

## РАЗРАБОТКА САЙТА «BEERJA»

**М. А. Каптюх, А. В. Стригалеv**

*ГyO «Институт бизнеса Белорусского государственного университета», г. Минск;  
mkaptyuh752@mail.ru, Strigaliou4working@gmail.com;  
науч. рук. – Е. Г. Гринеvич*

Научная работа на тему «Разработка сайта «BeerJa»» является актуальной для всех пользователей сети Интернет, которые употребляют алкогольные напитки, так как упрощают их выбор. Целью работы является изучение технологии разработки Web-сайтов, создание информационного продукта и организация доступа к нему пользователей. Разработанное приложение обладает эргономичным интерфейсом, реализованным на языке программирования JavaScript, и широкий спектр функциональных возможностей, представленных в нереляционной базе данных Mango DB Atlas.

**Ключевые слова:** разработка сайта; автоматизация; пользователь; javascript; react.js; node.js; mongodb; rest api; material ui.

В настоящее время интернет представляет собой высокоинтеллектуальную сферу, где происходит обмен, хранение и обработка огромного массива информации. Потребителем этих данных является пользователь, то есть человек.

Создание сайтов с отзывами и характеристиками товара той или иной категории – один из наиболее выгодных и перспективных инструментов онлайн бизнеса и разработки. На сегодняшний день системы управления сайтами дают возможность реализовывать любые требования пользователя.

Именно поэтому, в рамках научного исследования, нами был разработан сайт «BeerJa», на котором совершеннолетние пользователи могут получать информацию о различных видах пива, а также оставлять свои отзывы и комментарии. Актуальность нашей работы основывается на том, чтобы помочь с выбором тем людям, которые употребляют алкогольные напитки. Сделать их покупки более приятными и мобильными. Ведь на нашем сайте собраны и опубликованы отзывы пользователей, которые является важным критерием для выбора потенциальным клиентом данного товара.

По этой причине можно достигать не только внешних маркетинговых целей таких, как публикация отзывов в качестве рекламы, но и внутренних. Сбор и анализ информации по каналам обратной связи позволяет более точно определить целевую аудиторию, её желания и потребности.

Система управления сайтом предоставляет возможность добавления, редактирования и удаления содержимого страниц, баз данных и документов.

База данных для данного приложения представлена в JSON-формате нереляционной базы данных MongoDB Atlas [1].

Нереляционные базы данных используют разнообразные модели данных для доступа к данным и управления ими. Базы таких типов оптимизированы для приложений, которые работают с большим объемом данных, нуждаются в низкой задержке и гибких моделях. Все это достигается путем смягчения жестких требований к непротиворечивости данных, характерных для других типов БД.

В нашей работе предоставлено две сущности: USER (ПОЛЬЗОВАТЕЛЬ) И REVIEW (ОБЗОР). Созданы они были с помощью фреймворка mongoose. Нужно учитывать, что mongoDB также самостоятельно создает идентификатор для каждой записи. Идентификатор BSON со 12-битным значением, в котором 4 бита – это отметка времени (секунды с начала эпохи), 3 бита – идентификатор машины, 2 бита – идентификатор процессора, и трехбитный счетчик.

Обработка данных осуществлялась путем взаимодействия front-end с back-end через CRUD-операции, с последующим взаимодействием back-end с БД mongoDB Atlas. Например, Пользователь вводит свои регистрационные данные, которые, при срабатывании триггера обращаются к серверной части back-enda. Back-end проверяет валидность данных, и, при их валидности создание запись в базе данных. На front-end же, возвращается ответ от back-enda, в котором либо описана ошибка (если пользователь не ввел какое-то поле) или прилетает, что действие успешно выполнено [2].

Создание UX-UI дизайна осуществлялась путем использования библиотеки компонентом material UI для React.js. Так же использовался стиль CSS-In-JS для создания собственного дизайна внутри. jsx файлов.

В качестве среды разработки использован IDE WebStorm. Использован язык Javascript, база данных mongoDB. Использованы многочисленные библиотеки. Для front-enda был выбран фреймворк React.js, для back-enda Express.js.

На сайте присутствуют следующие модули:

- a) Страница регистрации пользователя;
- b) Каталог алкогольных напитков с обзорами;
- c) Личный кабинет пользователя;
- d) Страницы с отзывами.

