

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»»

ИНЖЕНЕРНО-ЭКОНОМИЧЕСКИЙ ИНСТИТУТ

**КУРСОВАЯ РАБОТА**

по дисциплине «Проектирование сетевых и мобильных приложений»

Тема: «Форма заявки на закупку мебели»

Студент группы ИЭс-61-21 \_\_\_\_\_ Реймер В.А.  
(Ф.И.О.)

Руководитель \_\_\_\_\_ Кондратьев К.В.  
(Ф.И.О.)

Сдана на проверку		
Возвращена на доработку		
Допущена к защите		
Оценка		

Москва 2024

## Оглавление

ВВЕДЕНИЕ .....	3
ГЛАВА 1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ .....	4
ГЛАВА 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ .....	10
ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ .....	13
ЗАКЛЮЧЕНИЕ .....	17
ПРИЛОЖЕНИЕ А .....	18

## ВВЕДЕНИЕ

Современные интернет-технологии значительно упрощают и ускоряют процесс взаимодействия между компаниями и клиентами. Одним из таких инструментов являются веб-формы, которые позволяют пользователям быстро и удобно отправлять свои запросы, заявки и заказы.

Постановка задачи:

В контексте мебельного магазина создание эффективной формы заявки на закупку мебели становится ключевой задачей. Такая форма должна быть интуитивно понятной и легко заполняемой.

Актуальность темы:

Актуальность данной темы обусловлена рядом факторов:

Пользователи предпочитают простые и понятные формы для совершения покупок или заказов. Чем легче и удобнее заполнение формы, тем выше вероятность завершения сделки.

Магазины и компании, предоставляющие услуги через интернет, стремятся к автоматизации процесса приема заказов и обработки заявок. Это позволяет сократить время на обслуживание клиентов и повысить общую эффективность бизнеса.

В условиях острой конкуренции важно предоставлять клиентам максимально удобные сервисы. Успешные интернет-магазины и компании активно развивают и совершенствуют свои онлайн-формы, чтобы привлечь и удержать клиентов.

Все эти факторы подчеркивают значимость разработки эффективной формы заявки на закупку мебели для мебельных магазинов, что делает эту тему актуальной и востребованной в современном бизнесе.

## ГЛАВА 1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ

Разработка полноценного веб-сервиса, такого как форма заявки на закупку мебели, включает в себя множество технологий и инструментов. Ниже представлен обзор нескольких ключевых технологий и их применимость для данного сервиса.

Фронтенд:

HTML/CSS/JavaScript: Основные технологии для разработки пользовательского интерфейса (UI). HTML используется для структуры страницы, CSS для оформления и стилей, JavaScript для интерактивности и динамических эффектов.

React, Vue.js, Angular: Модерные фреймворки и библиотеки для разработки пользовательских интерфейсов. Для формы заявки на закупку мебели, любой из них подойдет, но у каждого есть свои особенности:

React: Гибкий и мощный, но требует больше настройки и знаний о JavaScript.

Vue.js: легко освоить и использовать, имеет понятный синтаксис и небольшой размер.

Angular: Полноценный фреймворк, предоставляющий множество инструментов для разработки, но может быть сложен для новичков.

Бэкенд:

Node.js: Платформа для выполнения JavaScript на сервере. Отлично подходит для создания масштабируемых веб-приложений.

Express.js: Минималистичный и гибкий веб-фреймворк для Node.js. Идеально подходит для создания API и обработки HTTP-запросов.

Django, Flask (Python): если вы предпочитаете Python, то эти фреймворки обеспечат вас высокой производительностью и удобством разработки.

Базы данных:

MongoDB: Гибкая NoSQL база данных, которая хорошо масштабируется и позволяет хранить документы в формате JSON.

MySQL, PostgreSQL: Реляционные базы данных с богатым набором функций и возможностями для обеспечения целостности данных.

Валидация данных:

Joi (Node.js): Мощная библиотека для валидации данных в Node.js, которая обеспечивает гибкую настройку правил валидации.

Validator.js (JavaScript): Легковесная библиотека для валидации данных в браузере и Node.js.

Маскирование полей:

Vue Input Mask, React Input Mask: Библиотеки для маскирования ввода в полях формы. Они обеспечивают удобную и гибкую настройку формата ввода для различных типов данных, таких как номера телефонов, даты и т.д.

