|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кафедра |  |  |  |  |
|  |  | шифр |  | наименование кафедры, по которой выполняется работа |
| Дисциплина |  |  | | |
|  |  | наименование дисциплины | | |

|  |  |
| --- | --- |
| ИНДИВИДУАЛЬНЫЙ ИТОГОВЫЙ ОТЧЕТ | |
| Разработка веб-сайта для |
| «Веб-сайт по продаже электроники» |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ОБУЧАЮЩИЙСЯ** | | | | | | |
| группы | | | |  | | О741Б |
|  |  | Лунев А.В. | | | | |
| подпись |  | фамилия и инициалы | | | | |
|  | | | | |
| дата сдачи | | | | |
| **ПРОВЕРИЛ** | | | | | | |
| Ассистент кафедры О7 | | | | | | | |
| ученая степень, ученое звание, должность | | | | | | | |
|  |  | Добровольский В.С. | | | | | |
| подпись |  | фамилия и инициалы | | | | | |
| Оценка / балльная оценка | | |  | | | |
|  | | | | |
| дата проверки | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| г. Санкт-Петербург | | | | | | |
|  |  |  | 20 | 25 | г. |  |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc178681258)

[1 Постановка задачи 4](#_Toc178681259)

[2 Верстка страниц веб-сайта 5](#_Toc178681260)

[3 Реализация динамических форм 6](#_Toc178681261)

[4 Формирование структуры базы данных 7](#_Toc178681262)

[5 Процесс развертывания веб-сайта 8](#_Toc178681263)

[ЗАКЛЮЧЕНИЕ 9](#_Toc178681264)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10](#_Toc178681265)

# **ВВЕДЕНИЕ**

Современная веб-разработка играет ключевую роль в создании интерактивных, функциональных и удобных интернет-ресурсов. Сайты на сегодняшний день не ограничиваются простой подачей информации — они обеспечивают взаимодействие пользователей с контентом, обработку данных и персонализацию. Одним из популярных направлений является создание интернет-магазинов, где пользователь может самостоятельно выбирать товары, оформлять заказ и управлять своим аккаунтом.

Цель работы заключается в разработке веб-сайта L-U-N-E-V по продаже электроники.

Для достижения поставленной цели необходимо решить следующие задачи:

* описать функциональные требования к разрабатываемому веб-сайту;
* сверстать прототип веб-страниц;
* определить стили для разрабатываемого веб-сайта;
* реализовать динамические формы авторизации, безопасной аутентификации, вариационности товара.
* сформировать базу данных;
* развернуть разработанный веб-сайт на веб-сервере.

# **1 Постановка задачи**

В рамках данной работы необходимо спроектировать и разработать интернет-магазин электроники. Магазин будет называться «L-U-N-E-V». Основное назначение системы – это автоматизация процесса продажи товаров через веб-интерфейс, включая просмотр каталога, выбор техники и оформление заказа.

Необходимо создать работоспособное веб-приложение, реализующее полный цикл взаимодействия пользователя с магазином: от регистрации и поиска товара до добавления его в корзину или «Избранное».

Для достижения цели необходимо выполнить следующие этапы:

1. Разработать структуру веб-приложения по принципу «Клиент-Сервер», где фронтенд отвечает за визуализацию, а бэкенд за логику и работу с данными.
2. Спроектировать реляционную базу данных для хранения информации о пользователях, товарах, категориях и списках покупок. В качестве СУБД использовать PostgreSQL.
3. Написать API на платформе Node.js, которое будет обрабатывать HTTP-запросы, управлять сессиями пользователей и взаимодействовать с самой базой данных.
4. Создать набор веб-страниц (HTML/CSS) с адаптивной версткой и реализовать динамику интерфейса на JavaScript (регистрация, корзина, фильтрация товаров).
5. Реализовать механизм регистрации и авторизации пользователей с обязательным хешированием паролей.

В конце-концов необходимо будет получитьсайт, который будет корректно открываться как на ПК, так и на мобильных устройствах, каталог которого будет поддерживать различные характеристики товаров (память, цвет) и изображения. Также сайт должен обладать возможностью сохранять состояние корзины и избранного для зарегистрированных пользователей.

**2 Верстка страниц веб-сайта**

Целью данного этапа является создание визуальной части проекта (Frontend) – набора веб-страниц, которые обеспечивают интуитивно понятный, функциональный и эстетически привлекательный интерфейс. Обязательным требованием является адаптивность, позволяющая корректно отображать контент на устройствах с различным разрешением экрана (ПК, смартфоны).

Верстка реализована с использованием HTML5, CSS3 и JavaScript.

1. HTML5
   * Использована семантическая разметка (теги <header>, <main>, <footer>, <section>) для повышения доступности.
   * Разработаны ключевые макеты страниц: index.html (Главная), buy.html (Карточка товара), cart.html (Корзина), favorites.html (Избранное), а также формы login.html и register.html.
   * Общие элементы навигации и футера вынесены в отдельный модуль, который динамически подгружается и обновляется с помощью header.JS, позволяя менять ссылки в зависимости от статуса авторизации пользователя.
2. CSS3
   * Все стили централизованы в файле main.css. Использовались CSS-переменные (например, --accent, --bg) для упрощения дальнейшего изменения цветовой схемы проекта.
   * Стоит отметить градиентные заливки и эффекты при наведении, как показано в стилях для кнопки buy-button (файл buy.html).
3. JavaScript
   * Вся функциональная часть сайта обеспечена благодаря JavaScript: добавление товаров в избранное и корзину; регистрация и вход на сайт; корректирование количества товаров к корзине и в избранном.

Демонстрация работы Веб-сайта

При входе на сайт пользователя встречает главная страница интернет-сайта, на который представлена удобная навигация: логотип сайта(по нажатии, на который происходит переход на главную страницу) , кнопки: избранное, корзина, профиль.

Ниже шапки сайта представлено рекламное видео-окно с имеющимися на сайте товарами; пользователь способен переключать видеофайлы. Представлено на рисунке 1.

