Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Веб-технологии»

Отчет по курсовому проекту

по дисциплине «Инженерное проектирование»

Тема: «Сайт по заказу домашних тортов»

**Выполнил:**

Студент группы 181-321

Бычкова Е.С.

**Принял:**

Старший преподаватель

Даньшина М.В.

Москва 2020

ОГЛАВЛЕНИЕ

[1 ВВЕДЕНИЕ 3](#_Toc43741331)

[2 ОСНОВНАЯ ЧАСТЬ 4](#_Toc43741332)

[2.1 Анализ аналогов 4](#_Toc43741333)

[2.2 Проектирование 5](#_Toc43741334)

[2.2.1 Проектирование бизнес-процессов 5](#_Toc43741335)

[2.2.2 Проектирование модели прецедентов и модели классов 7](#_Toc43741336)

[2.2.3 Постановка основных требований 11](#_Toc43741337)

[2.2.4 Создание макета в Figma 15](#_Toc43741338)

[2.3 Разработка 20](#_Toc43741339)

[2.4 Тестирование 22](#_Toc43741340)

[2.5 Заполнение оценочного листа по проекту 25](#_Toc43741341)

[3 ЗАКЛЮЧЕНИЕ 28](#_Toc43741342)

[4 СПИСОК ИСТОЧНИКОВ 29](#_Toc43741343)

# ВВЕДЕНИЕ

Разрабатываемый проект: сайт по заказу домашних тортов «От Павловых».

Проблема: при создании домашнего бизнеса рано или поздно многие сталкиваются с рядом проблем:

1. Малая известность компании;
2. Неудобный формат взаимодействия с клиентами;
3. Неудобный формат работы с поступающей информацией о заказах и др.

Цель работы: создать сайт для заказа домашних тортов, который будет отображать всю необходимую для совершения целевого действия информацию пользователю, повысит удобство сбора требований к заказу, поспособствует увеличению числа клиентов.

Аналоги:

1. Кондитерский дом «DaDio»

Сайт: <https://dadio.ru/>

1. Кондитерская «Чудо-торт»

Сайт: <http://chudo-tort.tilda.ws/>

1. Кондитерская «Cake&Cake»

Сайт: <https://cakeandcake.ru/>

# ОСНОВНАЯ ЧАСТЬ

## 2.1 Анализ аналогов

Проанализируем сайт кондитерского дома «DaDio». Сайт – многостраничный. На главной странице отображается подробная информация о кондитерской. В шапке сайте – ссылки на каталог товаров, личный кабинет пользователя, информацию о контактах и об оплате. Перейдем в каталог тортов. На странице каталога тортов слева располагаются различные фильтры, также торты можно сортировать по различным показателям. Удобно то, что при нажатии на торт можно посмотреть более подробную информацию о нем. Торт в корзину может добавить каждый пользователь. При оформлении заказа необходимо в форме указать ФИО, телефон, email, адрес доставки, способ доставки. Цена за заказ пересчитывается в соответствии с выбранным способом доставки. Пользователь также может указать примечание/пожелания к заказу. Сайт достаточно удобен и прост в использовании.

Сайт кондитерской «Чудо-торт» - одностраничный. В меню размещены ссылки на различные блоки сайта. На сайте можно узнать информацию о кондитерской и о предоставляемых ею услугах, прочитать отзывы клиентов и просмотреть готовые работы (торты). Одним из преимуществ сайта является то, что пользователь может заказать торт за несколько минут. Для этого ему необходимо определиться с параметрами торта и заполнить очень короткую форму. Сайт очень простой, хорошо походит для домашних небольших кондитерских.

Проанализируем сайт «Cake&Cake». На главной странице расположен каталог тортов. В шапке сайта – ссылки на информацию о кондитерской, доставке и оплате, отзывы. Сразу можно отметить минусы сайта:

1. Множество небольших блоков с различной информацией о тортах путают пользователя;
2. Картинки и текст на сайте мелкие, также маленькие отступы.

Данный сайт не удобен в использовании, он является отличным примером того, как не надо делать сайты.

## 2.2 Проектирование

### 2.2.1 Проектирование бизнес-процессов

Начнем работу над проектом «Сайт по заказу домашних тортов». Целевая функция проекта – изготовление тортов по заказу клиентов. Чтобы разобраться в основной логике сайта, начнем с проектирования ключевых бизнес-процессов кондитерской. Опишем бизнес-процесс «Оформление заказа». Состав лиц, задействованных в данном бизнес-процессе: заказчик и администратор кондитерской. Входная информация бизнес-процесса «оформление заказа» - заявка клиента. Выходная информация – список заказчиков.