По сравнению с фреймворками React, Vue.js и Angular, для данной формы заявки на закупку мебели можно выбрать любой из них, так как они обеспечивают необходимый функционал для создания пользовательского интерфейса. Однако выбор фреймворка может зависеть от предпочтений разработчика, его уровня опыта и требований к проекту. Vue.js обычно считается более простым в изучении и использовании для небольших проектов, в то время как Angular может быть предпочтительным для крупных и масштабируемых приложений, а React обладает большим сообществом и экосистемой библиотек.

Аутентификация и авторизация:

JWT (JSON Web Tokens): Эффективный механизм аутентификации для защиты конфиденциальных данных и реализации авторизации в веб-приложениях. JWT может быть использован для создания токенов с различными уровнями доступа к ресурсам.

Тестирование:

Jest, Mocha, Chai (для JavaScript): Популярные библиотеки для тестирования кода в JavaScript. Jest часто используется с React, Mocha и Chai предоставляют широкий набор инструментов для тестирования любых типов приложений.

Pytest (для Python): Удобный и простой в использовании фреймворк для тестирования приложений на Python.

Преимущества и недостатки различных технологий могут зависеть от конкретных требований и условий проекта. Например, React обеспечивает высокую производительность и гибкость, но требует больше усилий для изучения и настройки. Vue.js, с другой стороны, предлагает простоту и интуитивный синтаксис, но может иметь меньшее сообщество и экосистему библиотек. Angular предоставляет всеобъемлющий набор инструментов для разработки, но может быть сложен для новичков. При выборе технологий для разработки сервиса важно учитывать требования проекта, опыт команды разработчиков и потенциальную масштабируемость приложения.

Использование системы контроля версий:

Для эффективного управления кодом и совместной работы разработчиков рекомендуется использовать систему контроля версий, такую как Git.

Git обеспечивает отслеживание изменений, ветвление и слияние кода, а также управление версиями проекта, что упрощает разработку и обеспечивает сохранность кодовой базы.

Непрерывная интеграция и развертывание:

Для автоматизации процесса сборки, тестирования и развертывания приложения рекомендуется использовать инструменты непрерывной интеграции и развертывания, такие как Jenkins, Travis CI или GitHub Actions.

Непрерывная интеграция и развертывание позволяют быстро и эффективно внедрять изменения в продуктивную среду, улучшая качество и производительность приложения.

Масштабирование и оптимизация производительности:

При проектировании архитектуры приложения необходимо учитывать возможность масштабирования и оптимизации производительности, чтобы обеспечить стабильную работу при росте нагрузки.

Использование кэширования, горизонтального и вертикального масштабирования, а также оптимизация запросов к базе данных помогут повысить производительность и масштабируемость приложения.

Сравнительный анализ фреймворков для фронтенда:

React: Один из самых популярных фронтенд фреймворков, предоставляющий большую гибкость и производительность благодаря виртуальному DOM и мощному сообществу. React использует JSX (расширение синтаксиса JavaScript), что упрощает разработку компонентов интерфейса.

Vue.js: Простой и гибкий фреймворк, который предоставляет интуитивно понятный API для создания интерфейсов. Vue.js обеспечивает простоту в использовании и быструю скорость разработки благодаря своей простой структуре и реактивным компонентам.

Angular: Полнофункциональный фреймворк, разработанный Google, который предоставляет всеобъемлющие инструменты для разработки сложных веб-приложений. Angular имеет высокую производительность благодаря статической типизации и встроенной системе модулей.

Выбор подходящего фреймворка:

Если требуется высокая производительность и гибкость, и команда разработчиков имеет опыт работы с JavaScript, то React может быть хорошим выбором.

Для небольших проектов с ограниченным бюджетом и неопытных разработчиков Vue.js может предложить простое и эффективное решение.

Angular подходит для крупных и сложных веб-приложений, где важны стабильность, безопасность и масштабируемость.

В результате анализа выявлены положительные стороны и недостатки рассматриваемых технологий:

React:

Положительные стороны:

Гибкость и масштабируемость: React позволяет разрабатывать масштабируемые приложения любой сложности благодаря модульной структуре компонентов.

Виртуальный DOM: Эффективное использование виртуального DOM обеспечивает быструю отрисовку интерфейса и повышает производительность приложения.

Большое сообщество: React имеет широкое сообщество разработчиков и обширный выбор библиотек и инструментов для разработки.

Недостатки:

Изучение кривой обучения: Некоторые концепции, такие как JSX и управление состоянием, могут потребовать времени для изучения, особенно для новичков.

