, в проекте уделено внимание созданию интуитивно понятного и доступного пользовательского интерфейса, что потребует применения современных фреймворков и библиотек для фронтенда, таких как React.js или Vue.js, обеспечивающих высокую производительность и отзывчивость интерфейса. Адаптивный дизайн и доступность станут ключевыми факторами в разработке, чтобы приложение было удобным для использования на любых устройствах и для пользователей с различными потребностями.

Таким образом, успешное создание веб-приложения "Masters.kz" потребовало комплексного подхода к решению технических, операционных и этических проблем. Это было достигнуто за счёт использования инструмента планирования, выбора наиболее эффективных технологических решений и не инвазивных методов оптимизации бизнес-процессов.

Аналитические исследования включают в себя сбор и анализ отзывов пользователей, мониторинг показателей взаимодействия с платформой и оценку улучшений производительности в результате внедрения новых технологий.

Практическая реализация проекта предполагает разработку подробного плана внедрения выбранных технологий и методик, включая тестирование и оптимизацию приложения для различных устройств и браузеров. Важным аспектом этого процесса является защита пользовательских данных и транзакций. Это требует использования передовых методов шифрования и защиты данных.

В ходе реализации проекта особое внимание уделено отзывам пользователей, чтобы адаптировать функциональность и интерфейс к их предпочтениям и потребностям. Это не только помогло проверить первоначальные гипотезы, но и позволило определить дополнительные пути улучшения и развития платформы.

**Методология.** В проекте "Masters.kz" используется компонентный метод [2] для создания легко понятной структуры приложения, в которой каждый компонент имеет свое назначение. В результате упрощается процесс разработки и обеспечение поддержки продукта. Как видно из рисунка 1, компоненты предназначены для повторного использования, что значительно ускоряет создание новых функций и усовершенствований для платформы.

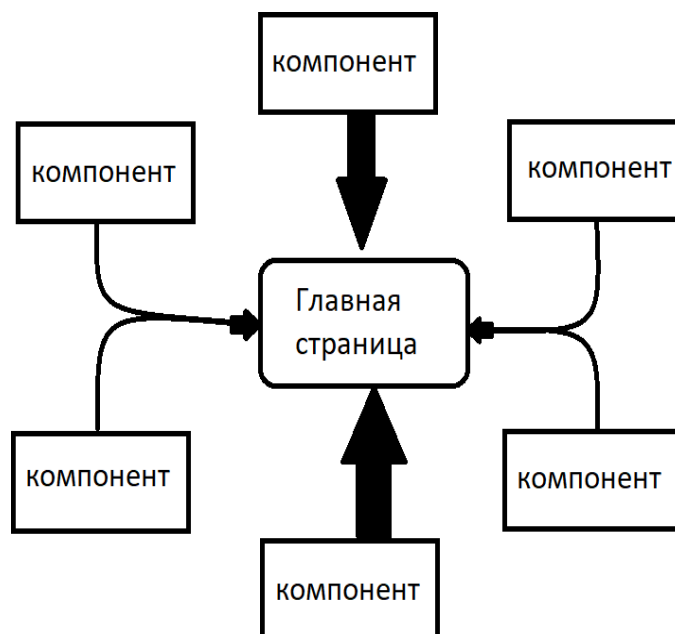


Рисунок 1. Компонентный подход.

Декларативное программирование упрощает разработку и отладку приложений, предоставляя прозрачный и интуитивно понятный пользовательский интерфейс. Этот метод позволяет задать пользовательский интерфейс для каждого уникального состояния приложения. Это повышает стабильность системы и упрощает управление переходами состояний.

Постоянное перемещение данных в приложении обеспечивается шаблоном однонаправленного потока данных [3], который облегчает управление состоянием и отслеживание изменений. Это, как видно на рисунке 2, приводит к повышению производительности и упрощению разработки, поскольку разработчики постоянно осведомлены об источнике и направлении данных.



Рисунок 2. Однонаправленный поток данных.

Крупные проекты, нуждаются в четкой и хорошо организованной структуре стилей для предотвращения конфликтов названий классов. Для этой цели была использована технология BEM (Block, Element, modifier) [4], которая также повышает предсказуемость стиля и упрощает его модификацию.

Использование Redux для управления состоянием приложения, создало централизованное хранилище состояний, что позволяет работать с ним прозрачно и предсказуемо. Это позволяет интерфейсу быть более отзывчивым и работать лучше, упрощая мониторинг изменений и синхронизацию между компонентами.