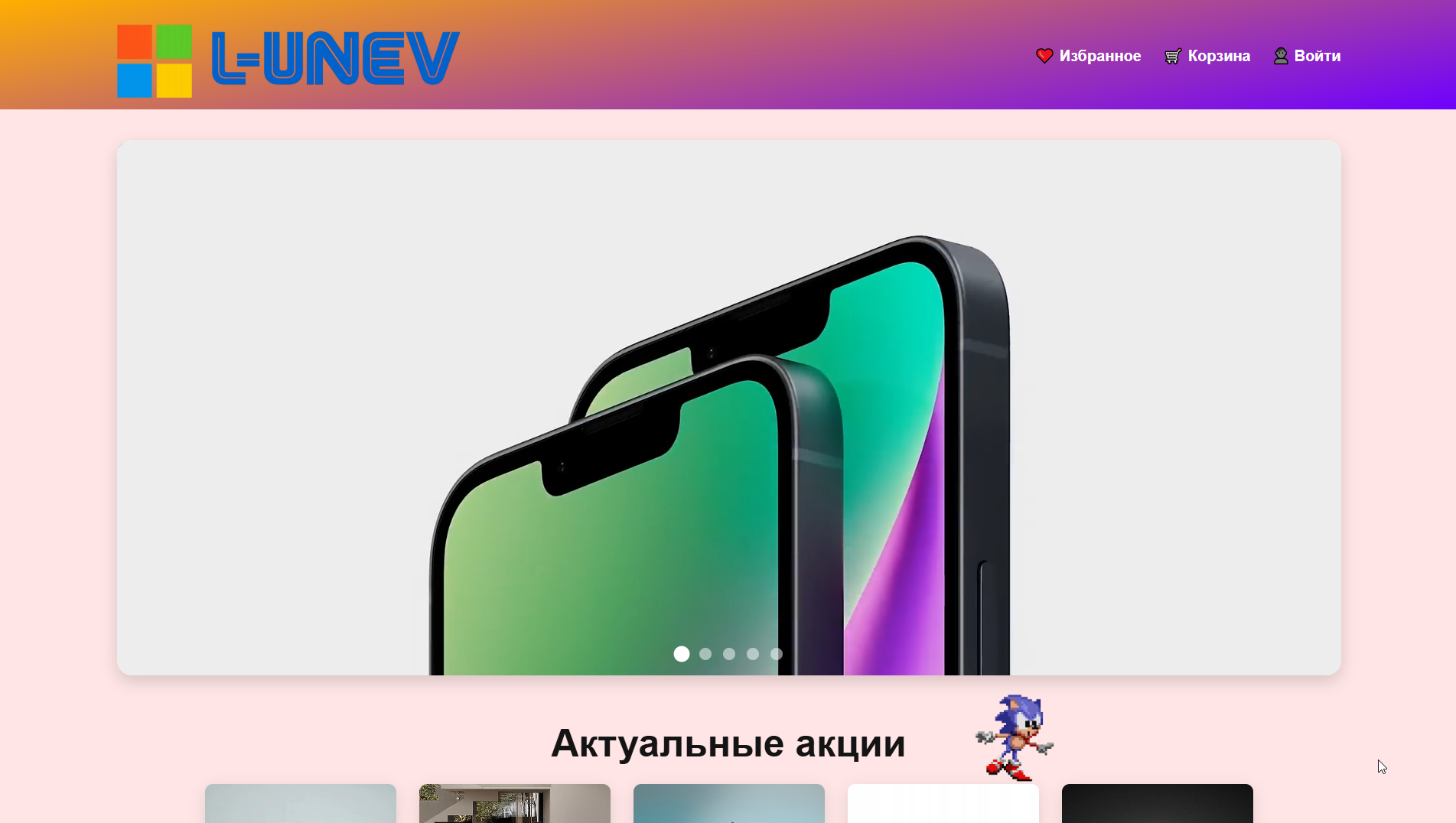


Рисунок 1 – Главная страница сайта

Ещё ниже представлено поле с актуальным акциями, в количестве 5 штук, с товарами и их описанием, которые пользователю могли бы понравиться.

Под этими карточками расположено 3 сторонних рекламных баннеры с кратким описанием продукта заказчика. Представлено на рисунке 2.

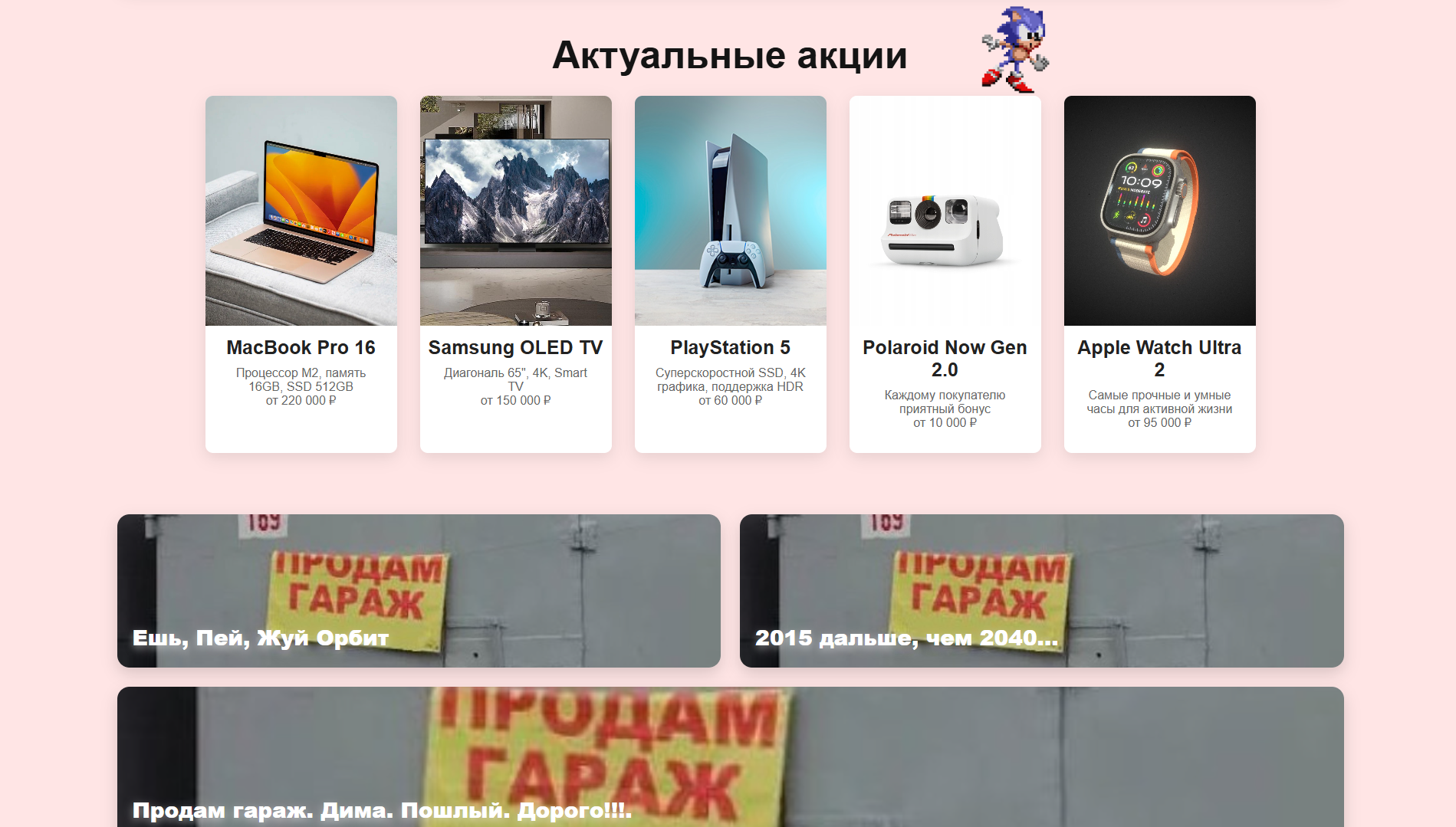


Рисунок 2 – Актуальные акции и баннеры

В самом низу страницы расположен футер, в котором располагается вся необходимая информация для обратной связи, вплоть до номера телефона и адреса электронной почты. Представлен слоган сайта и социальные сети, которые непосредственно связаны с магазином. Представлено на рисунке 3.

Стоит уточнить, что блок с контактами представлен на каждой странице сайта, тем самым создаётся ситуация с легкой обратной связью с создателями и модерацией.

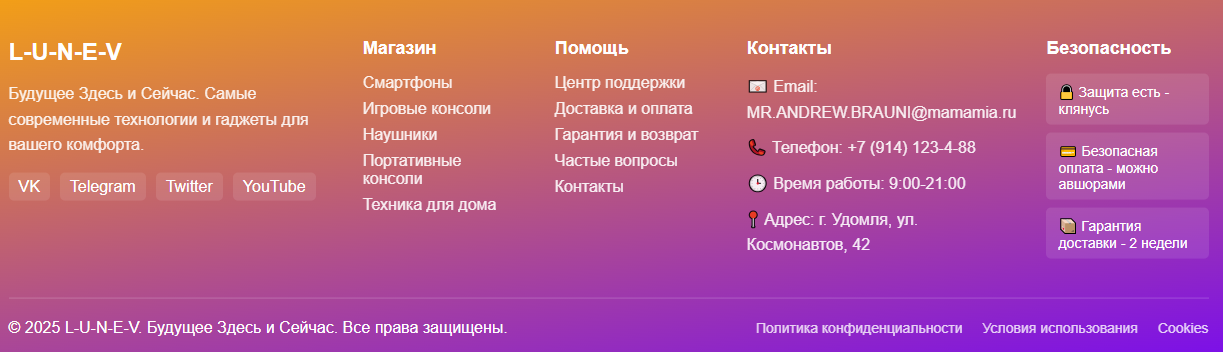


Рисунок 3 – футер

При нажатии на элемент в шапке под названием «Войти» появляется окно с вводом данных, в которое пользователь должен ввести свои данные для верификации своего аккаунта. Если же пользователь ещё не успел создать свой профиль, то необходимо пройти регистрацию для этого внизу окна расположена кнопка «Зарегистрироваться». Представлено на рисунке 4.