Нет встроенных инструментов: React является библиотекой для построения пользовательского интерфейса, поэтому для решения других задач, таких как маршрутизация и управление состоянием, требуются дополнительные библиотеки.

Vue:

Положительные стороны:

Простота использования: Vue обладает интуитивно понятным синтаксисом и простой структурой, что делает его привлекательным для начинающих разработчиков.

Реактивность: Vue предлагает простую и эффективную систему реактивности, что упрощает отслеживание и обновление данных в приложении.

Официальные плагины: Возможность использовать официальные плагины Vue, такие как Vue Router и Vuex, делает разработку приложений более удобной и эффективной.

Недостатки:



Меньшее сообщество: В сравнении с React и Angular, сообщество Vue менее обширно, что может привести к меньшему количеству доступных ресурсов и библиотек.

Недостаточная поддержка корпоративными компаниями: В сравнении с Angular, Vue имеет меньшую поддержку со стороны крупных корпораций, что может вызвать определенные затруднения при построении крупных приложений.

Angular:

Положительные стороны:

Полноценный фреймворк: Angular предоставляет все необходимые инструменты для разработки полноценных приложений, включая маршрутизацию, HTTP-запросы и управление состоянием.

Строгая типизация: Строгая типизация TypeScript и инструменты статического анализа кода позволяют выявлять ошибки на ранних этапах разработки.

Широкий набор инструментов: Angular предлагает широкий набор инструментов, включая Angular CLI, Angular Material и Angular Universal, что упрощает создание и развертывание приложений.

Недостатки:

Сложность: Angular обладает высоким порогом входа из-за сложности некоторых концепций и большого количества абстракций, что может затруднить начинающим разработчикам изучение и использование фреймворка.

Большой размер: Приложения, разработанные на Angular, могут иметь большой размер из-за встроенных инструментов и зависимостей, что может негативно сказаться на производительности.

## ГЛАВА 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Цель - разработка веб-приложения, с помощью которого можно заказать мебель.

Функциональные требования:

Форма заказа:

Пользователи могут заполнить форму заказа, включая поля для имени, контактной информации, адреса доставки и деталей заказанной мебели.

Форма должна быть интуитивно понятной и легко заполняться.

Каталог мебели:

Пользователи могут просматривать каталог мебели с описанием, изображениями и ценами.

Нефункциональные требования:

Производительность:

Веб-приложение должно обеспечивать быструю загрузку страниц и быстрое выполнение операций, таких как поиск и обработка заказов.

Масштабируемость:

Система должна быть способной масштабироваться для поддержки большого числа одновременных пользователей и заказов

Легкость использования:

Интерфейс приложения должен быть интуитивно понятным и легким для использования как для опытных, так и для неопытных пользователей.

Надежность:

Система должна быть стабильной и надежной, минимизируя возможность сбоев и потери данных.

Таблица 1. Структура формы

Поле	Тип	Обязательное	Правильность данных	Комментарий
Название мебели	Текстовое поле	Да	Нет	Введите название желаемой мебели
Тип мебели	Текстовое поле	Да	Нет	Укажите тип мебели (стол, стул, шкаф и т.д.)
Ширина	Текстовое поле	Да	Да	Укажите ширину мебели в сантиметрах
Длина	Текстовое поле	Да	Да	Укажите длину мебели в сантиметрах
Высота	Текстовое поле	Да	Да	Укажите высоту мебели в сантиметрах
Цвет	Текстовое поле	Нет	Нет	Укажите желаемый цвет мебели
Материал	Текстовое поле	Нет	Нет	Укажите желаемый материал мебели
Количество	Числовое поле	Да	Да	Укажите количество единиц мебели
Дата доставки	Поле для даты	Да	Да	Укажите дату, когда вы хотите получить мебель
Адрес доставки	Текстовое поле	Да	Нет	Укажите адрес доставки мебели
Имя	Текстовое поле	Да	Нет	Введите ваше имя
E-mail	Текстовое поле	Да	Да	Введите ваш email в формате example@example.ru
Телефон	Текстовое поле	Да	Да	Введите номер телефона в формате +7(xxx)-xxx-xx-xx, маска проставляется сама с помощью JavaScript

Структура формы представлена в таблице 1.