Построим структурную схему типа «черный ящик» для процесса оформления заказа (рис.2.2.1.1).



*Рис. 2.2.1.1 Реализация принципа «Черный ящик»*

Произведем декомпозицию бизнес-процесса оформления заказа.

Перечислим основные задачи процесса «оформление заказа»:

1. Задача создания заказа;

2. Задача рассмотрения заказа;

3. Задача расчета параметров заказа;

4. Задача подтверждения заказа, связи с клиентом;

5. Задача занесения заказчика в журнал заказов

Первую задачу выполняет клиент, он заполняет форму на сайте. Далее данные с формы (заявка на заказ) отправляются на рассмотрение. С этого момента в процесс включается администратор (работник кондитерской). После того, как данные с формы будут проанализированы, и, следовательно, будут рассчитаны основные параметры заказа, такие как цена и срок реализации, администратор должен подтвердить заказ клиента. После получения результата подтверждения заказа редактируется журнал заказов (клиент либо заносится в список, либо не заносится).

На данном этапе имеет смысл реализовать диаграмму IDEF0 бизнес-процесса оформление заказа (рис.2.2.1.2).



*Рис. 2.2.1.2 Диаграмма бизнес-процесса в нотации IDEF0*

После декомпозиции процесса «оформление заказа» рассмотрим одну из его функций – «рассмотрение заказа».

Этап 1. Внешняя сущность «Клиент» передаёт заказ на рассмотрение;

Этап 2. Администратор приступает к рассмотрению заказа:

- Сравнивается с таблицей цен для того, чтобы вычислить предполагаемую стоимость заказа»;

- Оценивает текущую занятость рабочих, просматривая журнал заказов, для того, чтобы вычислить срок реализации заказа

Этап 3. Администратор созванивается с клиентом, чтобы подтвердить заказ и сообщить параметры заказа.

Продемонстрируем данную инструкцию графически (рис.2.2.1.3).



*Рис. 2.2.1.3 Диаграмма бизнес-процесса в нотации DFD*

### 2.2.2 Проектирование модели прецедентов и модели классов

После проектирования основных бизнес-процессов необходимо точно определиться с основным функционалом сайта. Так как продажа домашних тортов – малый бизнес, семейный, то сайт должен быть простым, не иметь сложного функционала. Также сайт должен содержать всю необходимую информацию и быть удобным для пользователя. Чтобы было проще понять ключевой функционал сайта, разработаем модель прецедентов (рис.2.2.2.1).

Основные актеры: пользователь, работник кондитерской и администратор.

Действующее лицо заказчик может зайти на сайт, чтобы:

1. Оформить заказ:
   1. Заказ торта из каталога;
   2. Заказ торта по индивидуальным параметрам;
2. Посмотреть контактную информацию

Действующее лицо работник кондитерской может:

1. Войти;
2. Просмотреть список заказчиков;

Действующее лицо администратор может:

1. Войти;
2. Добавить/изменить/удалить различную информацию на сайте

Проанализировав данную модель прецедентов, можно выявить некоторые требования к функционалу сайта.

Требования к форме составления заказа:

1. Все необходимые поля должны быть заполнены;
2. Необходима проверка корректности заполнения полей (email, телефон)
3. Должна выводиться ошибка, предупреждающая пользователя о неполном/некорректном заполнении формы;
4. Форма должна отправляться в БД;

Требования к форме входа для работника/администратора:

1. Должна осуществляться проверка логина и пароля;
2. Должна выводиться ошибка при неправильном вводе логина/пароля.



*Рис.2.2.2.1 Модель прецедентов*

На следующем этапе проектирования построим модель классов (рис. 2.2.2.2), которая отобразит основную логику сайта и поможет понять, какие таблицы в базе данных будут необходимы для реализации основного функционала сайта.

Выделим четыре основные таблицы: «Cakes», «CustomerFirst», «CustomerSecond», «Workers» (таблицы могут добавляться при расширении функционала).

В «Cakes» будут храниться все торты из каталога. Поле «id» - уникальный идентификатор торта, поле «name» - название торта, «picture» - поле для хранения ссылки на изображение торта, «info» - основная информация о торте (состав и др.), «price» - поле, для хранения тортов.