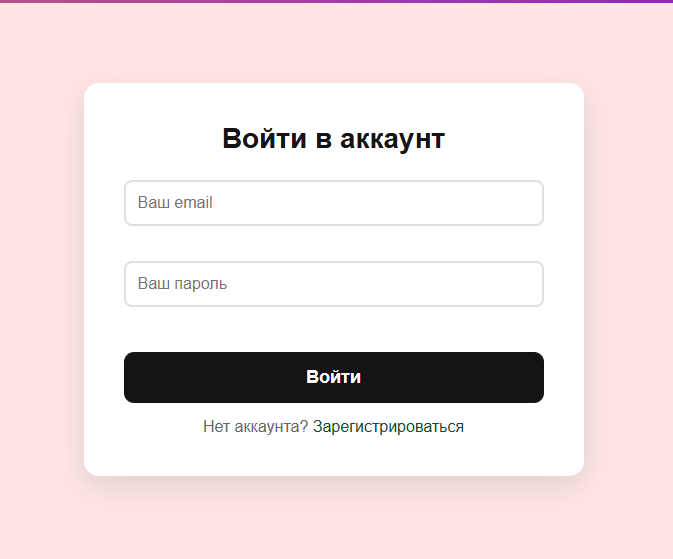


Рисунок 4 – Войти

При нажатии на соответствующую кнопку появляется окно регистрации. В которой необходимо заполнить поля: фамилия, имя, email, телефон, пароль, подтвердите пароль. После успешного ввода данных аккаунт успешно создастся и пользователя перекинет на главную страницу уже войденного в свой профиль. Представлено на рисунке 5.

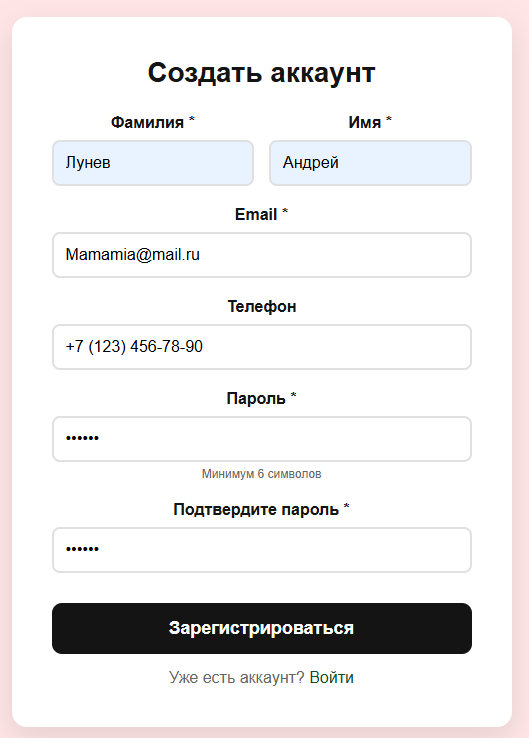


Рисунок 5 – Регистрация

После успешного процесса регистрации пользователю выдаст сообщение об успешном создании аккаунта. Представлено на рисунке 6.

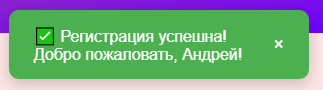


Рисунок 6 – Успешное создание аккаунта

При успешном входе в аккаунт пользователю выдаст соответствующее сообщение об успешном входе в аккаунт. Представлено на рисунке 7.

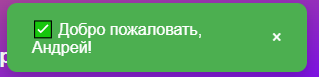


Рисунок 7 – Успешный вход в аккаунт

При создании профиля или входа в него можно посмотреть информацию, о себе, которую обыватель внёс ранее. Также с добавление товара в корзину/избранное происходит обновление счётчика и количество товаров повышается или понижается, в зависимости от выбора. Представлено на рисунке 8.

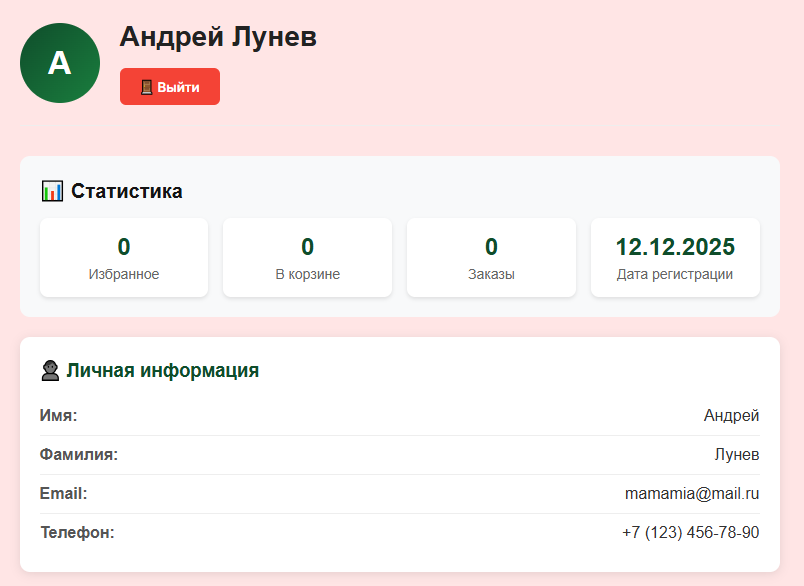


Рисунок 8 – Профиль

При нажатии на желанную категорию продукта, на главной странице, пользователя перекидывает на блок с товарами интересующих ему. Так после нажатия на видео-баннер с рекламой iPhone14 пользователя перекинет на страницу с телефонами, одним из которых и будет iPhone14. Представлено на рисунке 9.



Рисунок 9 – Телефоны

Так будет происходить с каждым слайдом видео-баннера. Представлено на рисунках 10, 11, 12, 13.