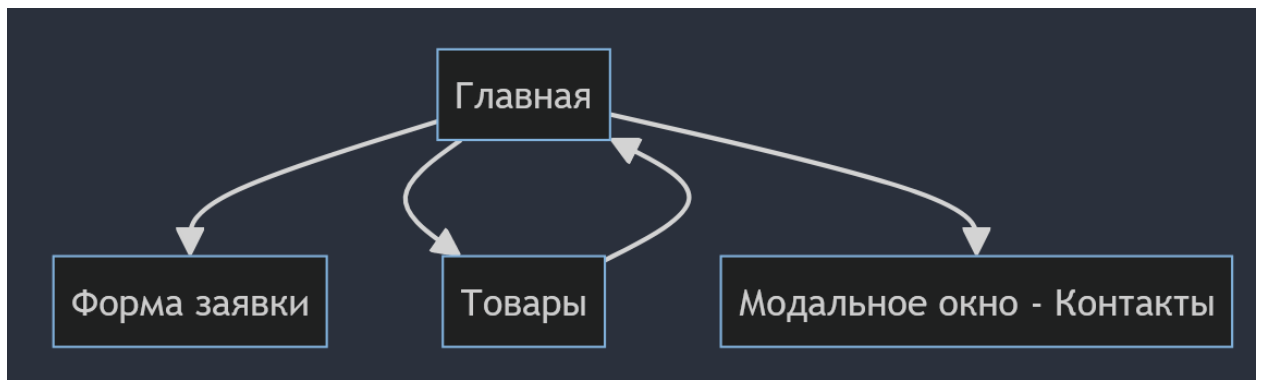


Рисунок 1. Структура приложения

Структура приложения представлена на Рисунке 1.

### ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ

Маска для номера телефона:

Используется библиотека IMask для создания маски для номера телефона в поле ввода. Маска определяет формат номера телефона, например, "+7(000)000-00-00".

Функция show\_hide\_password:

Эта функция позволяет пользователю переключать видимость пароля в поле ввода. При нажатии на элемент с определенным идентификатором (target) функция изменяет тип поля ввода с "password" на "text" и наоборот. Это обычно используется для обеспечения возможности скрытия и отображения пароля в текстовом виде.

Отправка формы:

Данный код добавляет обработчик события submit к форме с идентификатором purchaseForm. При отправке формы пользователю будет показано сообщение об успешной отправке формы.

Форма заявки на закупку мебели (Рисунок 2) содержит следующие поля:

Название мебели: Поле для ввода названия мебели (обязательное).

Тип мебели: Поле для ввода типа мебели (обязательное).

Ширина, Длина, Высота: Поля для ввода размеров мебели в сантиметрах (обязательные).

Цвет: Поле для ввода цвета мебели.

Материал: Поле для ввода материала мебели.

Количество: Поле для ввода количества единиц мебели (обязательное).

Дата доставки: Поле для выбора даты доставки (обязательное).

Адрес доставки: Поле для ввода адреса доставки (обязательное).

Имя: Поле для ввода имени заказчика (обязательное).

E-mail: Поле для ввода адреса электронной почты заказчика (обязательное, должно соответствовать формату e-mail).

Телефон: Поле для ввода номера телефона заказчика (обязательное, должно быть в формате "+7(xxx)-xxx-xx-xx", маска проставляется сама с помощью JavaScript).

Отправить заявку: Кнопка для отправки данных формы.

Если ввести верные данные и нажать на кнопку «Отправить заявку», то откроется окно с сообщением «Форма успешно отправлена» (Рисунок 3).

На Рисунках 4, 5, 6 представлены возможные сценарии, когда пользователь некорректно заполнил поля, в таком случае форма подскажет, где пользователю нужно изменить данные.

В Приложении А представлен листинг формы, состоящий из файлов form\_zakaz.html, form\_styles.css и script\_zakaz.js.

The image shows a web form titled "Форма заявки на закупку мебели" (Furniture Purchase Form) set against a light purple background. The form itself has a light green header and a white body with green borders. It contains the following fields: "Название мебели:" (Furniture name), "Тип мебели:" (Furniture type), "Ширина:" (Width) with a unit icon, "Длина:" (Length) with a unit icon, "Высота:" (Height) with a unit icon, "Цвет:" (Color), "Материал:" (Material), "Количество:" (Quantity) with a unit icon, "Дата доставки:" (Delivery date) with a date picker icon, "Адрес доставки:" (Delivery address), "Имя:" (Name), "Username(Если зарегистрированы, для начисления бонусов):" (Username, if registered, for bonus calculation) with a "Введите Username" (Enter Username) label and a small orange icon, "Email:", and "Телефон:" (Phone). At the bottom of the form is an orange button labeled "Отправить заявку" (Send request).