Таблицы «CustomerFirst» и «CustomerSecond» необходимы для хранения списка заказчиков тортов из каталога и заказанных по индивидуальным предпочтениям. Таблица «CustomerFirst» содержит поля «name\_customer» - для хранения имени заказчика, «address» - адрес заказчика, «tel» - телефон заказчика, «email» - email заказчика, «cake\_name» - название заказанного торта, «cake\_price» - цена торта, «date» - дата, к которой необходимо изготовить торт. В таблицу же «CustomerSecond» помимо данных о заказчике будут заноситься такие данные, как выбранный корж для торта – поле «corz», выбранная начинка – «filling», количество слоев – «tiers», вес торта – «weight», пожелания/комментарии к торту – «wishes».

Таблица «Workers» нужна для хранения логина и пароля для работников кондитерской.



*Рис.2.2.2.2 Модель классов*

### 2.2.3 Постановка основных требований

На основе полученной выше информации составим список основных требований к сайту (табл.2.2.3.1).

*Табл.2.2.3.1 Список требований*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Требование** | **Функциональное/нефункциональное** | **Прецедент использования** | **Важность** | **Срочность** | **Приоритет** |
| Возможность посмотреть подробную информацию о каждом торте, представленном в галерее | Функциональное | Посетитель | Средне | Срочно | 2 |
| Возможность выбора разнообразных параметров торта в форме для заказа торта по индивидуальным параметрам | Функциональное | Важно | Срочно | 1 |
| Возможность заказать торт из галереи | Функциональное | Важно | Срочно | 1 |
| Возможность заказать торт по собственным параметрам | Функциональное | Важно | Срочно | 1 |

*Табл.2.2.3.1 Список требований (Продолжение)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Реализация формы для заказа готовых тортов | Функциональное | Форма | Важно | Срочно | 1 |
| Реализация формы для заказа тортов по индивидуальным параметрам | Функциональное | Важно | Срочно | 1 |
| Вывод ошибки при некорректном заполнении полей формы | Функциональное | Средне | Не срочно | 5 |
| Форма должна отправляться в БД | Функциональное | Важно | Срочно | 1 |
| Невозможность отправить форму при заполнении не всех необходимых полей | Функциональное | Средне | Не срочно | 5 |
| После отправки формы показывается сообщение об удачной отправке | Функциональное | Средне | Не срочно | 5 |

*Табл.2.2.3.1 Список требований (Продолжение)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Возможность просмотреть список заказчиков | Функциональное | Работник | Важно | Срочно | 1 |
| Возможность осуществлять поиск заказчиков по имени | Функциональное | Средне | Не срочно | 5 |
| Возможность распечатать список заказчиков | Функциональное | Неважно | Не срочно | 6 |
| Возможность корректировать список тортов на сайте (удалять/добавлять/редактировать) | Функциональное | Администратор | Важно | Срочно | 1 |

*Табл.2.2.3.1 Список требований (Продолжение)*

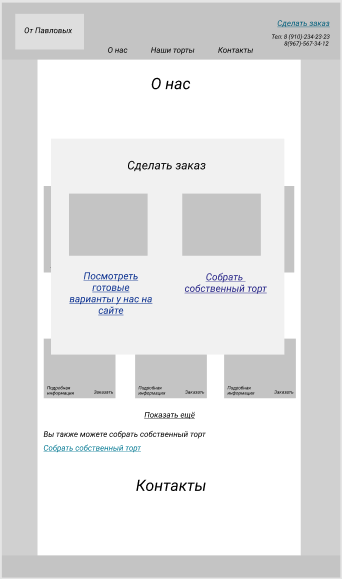
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Контактные телефоны и кнопка ЦД расположены в шапке справа | Нефункциональное | Общие требования | Средне | Не срочно | 5 |
| Данные о пользователе хранятся в БД | Функциональное | Важно | Срочно | 1 |
| Данные о тортах хранятся в БД | Функциональное | Важно | Срочно | 1 |
| Логотип расположен в левой стороне шапки | Нефункциональное | Неважно | Не срочно | 6 |
| Добавить ссылки на соц. Сети в блоке с контактами | Функциональное | Неважно | Не срочно | 6 |

### 2.2.4 Создание макета в Figma

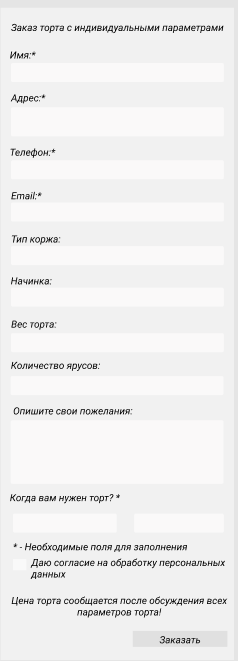
Создадим несколько макетов в Figma. Ниже представлены основные макеты для главной страницы (рис.2.2.4.1 – рис.2.2.4.4.)