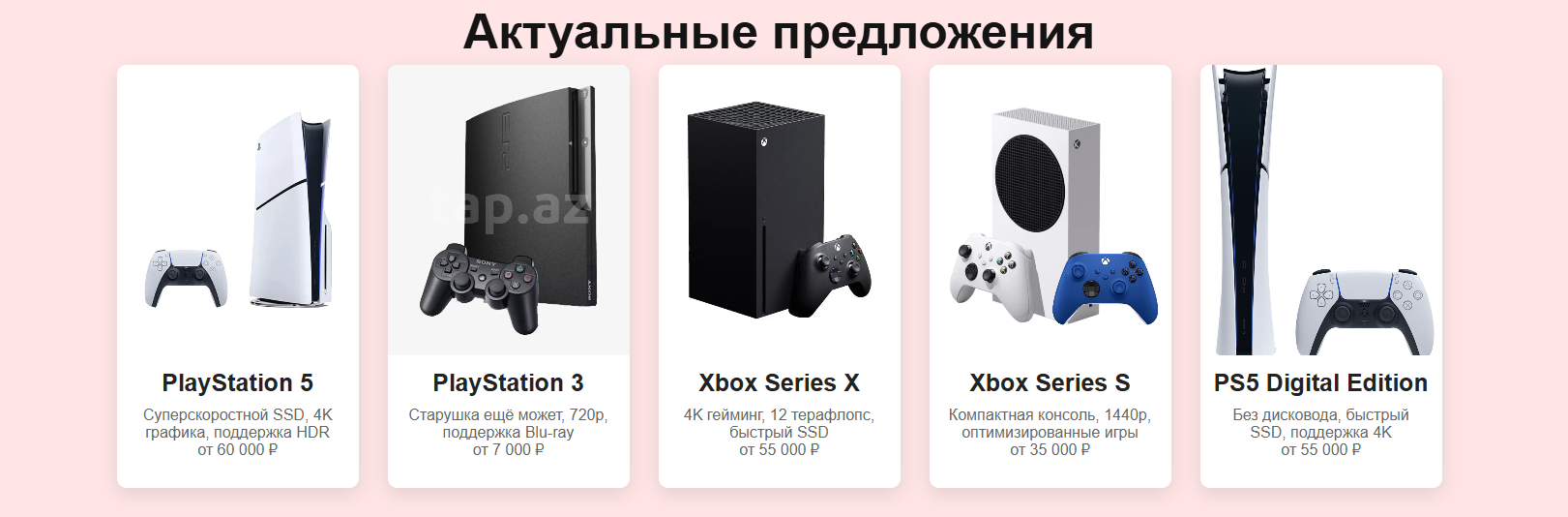


Рисунок 10 – Стационарные консоли



Рисунок 11 – Наушники

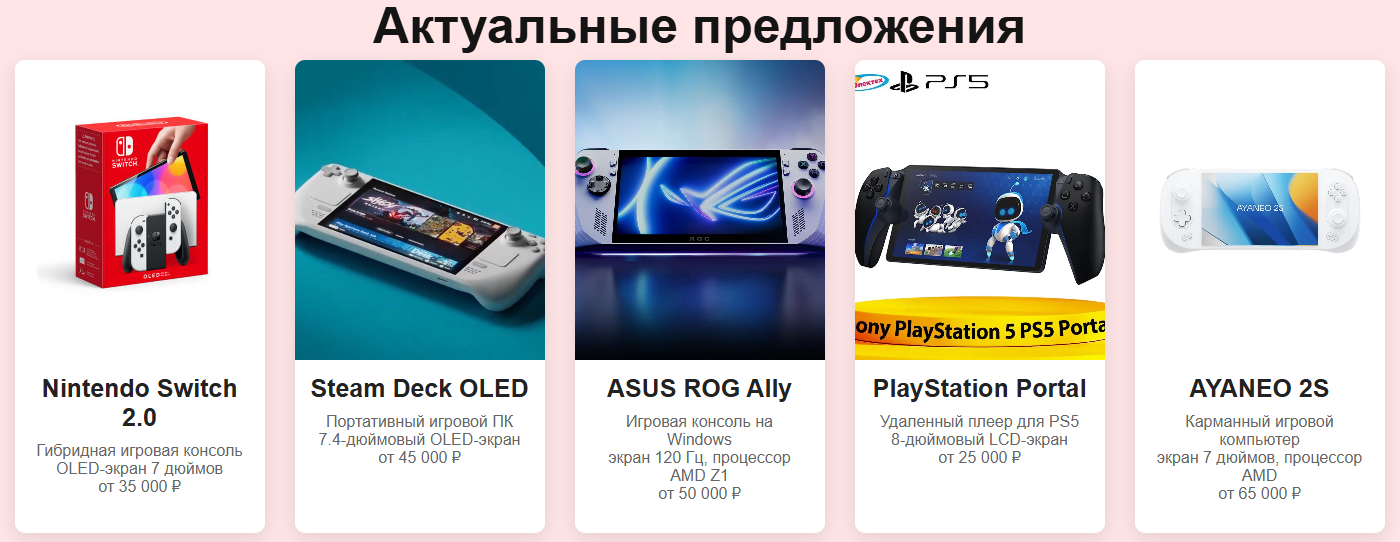


Рисунок 12 – Портативные консоли



Рисунок 13 – Стиральные машины

После нажатия на карточку с iPhone 14 Pro пользователя переносит на страницу покупки товара, где показаны изображения товара в нескольких разных положениях, предлагается возможность выбора количества встроенной памяти на устройстве, прилагается описание к товару и информация о условиях доставки/самовывоза. Пользователь волен добавить данный товар в корзину по нажатии на кнопку «ДОБАВИТЬ В КОРЗИНУ» или же добавить данный товар в избранное по нажатии на сердечко. Представлено на рисунке 14.



Рисунок 14 – Карточка товара 1

Примеры карточек с продуктами представлены на картинках 15,16,17.