Рисунок 2. Форма заявки на закупку мебели

Форма заявки на закупку мебели

Название мебели:  
Угловой шкаф-купе Марсип

Тип мебели:  
шкаф

Ширина:  
150

Длина:  
300

Высота:  
200

Цвет:  
Черный

Материал:  
ЛДСП

Количество:  
1

Дата доставки:  
21.09.2024

Адрес доставки:  
г. Москва, ул. Пречистая, д. 76

Имя:  
Владислав Реймер

Удостоверение(Если зарегистрированы, для начисления бонусов):  
\*\*\*\*\*

Е-mail:  
v.reimer@yandex.ru

Телефон:  
+7(999)123-22-11

Отправить заявку

127.0.0.1:5555  
Форма успешно отправлена!

Рисунок 3. Форма успешно отправлена

Форма заявки на закупку мебели

Название мебели:  
Поиск/Введите название или код: \_\_\_\_\_

Ширина:  
150

Длина:  
300

Высота:  
200

Цвет:  
Черный

Материал:  
ЛДСП

Количество:  
1

Дата доставки:  
21.09.2024

Адрес доставки:  
г. Москва, ул. Пречистая, д. 76

Имя:  
Владислав Реймер

Удостоверение(Если зарегистрированы, для начисления бонусов):  
\*\*\*\*\*

Е-mail:  
v.reimer@yandex.ru

Телефон:  
+7(999)123-22-11

Отправить заявку

Рисунок 4. Пример с пустым обязательным полем

**Форма заявки на закупку мебели**

Название мебели:  
Угловой шкаф-купе Маркет

Тип мебели:  
Шкаф

Ширина:  
150

Длина:  
300

Высота:  
200

Цвет:  
Черный

Материал:  
ЛДСП

Количество:  
1

Дата доставки:  
23.05.2018

Адрес доставки:  
г. Москва, ул. Пречистая, д. 76

Имя:  
Владислав Романов

Укажите(Если зарегистрированы, для начисления бонусов):  
\*\*\*\*\*

Е-mail:  
v.romanov@yandex.ru

Телефон:  
+7(999)123-22-11

Отправить заявку

Рисунок 5. Пример поля, в котором обязательно числовое значение

**Форма заявки на закупку мебели**

Название мебели:  
Угловой шкаф-купе Маркет

Тип мебели:  
Шкаф

Ширина:  
150

Длина:  
300

Высота:  
200

Цвет:  
Черный

Материал:  
ЛДСП

Количество:  
1

Дата доставки:  
23.05.2018

Адрес доставки:  
г. Москва, ул. Пречистая, д. 76

Имя:  
Владислав Романов

Укажите(Если зарегистрированы, для начисления бонусов):  
\*\*\*\*\*

Е-mail:  
v.romanov@yandex.ru

Телефон:  
+7(999)123-22-11

Отправить заявку

Рисунок 6. Пример некорректного ввода e-mail



## ЗАКЛЮЧЕНИЕ

Пользователи предпочитают простые и удобные формы, компании стремятся к автоматизации процесса приема заказов, а конкуренция на рынке требует предоставления клиентам максимально удобных сервисов. При этом важно обеспечить высокий уровень безопасности передачи информации через веб-формы, учитывая растущую озабоченность пользователей вопросами конфиденциальности и защиты данных.

Таким образом, разработка эффективной формы заявки на закупку мебели становится ключевой задачей для мебельных магазинов, что делает эту тему актуальной и востребованной в современном бизнесе.

## ПРИЛОЖЕНИЕ А

### form\_zakaz.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Форма заявки на закупку мебели</title>
  <link rel="stylesheet" type="text/css" href="form_styles.css">
</head>
<body>
  <h1 class="form-title">Форма заявки на закупку мебели</h1>

  <form id="purchaseForm">
    <label for="furnitureName"><b>Название мебели:</b></label><br>
    <input type="text" id="furnitureName" name="furnitureName"
list="furnitureNameList" required><br>

    <datalist id="furnitureNameList">
      <option value="Угловой диван Стингер">
      <option value="Кресло-кровать Айви">
      <option value="Письменный стол Алтай">
      <option value="угловой шкаф-купе Маршал">
      <option value="Обеденная группа Апель">
      <option value="Обувница Глосс"></option>
    </datalist>

