Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

Отчет по лабораторной работе №0

Дисциплина: «Технологии сетевого программирования»

Тема «Проектирование приложения»

Выполнили: Зяббарова К.З,

Буянов С.А.

Группа: 6302-010302D

Самара, 2025

# **Концепт приложения**

Мы создадим SPA приложение - магазин кондитерских изделий, написанное на React и NodeJS+Express с использованием MongoDB.

SPA (Single Page Application) — это веб-приложение, которое загружает только одну HTML-страницу. При взаимодействии с пользователем оно динамически обновляет её содержимое без необходимости полной перезагрузки страницы.

Основная идея SPA заключается в том, что весь необходимый HTML, CSS и JavaScript загружаются на клиентскую сторону (браузер) один раз. После этого приложение общается с сервером через API (обычно REST) для загрузки данных по мере необходимости.

1. Главная страница. На ней представлены категории десертов для удобства поиска.
2. Доставка и оплата. В приложении можно реализовать функцию оформления удобной доставки (доставка, самовывоз), а также навигацию по местонахождению предприятия.
3. Контакты. На странице можно указать контактные данные и режим работы.
4. Вакансии. В приложении можно добавить раздел с вакансиями для работы в сфере кондитерских изделий.

# **Схема взаимодействия компонентов**

• Сервер - отвечает на запросы от клиента, осуществляя сохранение и получение данных из базы данных.

• Клиент - формирует запросы к серверу (например, на добавление, удаление или редактирование записи) и получает информацию из базы данных через сервер (например, для просмотра своих данных).

• База данных - хранит данные и предоставляет необходимую информацию клиенту.

# **Логическая схема базы данных**

1. **Product**

{

title: {

type: String,

required: true,

},

proteins:{

type: Number,

required: true,

},

fats:{

type: Number,

required: true,

},

carbohydrates:{

type: Number,

required: true,

},

photoPath:{

type: String,

required: true,

},

productID:{

type: String,

required: true,

},

energy: {

type: Number,

required: true,

},

expiration: {

type: Number,

required: true,

},

price:{

type: String,

required: true,

},

composition:{

type: String,

required: true,

},

box\_weight:{

type: Number,

required: true,

},

isActive:{

type: Boolean,

required: true,

}

}

1. **Order**

{ userId: {

type: String, required: true, ref: 'User'

},

orderId: {

type: String, required: true

},

purchasedProductlist: [

{

type: String, ref: 'Product'

}

],

orderTime:

{

type: String, default: Date.now

},

status: {

type: String, default: 'not confirmed'

}

}

1. **Token**

{

user: {

type: Schema.Types.ObjectId, ref: 'User'

},

refreshToken: {

type: String, required: true

}

}

1. **User**

{

email: {

type: String, unique: true, required: true

},

password: {

type: String, required: true

},

isActivated: {

type: Boolean, default: false

},

activationLink: {

type: String

},

role: {

type: String, default: 'user'