Таким образом, разработанный "Masters.kz" представляет собой эффективное, надежное и легко масштабируемое веб-приложение, которое отвечает потребностям пользователей и предоставляет высококачественные услуги благодаря использованию современных методов и технологий разработки.

**Результаты.** Первый шаг в разработке веб-приложения "Masters.kz" предполагает тщательный анализ существующего материала [5]. На этот проект влияют особенности и макет веб-сайта profi.ru. В результате исследований все важные элементы сайта были доработаны, а базовая функциональность протестирована и задокументированная. Это позволило получить четкое представление о дизайне и функциях аналогичного веб-приложения. Основываясь на собранных данных, принято решение создать веб-приложение со структурой и логикой, аналогичными показанным на рисунке 3.



Рисунок 3. Схема приложения.

Следующий этап работы – это проектирование дизайна будущего приложения. Определяются основные цветовые решения, и начинается создание эскизов, которые закладывают основу визуальной концепции проекта. После утверждения концептуальных решений начинается разработка веб-приложения. Разработка происходит поэтапно, где первым шагом является создание собственного окружения [6]. Процесс создания приложения начинается с команды «`npm init react-app my-app`» которую нужно ввести в `cmd`, после чего сайт начинает загружаться вместе со всеми зависимостями и самим приложением, что может занять от одной до трех минут.

Для доступа к проекту используется команда «`cd my-app`», а для запуска локального сервера и тестирования проекта применяется команда «`npm run start`». Начальная структура проекта представлена на рисунке 4.

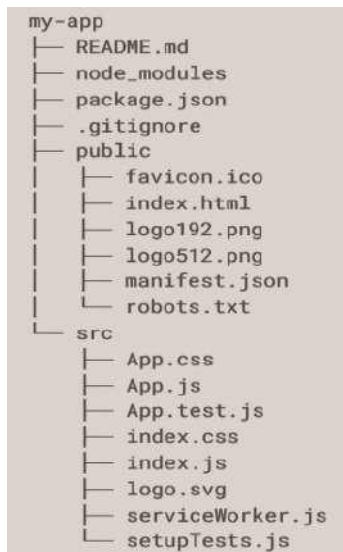


Рисунок 4. Структура проекта.

Важными элементами структуры проекта являются файл `public/index.html`, служащий шаблоном страницы приложения, и файл `src/index.js`, который является входной точкой для JavaScript-кода. Система сборки Webpack обрабатывает только файлы в директории `src`, что делает тщательную организацию файловой структуры проекта критически важной. После создания проекта следует внести изменения, начиная с начальной страницы, которая изначально выглядит как на рисунке 5.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

Рисунок 5 – Главный файл.

После внесения правок данный код стал выглядеть как на рисунке 6.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { Provider } from 'react-redux';

import store from './redux/store';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}>
    <App />
  </Provider>,
);
```

Рисунок 6. Главный файл после изменений.

В процессе дальнейшей разработки были внесены значительные изменения в исходный код, включая добавление провайдера для управления состоянием приложения с помощью Redux и настройку маршрутизации с использованием React-Router для организации навигации по разделам веб-приложения (рисунок 7).

```

function App() {
  const router = createBrowserRouter([
    {
      path: Path.register,
      element: <Register />,
    },
    {
      path: Path.home,
      element: <Main />,
    },
    {
      path: Path.activeOrders,
      element: <ActiveOrders />,
    }
  ]);
  return (
    <div className='App'>
      <RouterProvider router={router} />
    </div>
  );
}

```

Рисунок 7. App файл.

В самом роутере нужно добавить состояние, которое будет хранить путь на страницу, а также сам элемент (рисунок 8).

```

    {
      path: Path.home,
      element: <Main />,
    },

```

Рисунок 8. Router.

В приложении используется следующая логика: если есть определенный token (например, идентификатор пользователя), то должен отрисоваться компонент, который называется orderprofile, как показано на рисунке 9. Этот компонент имеет пропсы userName и userID. Если же данного token'a нет, то будет отображаться неклиентский компонент.

Кроме того, внедрена система отображения контента и контроля доступа, основанная на статусе пользователя. Предлагаемая система позволяет незарегистрированным пользователям просматривать общедоступный интерфейс, а зарегистрированным пользователям - получать доступ к индивидуальному контенту. Это достигается путем проверки статуса аутентификации пользователя и условного отображения компонентов приложения на его основе, что позволяет создавать более приватные и безопасные условия для пользователей (рис. 9).