    <label for="furnitureType"><b>Тип мебели:</b></label><br>
    <input type="text" id="furnitureType" name="furnitureType"
list="furnitureTypeList" required><br>

    <datalist id="furnitureTypeList">
      <option value="Диван">
      <option value="Кресло-кровать">
      <option value="Стол">
      <option value="Шкаф">
      <option value="Обеденная группа">
      <option value="Обувница"></option>
    </datalist>

    <label for="width"><b>Ширина:</b></label><br>
    <input type="number" id="width" name="width" required><br>

    <label for="length"><b>Длина:</b></label><br>
    <input type="number" id="length" name="length" required><br>

    <label for="height"><b>Высота:</b></label><br>
    <input type="number" id="height" name="height" required><br>

    <label for="color"><b>Цвет:</b></label><br>
    <input type="text" id="color" name="color" list="furnitureColorList"><br>
```

```

        <datalist id="furnitureColorList">
            <option value="Белый">
            <option value="Чёрный">
            <option value="Синий">
            <option value="Красный">
            <option value="Серый">
            <option value="Коричневый"></option>
        </datalist>

        <label for="material"><b>Материал:</b></label><br>
        <input type="text" id="material" name="material"
list="furnitureMaterialList"><br>

        <datalist id="furnitureMaterialList">
            <option value="МДФ">
            <option value="ДСП">
            <option value="ЛДСП">
        </datalist>

        <label for="quantity"><b>Количество:</b></label><br>
        <input type="number" id="quantity" name="quantity" required><br>

        <label for="deliveryDate"><b>Дата доставки:</b></label><br>
        <input type="date" id="deliveryDate" name="deliveryDate" required><br>

        <label for="deliveryAddress"><b>Адрес доставки:</b></label><br>
        <input type="text" id="deliveryAddress" name="deliveryAddress"
required><br>

        <label for="contactName"><b>Имя:</b></label><br>
        <input type="text" id="contactName" name="contactName" required><br>

        <div class="password">
            <label for="password-input"><b>Username(Если зарегистрированы, для
начисления бонусов):</b></label><br>
            <input type="password" id="password-input" placeholder="Введите
Username" name="password">
            <a href="#" class="password-control" onclick="return
show_hide_password(this);"></a>
        </div>

        <label for="contactEmail"><b>E-mail:</b></label><br>
        <input type="email" id="contactEmail" name="contactEmail" required><br>

        <label for="contactPhone"><b>Телефон:</b></label><br>
        <input type="tel" id="contactPhone" name="contactPhone" required>

        <input type="submit" value="Отправить заявку">
    </form>

```

```

    <script src="https://cdn.jsdelivr.net/npm/imask"></script>
    <script src="../scripts/script_zakaz.js"></script>
</body>
</html>

```

## form\_styles.css

```

/* Стили для Body */
body {
    font-family: Arial, sans-serif;
    /* Создает градиентный фон для всего документа */
    background: linear-gradient(to right, #ffffff, #e1e1e1);
    /* Устанавливает цвет текста для всего документа */
    color: #000000;
    /* Размер шрифта */
    font-size: 15px;
    /* Фон */
    background-image: url("../images/background-form.jpg");
}

/*Заголовок формы*/
.form-title {
    /* Выравнивание по центру */
    text-align: center;
    /* Размер шрифта */
    font-size: 45px;
}

/* Стили для формы */
form {
    /* Максимальная ширина формы */
    max-width: 600px;
    /* Центрирует форму по горизонтали и добавляет отступ сверху */
    margin: 50px auto;
    /* Внутренний отступ вокруг содержимого формы */
    padding: 20px;
    /* Фон формы */
    background: #f3ffd6;
    /* Скругление углов формы */
    border-radius: 10px;
    /* Добавляет тень вокруг формы */
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    /* Использует Flexbox для управления расположением элементов внутри формы */
    display: flex;
    /* Указывает, что элементы должны быть расположены в колонку */
    flex-direction: column;
}

/* Стили для меток */

```

```

label {
  /* Отображает метки как блочные элементы */
  display: block;
  /* Добавляет отступ снизу для меток */
  margin-bottom: 5px;
  /* Выравнивает текст меток по левому краю */
  text-align: left;
  /* не позволяет тексту переноситься естественным образом на новую строку */
  white-space: nowrap;
}