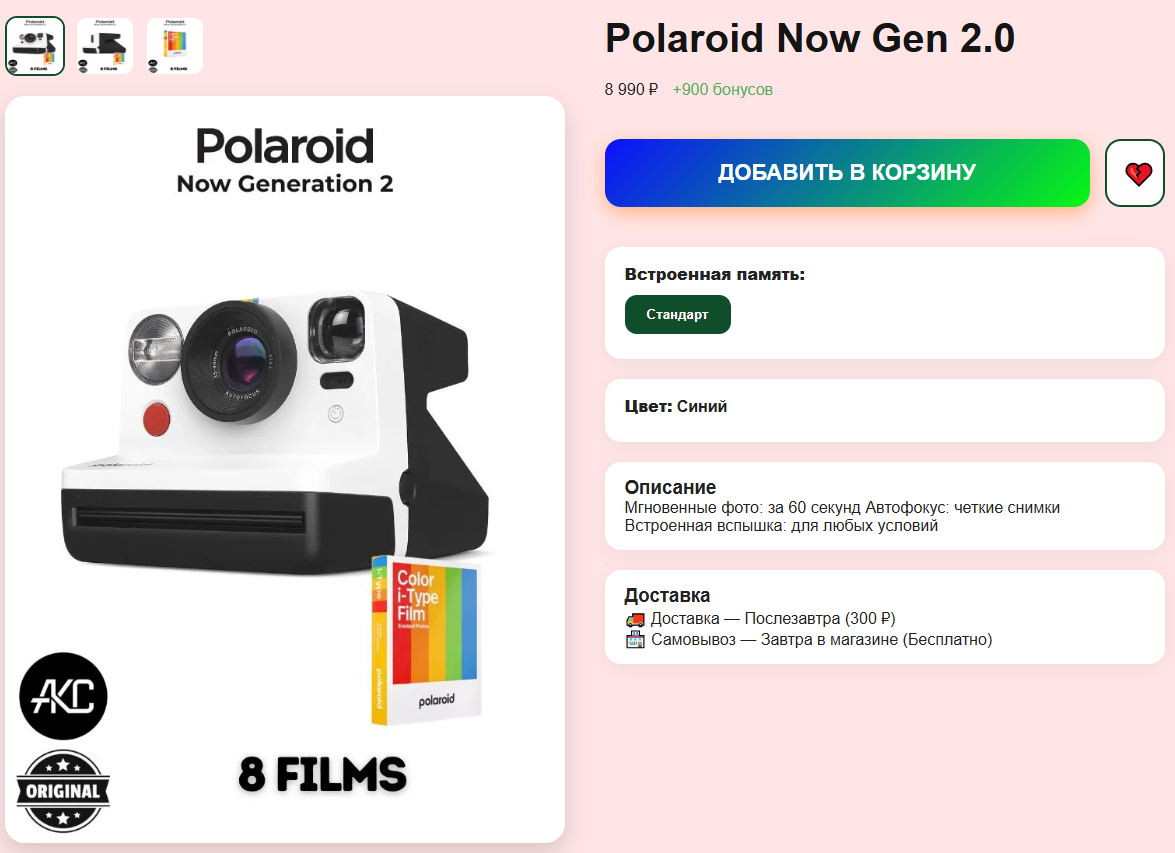


Рисунок 15 – Карточка товара 2

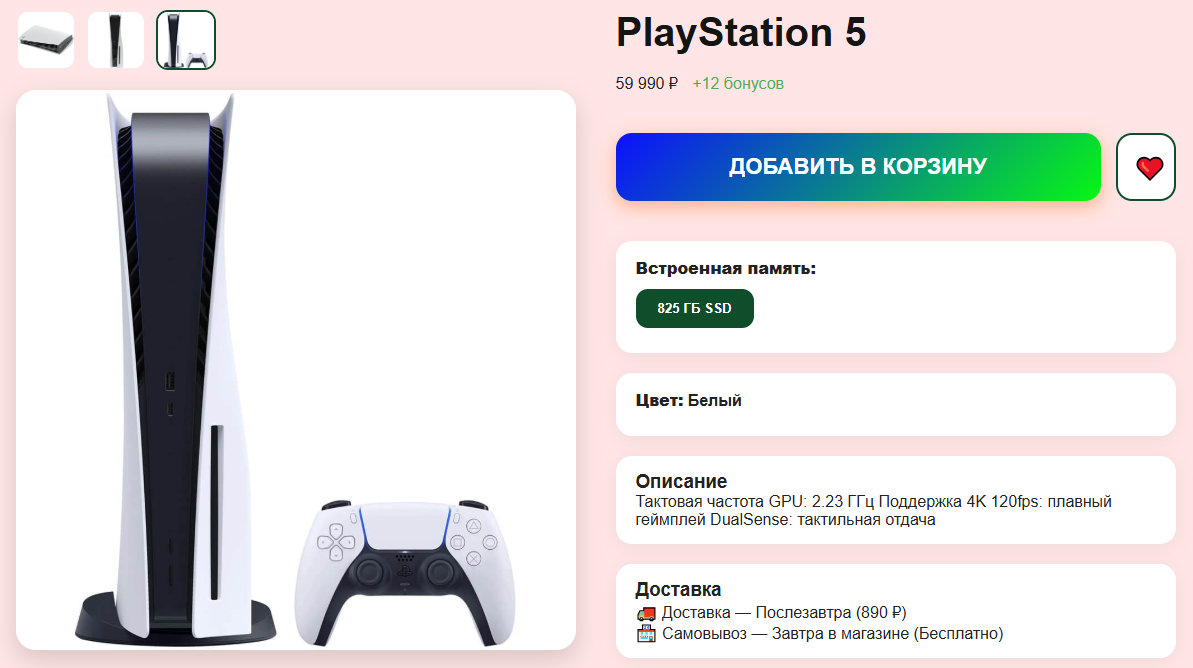


Рисунок 16 – Карточка товара 3

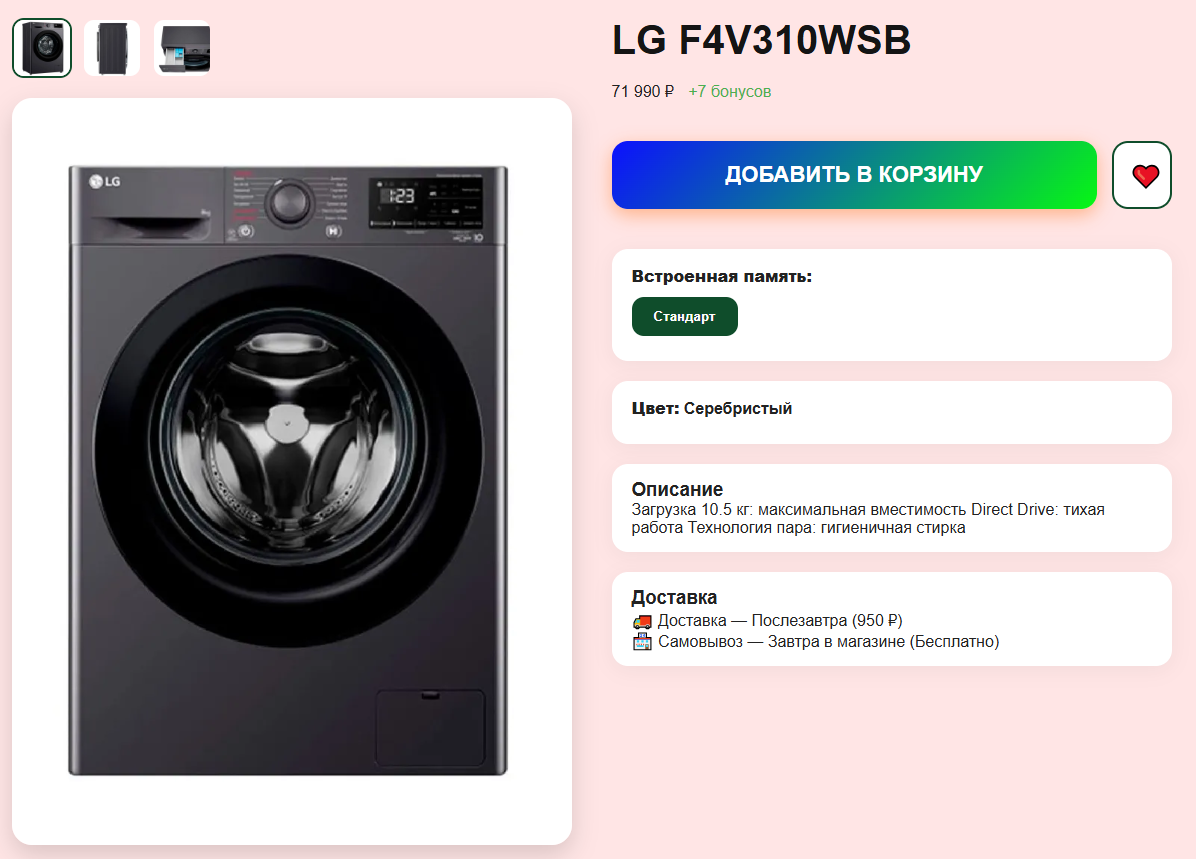


Рисунок 17 – Карточка товара 4

При добавлении в корзину/избранное товара соответствующие кнопки меняют своё отображение сообщая об успешном завершении процесса. Представлено на рисунке 18.

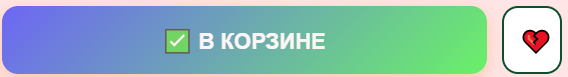


Рисунок 18 – Успешно добавление

Также добавлению товара в корзину сопутствует надпись об успешном добавлении в корзину объекта. Представлено на рисунке 19.

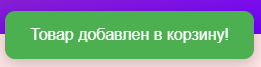


Рисунок 19 – Успешно добавлен 1

Также добавлению товара в избранное сопутствует надпись об успешном добавлении в избранное объекта. Представлено на рисунке 20.

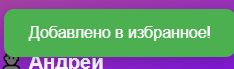


Рисунок 20 – Успешно добавлен 2

Корзина после добавления различных товаров в неё автоматически считает общую сумму товаров, находящихся в ней самой. Внешний вид корзины представлен на рисунке 21.

В корзине пользователь волен изменять количество определённого товара, удалять его по клику на урну.

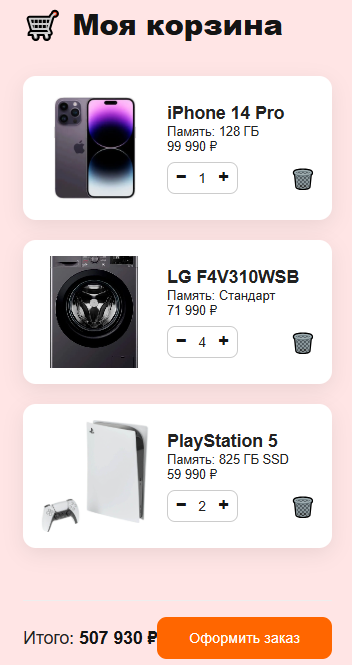


Рисунок 21 – Корзина

Избранное представлено в схожем с корзиной виде, пользователь может удалить товар, нажав на крестик в блоке товара, добавить в корзину желаемый ему товар, получить информацию о товаре, нажав на «Подробнее». Внешний вид избранного представлен на рисунке 22.