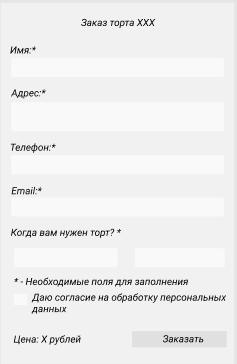
*Рис.2.2.4.1 Макет в Figma (1)*



*Рис.2.2.4.2 Макет в Figma (2)*



*Рис.2.2.4.3 Макет в Figma (3)*



*Рис.2.2.4.4 Макет в Figma (4)*

## Разработка

Для начала необходимо определиться с основными инструментами разработки. Данный проект будет разрабатываться с использованием Фреймворков Angular и Bootstrap. Также будут использованы такие технологии, как Webpack, firebase, git, Jasmine. Проект будет состоять из следующих компонентов Angular:

1. Header – для отображения шапки сайта;
2. Footer – для отображения подвала;
3. Login – для отображения страницы с входом;
4. Login-worker – для отображения формы входа работника;
5. Worker-panel/customer-first – для отображения заказчиков тортов из каталога;
6. Worker-panel/customer-second– для отображения заказчиков тортов по заказу;
7. Dashboard – для отображения дашборда;
8. Admin-login – для формы входа для администратора;
9. Admin-change – для отображения административной панели;
10. Main – содержит основной контент главной страницы сайта;
11. Cake-info – для отображения информации о каждом торте, при клике на него;
12. Form-own – содержит форму для заказа тортов по индивидуальным параметрам;
13. Form-ready – содержит форму для заказа тортов из списка.

Каталог тортов будет храниться в локальной базе данных (файл db.json). Так как данный проект учебный, будет полезно попробовать использовать новые технологии, поэтому кроме локальной базы данных попробуем использовать firebase. В firebase будут храниться остальные необходимые таблицы, а именно таблица заказов тортов из каталога, таблица с заказами тортов по индивидуальным параметрам, таблица с работниками, таблица с отзывами (функционал на будущее) и таблица с заявками на обратный звонок (функционал на будущее).

Для стилизации компонентов в основном будет использоваться Bootstrap. Однако, для придачи оригинальности сайту, некоторые стили будут задаваться вручную. Рассмотрим интересные решения оформления страниц сайта с помощью CSS, использованные в данном проекте.

1. Свойство «opacity» помогает управлять прозрачностью элемента;
2. Свойство «background-size: contain» масштабирует изображение с сохранением пропорций таким образом, чтобы картинка целиком поместилась внутрь блока;
3. Свойство «transition» задает эффект перехода между двумя состояниями элемента;
4. Свойство «transform» трансформирует элемент.
5. Свойство «justify-content: space-between» помогает равномерно распределить элементы контейнера вдоль главной оси контейнера.

## Тестирование

Проверим работоспособность сайта с помощью модульного тестирования. Создадим несколько тестов.

Тест 1.Проверка сервиса. Тестирование функции getAll(), которая должна возвращать все объекты Cakes.

it('should retrieve all cakes', () => {

const dummyCakes:CakeModule[] = [

{id:1, name:"Фруктовый", picture:"/assets/img/cake1.png", info:"Информация о торте", price:"1000 руб."},

{id:2, name:"Шоколадный", picture:"/assets/img/cake2.png", info:"Информация о торте", price:"800 руб."},

];

cakesService.getAll().then(cakes =>{

expect(cakes).toEqual(dummyCakes);

});

const request = httpTestingController.expectOne(`http://localhost:3000/${cakesService.url}`);

expect(request.request.method).toBe('GET');

request.flush(dummyCakes);

});

Тест 2.Проверка сервиса. Тестирование функции getById(), которая должна возвращать один объект по его id.