```

<Layout>
  <div className='container_time'>
    <AsideNav />
    <section className='main_aside'>
      {token ? (
        <OrderProfile userName={res} userID={id} />
      ) : (
        <NotClient />
      )}
    </section>
  </div>
</Layout>

```

Рисунок 9. Страница проверки.

В проект также внедрена функция, которая может динамически изменять содержимое в зависимости от контекста пользователя как показано на рисунке 10. Данная функция отображает сообщения о неизвестных материалах. Это демонстрирует гибкость и адаптивность разработанного веб-приложения, которое позволяет ему адаптироваться к различным ситуациям и предоставлять пользователям соответствующую обратную связь, когда они пытаются получить доступ к несуществующим сайтам или ресурсам.

```

export const NotFound = () => {
  ScrollToTop();
  return (
    <Layout>
      <div className='NotFound'>
        <h1 className='NotFound_title'>Страница находится в разработке</h1>
        <img src={Image} alt='' />
        <Link to={'/'}>Назад</Link>
      </div>
    </Layout>
  );
};

```

Рисунок 10. Страница not found.

А также большое внимание было уделено созданию разработки многофункционального компонента на React, целью которого является подробная фильтрация предложений по услугам или товарам. Этот элемент обеспечивает простой в использовании интерфейс, позволяя пользователю устанавливать разнообразные критерии поиска, включая флажки, поля

для текста, календарь для определения дат и возможность загрузки файлов. Отличительной чертой компонента является его способность запоминать прогресс фильтрации пользователя благодаря локальному хранилищу, что позволяет без потери данных возвращаться к уже заданным параметрам.

Разработанный элемент управляет процессом выбора с помощью функций для перемещения между шагами фильтрации. Начальные состояния компонента устанавливаются на основе данных из локального хранилища, что делает процесс повторного выбора категорий и услуг, а также уточнения деталей задач, бюджетных ограничений и сроков выполнения, удобным для пользователя. В конечном счёте структура данного компонента стала выглядеть так как на рисунке 11.

```
<Layout>
  <div className='container_time'>
    <AsideNav />
    <section className='main_aside'>
      <div className='content'>
        <div className='filter_filling'>
          <form className='filter_form'>
            ...
          </form>
          <form className='filter_form--data'>
            ...
          </form>
          <div className='filter_Goback'>
            ...
          </div>
        </div>
      </div>
    </section>
  </div>
</Layout>
```

Рисунок 11. Страница фильтра.

**Заключение.** Стоит подчеркнуть, что проект "Masters.kz" олицетворяет собой современный подход к созданию онлайн-платформ, направленных на удовлетворение широкого спектра пользовательских потребностей в услугах и товарах. Основываясь на детальном анализе и изучении передовых практик в веб-разработке, в проекте были реализованы инновационные решения, способствующие улучшению пользовательского опыта и обеспечению высокой функциональности приложения.

Особенное внимание в процессе разработки было уделено интерфейсу и взаимодействию с пользователем, что выразилось в создании многофункционального компонента фильтрации, обладающего гибкими настройками и способного адаптироваться под индивидуальные предпочтения каждого пользователя. Это стало возможным благодаря применению компонентного подхода на базе React, который обеспечил модульность, повторное использование кода и высокую адаптируемость приложения.

Значительная работа была проделана и в области обеспечения безопасности и надежности веб-приложения, что особенно важно в контексте современных требований к

цифровым сервисам. Использование современных технологий шифрования и аутентификации позволило создать защищенную среду для пользователей "Masters.kz".

В заключение, проект "Masters.kz" демонстрирует успешное сочетание инновационных технологических решений с глубоким пониманием потребностей пользователей. Разработанное веб-приложение не только облегчает поиск и предложение услуг и товаров, но и способствует формированию доверительных и долгосрочных отношений с пользователями, что является ключом к успеху в современном цифровом мире.

**Список использованных источников:**

1. MDN web docs. [Электронный ресурс] – Режим доступа:  
[https://developer.mozilla.org/ru/docs/Web/Guide/CSS/Getting\\_started/What\\_is\\_CSS](https://developer.mozilla.org/ru/docs/Web/Guide/CSS/Getting_started/What_is_CSS)
2. Национальная библиотека им. Н.Э. Баумана Bauman National Library. [Электронный ресурс] – Режим доступа: [https://ru.bmstu.wiki/Сервлет\\_\(Java\)](https://ru.bmstu.wiki/Сервлет_(Java))
3. AWS [Электронный ресурс] – Режим доступа:  
<https://aws.amazon.com/ru/websites/>
4. Современный учебник JavaScript: сайт – URL: <https://learn.javascript.ru>
5. Learning React: A Hands-On Guide to Building Web Applications Using React and Redux
6. React-router-dom — The essentials Nora Mensah