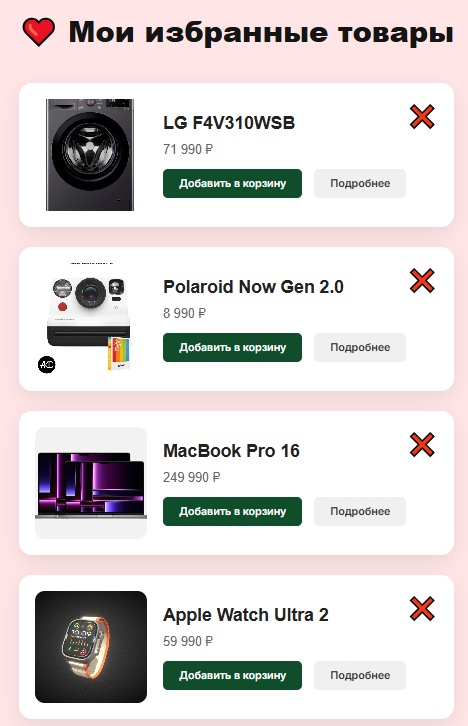


Рисунок 22 – Избранное

Была проведена адаптация сайта под телефоны, была проделана адаптивность дизайна всех страниц. Представлено на рисунках 23,24,25,26,27.