it('should retrieve cake by ID', () => {

const dummyCake:CakeModule[] = [

{id:1, name:"Фруктовый", picture:"/assets/img/cake1.png", info:"Информация о торте", price:"1000 руб."}

];

const id = 1;

cakesService.getById(id).then(cake =>{

expect(cake).toEqual(dummyCake);

});

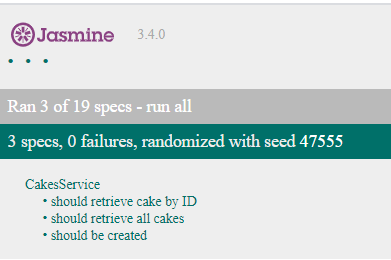
const request = httpTestingController.expectOne(`http://localhost:3000/${cakesService.url}/${id}`);

expect(request.request.method).toBe('GET');

request.flush(dummyCake);

});

Убедимся, что проверка сервиса прошла успешно (рис. 2.4.1)



*Рис.2.4.1 Модульное тестирование*

Также сайт успешно был протестирован вручную. Ниже приведен тест-план для страниц пользователя (табл.2.4.1). Более подробную информацию о тестировании можно посмотреть в файле «Стратегия тестирования.xlsx».

*Табл.2.4.1 Тест-план для страниц пользователя*

|  |  |  |
| --- | --- | --- |
| **Область функционала** | **Приоритет** | **Стратегия тестирования** |
| Меню + кнопки | 2 | Все кнопки кликабельны, переход по ссылкам осуществляется корректно. |
| Форма-заказ тортов по индивидуальным параметрам | 1 | Нельзя отправить пустую форму или форму с некорректными данными, с незаполненными обязательными полями. Форма отправляется в БД. После отправки формы выводится сообщение о корректной отправке. |

*Табл.2.4.1 Тест-план для страниц пользователя (Продолжение)*

|  |  |  |
| --- | --- | --- |
| Форма-заказ тортов из списка | 1 | Нельзя отправить пустую форму или форму с некорректными данными, с незаполненными обязательными полями. Форма отправляется в БД. После отправки формы выводится сообщение о корректной отправке. Корректное название торта и его цена также отправляются в БД вместе с данными о заказчике. |
| Страница с подробной информацией о торте | 2 | Информация о торте на странице верна для каждого торта (выводится в соответствии с БД "cakes"). |

## Заполнение оценочного листа по проекту

Оценочный лист по дисциплине «Инженерное проектирование»

ФИО Бычкова Елена Сергеевна

Группа 181-321

Название проекта Сайт по заказу домашних тортов

Баллы раздел 1 Баллы раздел 2 Итог Оценка

Раздел 1. Структура проекта

|  |  |  |  |
| --- | --- | --- | --- |
| № | Критерий | Отметка о выполнении Да/Нет  (заполняет студент) | Балл (заполняет преподаватель) |
| 1 | Работа выполнена индивидуально | Да |  |
| 2 | Работа выполнена в команде | Нет |  |
| 3 | Есть база данных. Минимум 6 таблиц, все таблица содержат нетестовые записи, минимум 10 записей в каждой таблице (кроме  таблиц-справочников). | Да |  |
| 4 | Многостраничный сайт/ приложение с несколькими экранами (минимум 4 страницы/экрана) | Да |  |
| 5 | Реализован CRUD | Да |  |
| 6 | Реализован вывод агрегирующей информации | Да |  |
| 7 | Реализован интернет-магазин | Да |  |
| 8 | Реализован интернет-портал | Нет |  |
| 9 | Реализована минимум 3 типа пользователей системы | Да |  |
| 10 | Реализована административная часть | Да |  |
| 11 | Реализован дашборд | Да |  |
| 12 | Используется фреймворк | Да |  |
| 13 | Реализована интеграция по API | Да |  |

Раздел 1. Структура проекта (Продолжение)

|  |  |  |  |
| --- | --- | --- | --- |
| 14 | Реализована фильтрация | Да |  |
| 15 | Реализован поиск | Да |  |
|  | Итого (1 Да = 1 балл) | 13 |  |