/* Стили для полей ввода с разным типом*/
input[type="text"],
input[type="number"],
input[type="date"],
input[type="email"],
input[type="tel"] {
  /* Устанавливает ширину полей ввода на 97% от ширины родительского элемента */
  width: 97%;
  /* Добавляет внутренний отступ внутри полей ввода */
  padding: 10px;
  /* Добавляет отступ снизу для полей ввода */
  margin-bottom: 20px;
  /* Устанавливает границу для полей ввода */
  border: 1px solid #ddd;
  /* Скругление углов полей ввода */
  border-radius: 5px;
}

/* Стили для кнопки отправки */
input[type="submit"] {
  /* Градиентный фон для кнопки */
  background: linear-gradient(to right, #ff7e5f, #feb47b);
  /* Цвет текста кнопки */
  color: #fff;
  /* Убирает границу кнопки */
  border: none;
  /* Скругление углов кнопки */
  border-radius: 5px;
  /* Внутренний отступ внутри кнопки */
  padding: 10px 20px;
  /* Изменяет курсор при наведении на кнопку */
  cursor: pointer;
  /* Добавляет плавную анимацию при изменении фона */
  transition: background 0.3s ease;
  /* Устанавливает ширину кнопки на 100% от ширины родительского элемента */
  width: 100%;
  /* Устанавливает высоту кнопки */
  height: 50px;
}

```

```

    /* Устанавливает размер шрифта кнопки */
    font-size: 18px;
    /* Ограничивает текст кнопки, если он не помещается */
    text-overflow: ellipsis;
}

/* Стили для кнопки отправки при наведении */
input[type="submit"]:hover {
    /* Изменяет фон кнопки при наведении */
    background: linear-gradient(to right, #feb47b, #ff7e5f);
}

/* Стили для контейнера пароля */
.password {
    /* Ширина контейнера */
    width: 300px;
    /* Позиционирование контейнера */
    position: relative;
}

/* Стили для поля ввода пароля */
#password-input {
    /* Ширина поля ввода */
    width: 602px;
    /* Внутренние отступы поля ввода */
    padding: 5px 0;
    /* Высота поля ввода */
    height: 30px;
    /* Высота строки текста */
    line-height: 40px;
    /* Отступ первой строки текста */
    text-indent: 10px;
    /* Внешние отступы поля ввода */
    margin: 0 0 15px 0;
    /* Граница поля ввода */
    border: 1px solid #ddd;
    /* Скругление углов поля ввода */
    border-radius: 5px;
    /* Размер шрифта */
    font-size: 18px;
}

/* Стили для кнопки отображения пароля */
.password-control {
    /* Позиционирование кнопки отображения пароля */
    position: absolute;
    /* Отступ от нижнего края контейнера */
    bottom: 17%;
    /* Отступ от левого края контейнера */

```

```
left: 185%;
/* Отображение кнопки как блочного элемента */
display: inline-block;
/* Ширина кнопки */
width: 14%;
/* Высота кнопки */
height: 40%;
/* Фон кнопки отображения пароля */
background: url(../images/eye-pass.gif) 0 0 no-repeat;
}

/* Стили для кнопки скрытия пароля */
.password-control.view {
    /* Фон кнопки скрытия пароля */
    background: url(../images/eye-pass-hide.png) 0 0 no-repeat;
}
```

## script\_zakaz.js

```
// Получаем поле ввода номера телефона
let phoneInput = document.getElementById('contactPhone');

// Создаем маску для номера телефона
let phoneMask = IMask(phoneInput, {
  mask: '+{7}(000)000-00-00' // Задаем маску для номера телефона
});

// Функция для отображения/скрытия пароля
function show_hide_password(target){
  let input = document.getElementById('password-input');
  if (input.getAttribute('type') == 'password') {
    target.classList.add('view'); // Добавляем класс для изменения вида
    кнопки
    input.setAttribute('type', 'text'); // Изменяем тип поля ввода на текст,
    чтобы отобразить пароль
  } else {
    target.classList.remove('view'); // Удаляем класс для изменения вида
    кнопки
    input.setAttribute('type', 'password'); // Возвращаем тип поля ввода к
    паролю для скрытия его значения
  }
  return false; // Предотвращаем стандартное поведение ссылки
}

// Обработчик события отправки формы
document.getElementById('purchaseForm').addEventListener('submit',
function(event) {
  alert('Форма успешно отправлена!'); // Выводим сообщение об успешной отправке
  формы
});
```