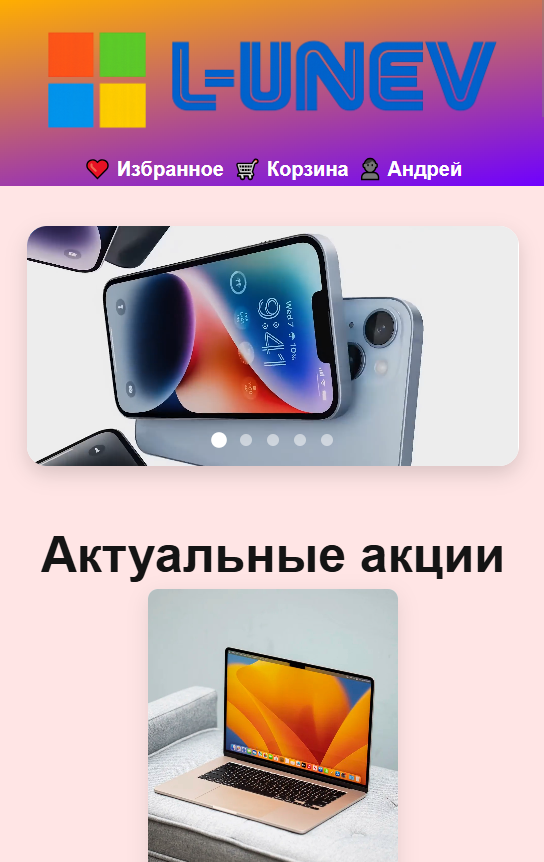


Рисунок 23 – Адаптированный экран под главную страницу

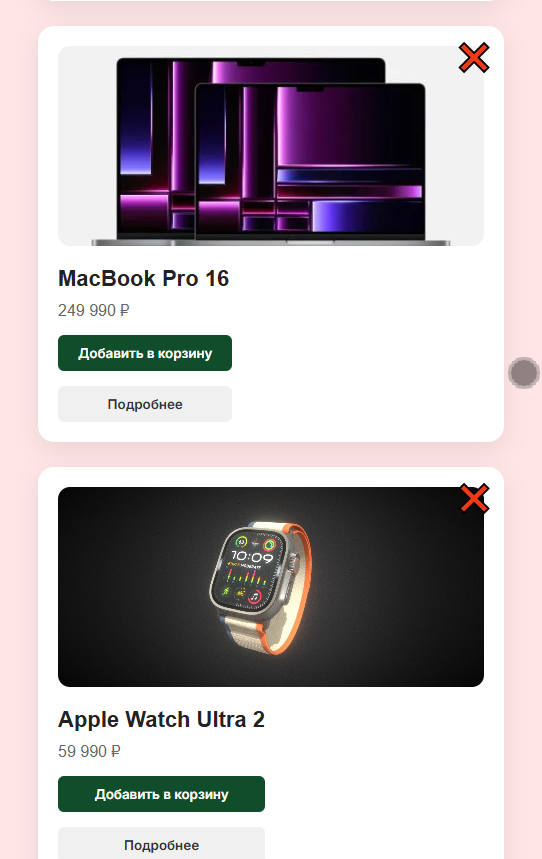


Рисунок 24 – Адаптированный экран под избранное

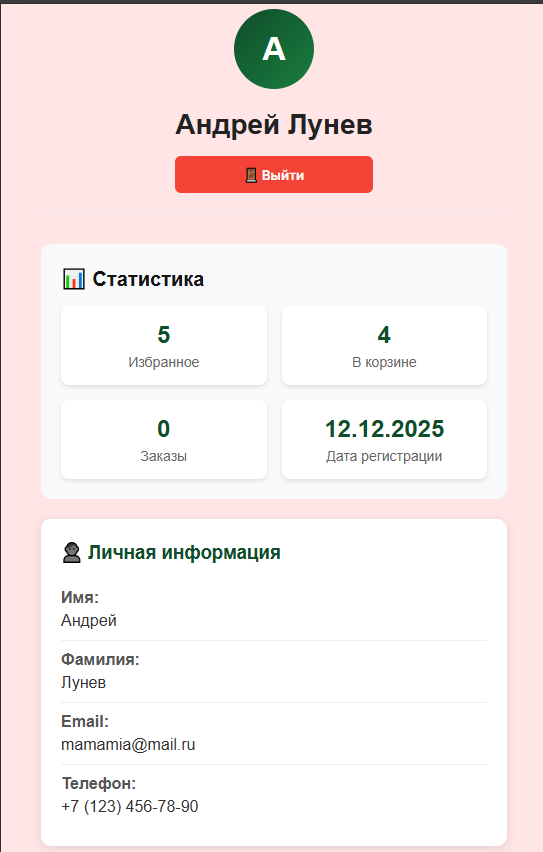


Рисунок 25 – Адаптированный экран под профиль

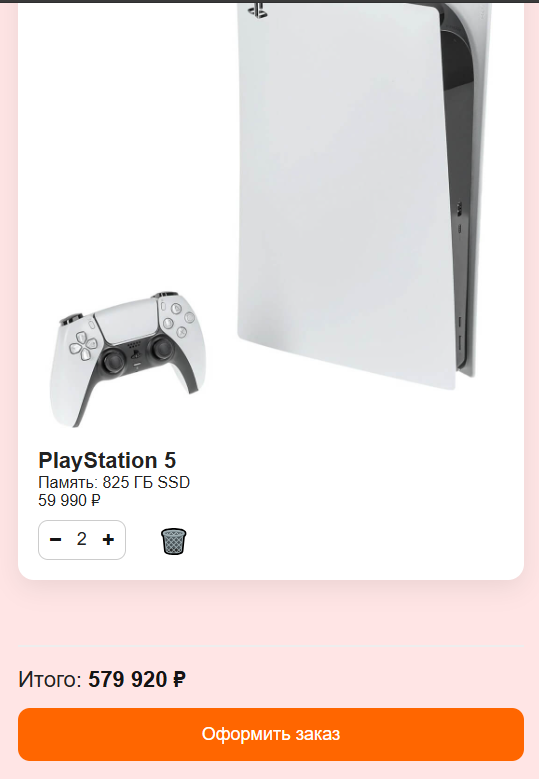


Рисунок 26 – Адаптированный экран под корзину

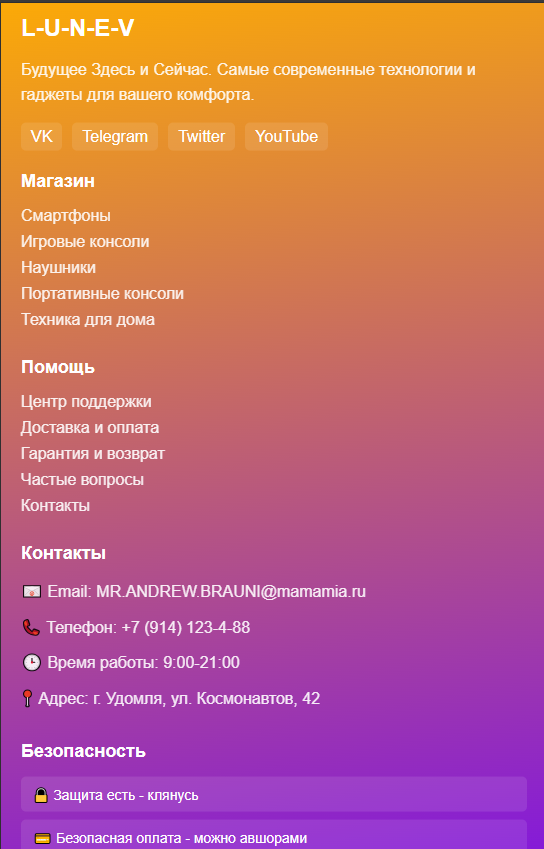


Рисунок 27 – Адаптированный экран под футер

**3 Реализация динамических форм**

Целью данного этапа является обеспечение интерактивности веб-сайта, реализация бизнес-логики на стороне клиента, а также управление клиентской сессией и локальным состоянием после успешной аутентификации на сервере.

Вся динамика интерфейса, включая обработку форм, валидацию и асинхронное взаимодействие, реализована на JavaScript.

##### 1. Аутентификация

Клиентская логика в файлах login.html и register.html не содержит резервного механизма авторизации. Процесс входа/регистрации всегда требует успешного ответа от серверного API, который, в свою очередь, обращается к базе данных PostgreSQL для проверки учетных данных (вход) или создания новой записи (регистрация).

* Серверный запрос: Скрипт собирает данные формы и отправляет их асинхронным запросом fetch на API (/api/login, /api/register).
* Роль localStorage: После того как сервер успешно проверил или создал пользователя в БД, localStorage используется для хранения данных сессии. Это позволяет JavaScript-модулю header.JS отображать имя пользователя и управлять навигацией на последующих страницах без повторного обращения к базе данных.

##### 2. Управление Корзиной

В отличие от аутентификации, функционал корзины использует:

* Гостевая корзина: Класс Cart использует localStorage для хранения списка товаров. Это позволяет гостям, которые еще не вошли в систему, собирать покупки.
* Серверная синхронизация (Авторизованный пользователь): При переходе к оформлению заказа или авторизации данные корзины могут быть синхронизированы с таблицей cart в PostgreSQL.

Демонстрация валидности

При вводе своих данных в окно регистрации при не полном вводе пользователя выдаст предупреждение, говорящее о надобности заполнения строки пользователем. Представлено на рисунках 28 - 36.

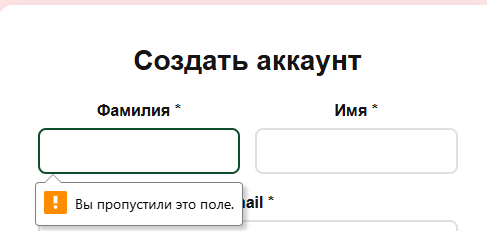


Рисунок 28 – Пропуск для ввода данных 1

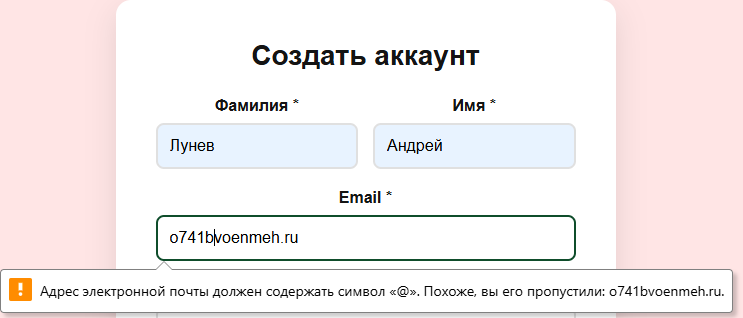


Рисунок 29 – Пропуск для ввода данных 2

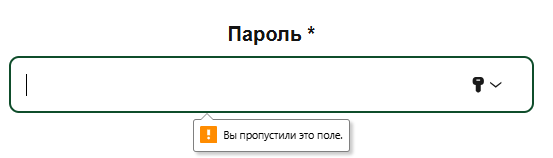


Рисунок 30 – Пропуск для ввода данных 3

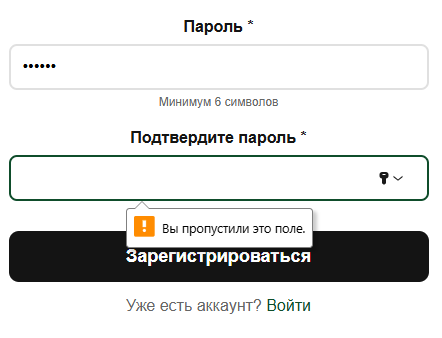


Рисунок 31 – Пропуск для ввода данных 3

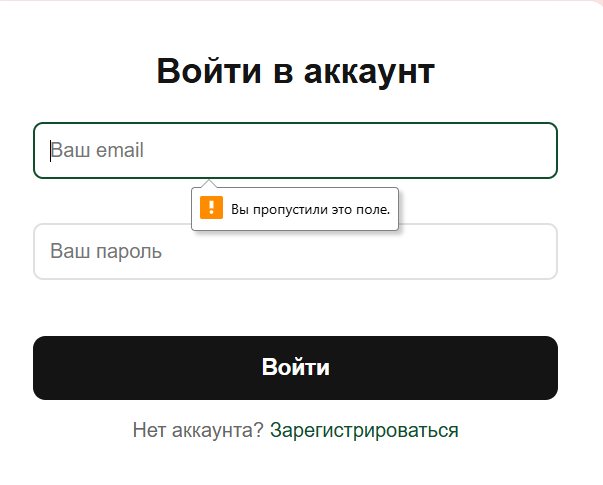


Рисунок 32 – Пропуск для ввода данных 4

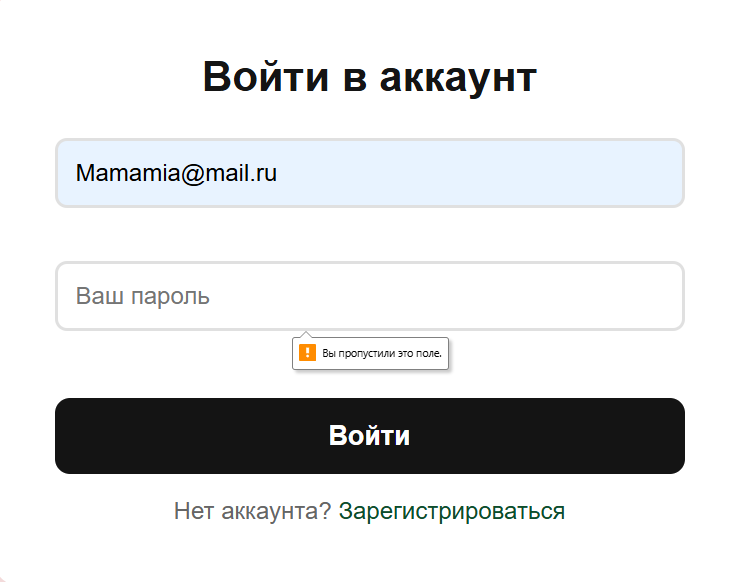


Рисунок 33 – Пропуск для ввода данных 5

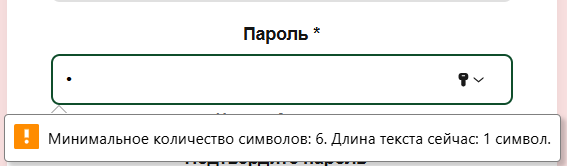


Рисунок 34 – Неудовлетворяющие данные 1

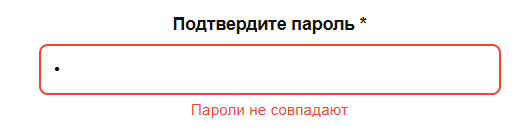


Рисунок 35 – Неудовлетворяющие данные 2

При неудачной попытке входа в аккаунт выдаёт оповещение о провальной попытке входа. Представлено на рисунке 36.