Раздел 2. Реализация проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Критерий | Балл max | Отметка о | Балл |
|  |  |  | выполнении | (заполняет |
|  |  |  | (заполняет | преподаватель) |
|  |  |  | студент) |  |
| 1 | Проект выложен на сервере fit.mospolytech.ru | 2 | 0 |  |
| 2 | Создать репозиторий проекта, вести разработку в нем с использованием веток. | 2 | 2 |  |
| 3 | Создать лендинг проекта с описанием проблемы, используемых технологий, команды, результатов работы, встроенная презентация и видео. | 3 | 3 |  |
| 4 | Разработка адаптивной верстки - минимум для трех размеров экранов. | 3 | 3 |  |
| 5 | Код проходит валидацию (HTML/CSS) | 2 | 2 |  |
| 6 | Богатый CSS код (различные типы селекторов, возможности оформления текстов, изображений, фона, форм, таблиц и пр.) - включить в отчет минимум 5 современных “решений” для оформления страницы. | 2 | 2 |  |
| 7 | Создание страницы для печати (минимум одна страница, которую по смыслу проекта возможно пользователю придется распечатать) | 2 | 2 |  |
| 8 | Использование внешних CSS и JS библиотек | 5 | 5 |  |
| 9 | Использование технологий автоматизированного тестирования проекта | 5 | 5 |  |
| 10 | Использование технологий ручного тестирования проекта | 5 | 5 |  |

Раздел 2. Реализация проекта (Продолжение)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | Использование Webpack | 5 | 5 |  |
| 12 | Использование методологии БЭМ | 5 | 2 |  |
| 13 | Использование препроцессора | 5 | 0 |  |
| 14 | Использование GraphQL | 10 | 0 |  |
| 15 | Single Page Application (React, Angular, Vue.js) | 10 | 10 |  |
| 16 | Разработка макета для проекта в Figma | 5 | 5 |  |
| 17 | User friendly дизайн | 5 | 5 |  |
| 18 | Посещение профильных мероприятий крупных ИТ компаний + внедрение что узнали на мероприятии в проект (допускается просмотр онлайн трансляции). | 5 | 5 |  |
|  | Итого | 81 | 61 |  |

Раздел 3. Документация

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Критерий | Балл max | Отметка о выполнении (заполняет студент) | Балл (заполняет преподаватель) |
| 1 | Анализ аналогов (поиск отечественных и зарубежных сайтов, анализ структуры сайта, юзабилити, функциональности) | 5 | 5 |  |
| 2 | Проектирование (обзор кейсов использования сайтов, CJM, выбор функционала, проектирование бизнес-процессов и т.д.) | 5 | 5 |  |
| 3 | Разработка и тестирование (архитектура проекта, структура базы данных, типовые запросы к БД, алгоритмы, ПМИ и т.д.) | 5 | 5 |  |
| 4 | Заполнение оценочного листа по проекту (см. шаблон) | 5 | 5 |  |

# ЗАКЛЮЧЕНИЕ

Результаты работы:

1. Создан сайт, позволяющий пользователям заказать желаемый торт, а работникам кондитерской - быстро и в удобном формате собрать требования к заказу;
2. Отточены навыки работы над проектом, полученные за 2 года обучения в университете;
3. Получены новые знания и опыт в ходе работы над проектом.

Ссылки на итоговые материалы:

Репозиторий проекта: <https://github.com/Bychkova-Elena/WebsiteForOrderingCakes>

Вход для работника:

Логин: worker@gmail.com

Пароль: worker

Вход для администратора:

Логин: admin@gmail.com

Пароль: admin

Firebase: <https://console.firebase.google.com/u/0/project/order-cakes-9ac74>

Лендинг проекта: <http://landing.std-716.ist.mospolytech.ru/>

Figma: [ссылка](https://www.figma.com/file/Jvxug6WzfIluyij8DQDMvW/%D0%94%D0%BE%D0%BC%D0%B0%D1%88%D0%BD%D0%B8%D0%B5-%D1%82%D0%BE%D1%80%D1%82%D1%8B-%D0%BD%D0%B0-%D0%B7%D0%B0%D0%BA%D0%B0%D0%B7?node-id=0%3A1)

# СПИСОК ИСТОЧНИКОВ

1. [Angular Docs](https://angular.io/docs)
2. [Bootstrap Documentation](https://getbootstrap.com/docs/4.5/getting-started/introduction/)
3. [Firebase Docs](https://firebase.google.com/docs)
4. [Metanit.com](https://metanit.com/)
5. [Htmlbook.ru](http://htmlbook.ru/)
6. MDN
7. [Habr.com](https://habr.com/)
8. [WebforMyself](https://webformyself.com/)
9. [Medium.com](https://medium.com/)