Страница с регистрацией или авторизацией пользователя являются стартовыми страницами на нашем сайте, чтобы несовершеннолетние

пользователи не смогли просматривать его содержимое. На данной вкладке пользователь может запомнить свои данные (в таком случае они сохраняются в LocalStorage, если сохранять данные не нужно, то в SessionStorage). Также пользователю доступна функция восстановления пароля по почте.

Каталог алкогольных напитков включает в себя поиск пива по имени и ключевым словам. Весь каталог состоит из 400 экземпляров. Также можно включить фильтрацию по особенностям пива (алкоголь, горечь и цвет). Фильтрация происходит по принципу «больше, чем <значение>».

В личном кабинете пользователя данные берутся из данных MongoDB Atlas, фотография добавляется из облачного хранилища данных clouniary. Также есть возможность смены пароля [3].

На странице с отзывами пользователь может оставлять комментарии, которые будут появляться автоматически у всех юзеров, используя WebSockets.

Необычные решения с кодом включают в себя следующие позиции:

1) Создание jwt-стратегии для авторизации пользователя по генерируемым токенам:

```
const JwtStrategy = require("passport-jwt").Strategy;
const { ExtractJwt } = require("passport-jwt");
const mongoose = require("mongoose");

const User = mongoose.model("Users");
const keys = require("../config");

const opts = {};
opts.jwtFromRequest = ExtractJwt.fromAuthHeaderAsBearerToken();
opts.secretOrKey = keys.secretOrKey; // secret

module.exports = passport => {
  passport.use(
    new JwtStrategy(opts, (jwtPayload, done) => {
      User.findById(jwtPayload.id).then(user => {
        if (user) {
          return done(null, user);
        }
        return done(null, false);
      });
    })
  );
};
```

Рис. 1. Код разработки jwt-стратегии

## 2) Мейлер-функции:

```
class Mailer {
  constructor() {
    this.smtp = nodemailer.createTransport({ transporter: {
      service: "Gmail",
      auth: {
        user: "BeerMailer",
        pass: "123456789"
      },
      port: 465,
      secure: true
    });
  }

  send(data) {
    let mailOptions;
    switch (data.source) {
      case "email-verification": {
        mailOptions = {
          to: data.login,
          from: "vikadarova1337@gmail.com",
          subject: `Beer Shop | Email verification`,
          html: `Hello.<br> Confirm your email and activate account: ${data.login}<br><br>
                <a href="${data.link}" target="_blank">Click here</a>`
        };
        break;
      }
    }
  }
}
```

Рис. 2. Мейлер функция

## 3) Создание WebSocket сервера

```
const brewRepository = require("../repositories/brewRepository");

module.exports = io => {
  io.on("connection", socket => {
    socket.on("add-message", async id => {
      const brew = await brewRepository
        .find({ param: "_id", id })
        .populate("author", "firstName");
      io.sockets.emit("get_data", brew);
    });
  });
};
```

Рис. 3. Создание WebSocket сервера

В перспективе мы собираемся внедрить jwt-стратегию, так как это позволит генерировать access token для безопасной авторизации запросов. Добавить mailer-функции, чтобы у пользователей была возможность привязать свой email к аккаунту, восстановить аккаунт в случае утери пароля. А также усовершенствовать CRUD-запросы, чтобы оптимизировать наш сайт. Отдельное внимание хотим уделить SEO разработкам, чтобы продвинуть нашу страницу в ТОП Google и Яндекс, а также составить шаблоны генерации и формирования метатегов.

### Библиографические ссылки

1. Введение в MongoDB. Сайт о программировании; [Электронный ресурс]. – Режим доступа: <https://metanit.com/nosql/mongodb/1.1.php>. – Дата доступа: 04.04.2020.

2. Front-End разработка. Digital Marketing ROI; [Электронны ресурс]. – Режим доступа: <https://klondike-studio.ru/wiki/front-end/>. – Дата доступа: 07.04.2020.
- 3.Руководство по MongoDB. Записки задумчивого программиста; [Электронный ресурс]. Режим доступа: <https://proselyte.net/tutorials/mongodb/>. – Дата доступа: 06.04.2020.