}

}

1. **Vacancy**

{

title: {

type: String,

required: true,

},

conditions:{

type: Array,

required: true,

},

duties:{

type: Array,

required: true,

},

requirements:{

type: Array,

required: true,

},

salary:{

type: Number,

required: true,

},

description:{

type: Number,

required: true,

},

isActive:{

type: Boolean,

required:true

}

}

# **Структура API**

|  |  |  |  |
| --- | --- | --- | --- |
| Действие | Метод | URL | Формат ответа |
| Получить все продукты | GET | /products | [ { "\_id": "654321abcdef01234567890",  "title": "Название продукта 1",  "proteins": 25,  "fats": 10,  "carbohydrates": 50,  "photoPath": "/images/product1.jpg",  "productID": "prod001",  "energy": 300,  "expiration": 365,  "price": "150",  "composition": "Состав продукта 1",  "box\_weight": 1000,  "isActive": true,  "\_\_v":0} ] |
| Получить количество продуктов | GET | /products/sum | { 15 } |
| Получить продукты из каталога | GET | /products/catalog | [  {  "\_id": "654321abcdef0123456789a",  "title": "Название продукта 16",  },  {  "\_id": "654321abcdef0123456789b",  "title": "Название продукта 17",  },  ] |
| Получить ID продукта | GET | /products/:id | { [  {  "\_id": "654321abcdef0123456789c",  "title": "Название конкретного продукта",  // … другие поля продукта  "productID": "ID продукта"  }  ] } |
| Обновить продукт | POST | /admin/update/product | {  "\_id": "mongodb\_object\_id", "title": "Новое название продукта",  "proteins": 30,  "fats": 12,  "carbohydrates": 55,  "photoPath": "/images/updated\_product.jpg",  "productID": "идентификатор\_продукта",  "energy": 320,  "expiration": 300,  "price": "175",  "composition": "Новый состав продукта",  "box\_weight": 1100,  "isActive": true,  "\_\_v": 0  } |
| Обновить вакансию | POST | /admin/update/vacancy | {  "\_id": "mongodb\_object\_id",  "title": "Новое название вакансии",  "conditions": ["Гибкий график"],  "duties": ["обязанность1", "обязанность2"],  "requirements": ["Опыт работы 3+ года", "..."],  "salary": "От n руб.",  "description": "Описание вакансии...",  "isActive": true,  "\_\_v": 0  } |
| Добавить вакансию | POST | /admin/add/vacancy | {  "\_id": "новый\_mongodb\_object\_id", (сгенерирован MongoDB)  "title": "Название новой вакансии",  "conditions": ["Дружный коллектив"],  "duties": ["...."],  "requirements": ["Знание"],  "salary": "От n руб.",  "description": "Описание новой вакансии",  "isActive": true,  "\_\_v": 0  } |
| Получить все вакансии | GET | /vacancies | [  {  "\_id": "654321abcdef01234567890",  "title": "Название вакансии",  "conditions": ["условия", "условия"],  "duties": ["обязанности", "обязанности"],  "requirements": ["требования", "требования"],  "salary": "n рублей",  "description": "описание",  "isActive": true,  "\_\_v": 0  },] |
| Получить вакансию по id | GET | /vacancies/:id | [  {  "\_id": "654321abcdef01234567890",  "title": "Название",  "conditions": ["условия", "условия"],  "duties": ["обязанности", "обязанности"],  "requirements": ["требования", "требования"],  "salary": "n рублей",  "description": "описание",  "isActive": true,  "\_\_v": 0  }  ] |
| Регистрация | POST | /registration | {  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5",  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5",  "user": {  "\_id": "654321abcdef01234567890",  "email": "user@example.com",  "isActivated": false,  "role": "user"  }  }  Авторизация: Post : '/login' : {  "accessToken": "...",  "refreshToken": "...",  "user": {  "\_id": "...",  "email": "...",  "isActivated": true,  "role": "..."  }  } |
| Логин пользователя | POST | /login | {  "accessToken": "...",  "refreshToken": "...",  "user": {  "\_id": "...",  "email": "...",  "isActivated": true,  "role": "..."  }  } |
| Выход пользователя | POST | /logout | { "message": "Успешный выход" } |
| Активация пользователя | POST | /activate/:link |  |
| Обновление токена | GET | /refresh | {  "accessToken": "...",  "refreshToken": "...",  "user": {  "\_id": "...",  "email": "...",  "isActivated": true,  "role": "..."  }  } |
| Получить всех пользователей | GET | /users | [  {  "\_id": "654321abcdef01234567890",  "email": "user1@example.com",  "isActivated": true,  "role": "user"  },  {  "\_id": "654321fedcba9876543210",  "email": "user2@example.com",  "isActivated": false,  "role": "admin"  }  // ... и так далее для каждого пользователя в базе данных  ] |
| Оформить заказ | POST | /orders | {  "userId": "654321abcdef01234567890",  "purchasedProductlist": ["part1", "part2"],  } |
| Получение заказов по ID | GET | /order/:id | {  "userId": "654321abcdef01234567890",  "purchasedProductlist": ["part1", "part2"],  } |
| Получить все заказы | GET | |  | | --- | |  |  |  | | --- | | /orders | | [  {  "userId": "654321abcdef01234567890",  "purchasedProductlist": ["part1", "part2"],  "orderTime": "2023-03-20T12:00:00Z",  "status": "not confirmed"  },  // ... и так далее для каждого заказа  ] |
| Изменение статуса заказа | POST | /orders/status | {  "orderId": "12345-abcde",  "status": "confirmed"  } |
| Добавить продукт | POST | /products | {  "newProduct": {  "title": "2пирожки",  "proteins": "0",  "fats": "60",  "carbohydrates": "100",  "photoPath": "false",  "productID": "01",  "energy": "250",  "expiration": "7",  "price": "80",  "composition": "мука, яйца, соль, сахар, разрыхлитель",  "box\_weight": "30",  "isActive": true  }  } |

1. **Стэк технологий.**

**Frontend:** React

**Backend:** Express, node.js, mongoose

**База данных:** MongoDB

Клиентская часть будет написана на React.

Серверная часть будет написана на Node.js, с использованием фреймворка Express.

1. **Ссылка на Git-репозиторий:**

<https://github.com/sbkz/ssausbkz>