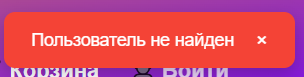


Рисунок 36 – Неверный логин или пароль

**4 Формирование структуры базы данных**

Целью данного этапа является проектирование и реализация надежной реляционной модели данных, которая служит для хранения всей информации о пользователях, товарах и транзакционной активности (корзина, избранное).

Выбор СУБД PostgreSQL обусловлен ее надежностью, поддержкой сложных типов данных и возможностью обеспечения высокой целостности данных. Представлено на рисунке 37.

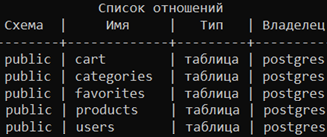


Рисунок 37 – Все таблицы БД

База данных PostgreSQL интернет-магазина shop\_lunev включает 5 основных таблиц, необходимых для полноценного функционирования:

* users: Хранит учетные записи пользователей (имя, email, хешированный пароль).
* products: Содержит полный каталог товаров, включая их цену и гибкие опции (например, объем памяти, цвет).
* categories: Обеспечивает классификацию товаров («Смартфоны», «Техника для дома»).
* cart: Хранит текущее содержимое корзины каждого пользователя (связь между пользователем, товаром и выбранной опцией).
* favorites: Хранит список товаров, добавленных пользователем в «Избранное».

Ключевые поля таблицы users:

* user\_id;
* email;
* first\_name;
* last\_name.

Главное назначение этой таблицы – это то, что учетные данные пользователей хранятся безопасно. Это видно по столбцу password\_hash\_prefix, который содержит не открытый пароль, а криптографический хеш. Наличие этого хеша доказывает, что реализован надежный механизм защиты паролей от несанкционированного доступа. Представлено на рисунке 38.

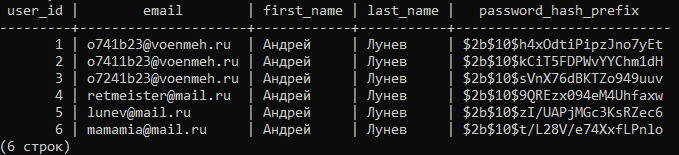


Рисунок 38 – Таблица users

Таблица products демонстрирует наполнение каталога товаров.

Главная особенность реализация хранения данных для характеристик товаров. Столбец memory\_options содержит несколько вариантов (128 ГБ,256 ГБ...). Это позволяет хранить список доступных опций (объем памяти, цвета) для каждого товара в одной записи, что упрощает управление каталогом. Представлено на рисунке 39.

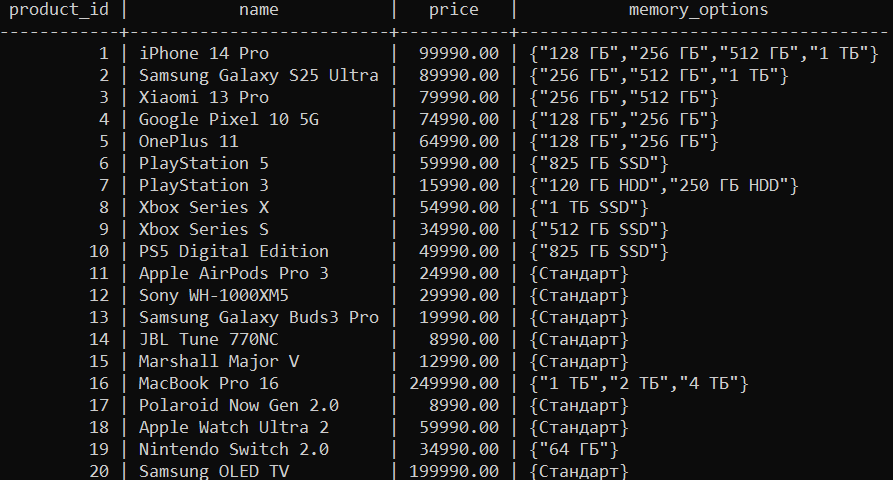


Рисунок 39 – Таблица products

Между пользователями (user\_id) и товарами (product\_id). Каждая запись в этой таблице (favorite\_id) связывает конкретного пользователя с товаром, который он добавил в избранное. Представлено на рисунке 40.

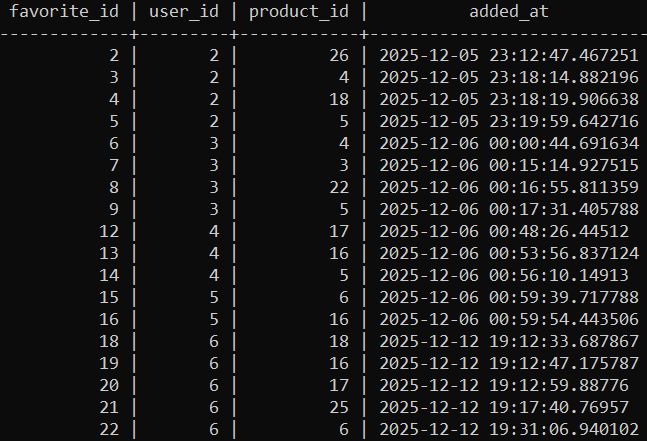


Рисунок 40 – Таблица favourite

Столбец selected\_memory показывает, что система корректно учитывает выбранную пользователем опцию (128 ГБ, 1 ТБ). Благодаря уникальному ключу по user\_id, product\_id и selected\_memory (см. строки с user\_id = 2), в корзине может быть один и тот же товар, но с разными опциями (например, product\_id = 4 с опцией 128 ГБ и без опции). Столбец quantity демонстрирует, что пользователи могут добавлять несколько единиц одного и того же товара (product\_id = 5 в количестве 3, product\_id = 16 в количестве 5, product\_id = 6 в количестве 2). Представлено на рисунке 41.

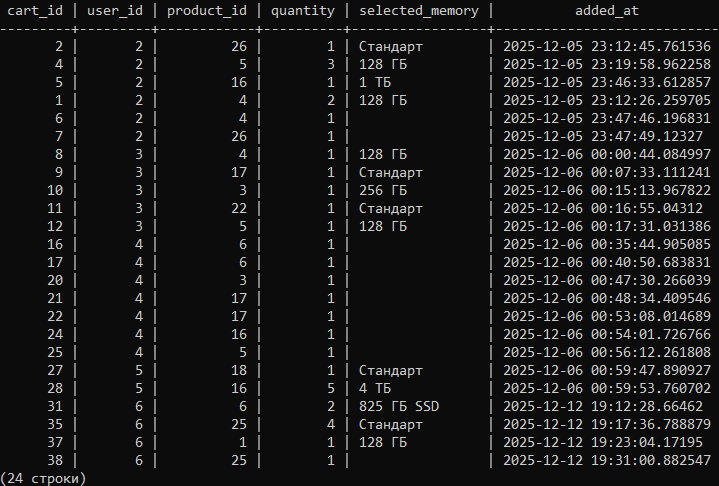


Рисунок 41 – таблица cart

Таблица содержит список всех категорий товаров, таких как «Смартфоны», «Игровые консоли» и «Ноутбуки». Столбец slug демонстрирует, что для каждой категории создано системное имя, которое будет использоваться для формирования чистых URL-адресов на сайте. Представлено на рисунке 42.

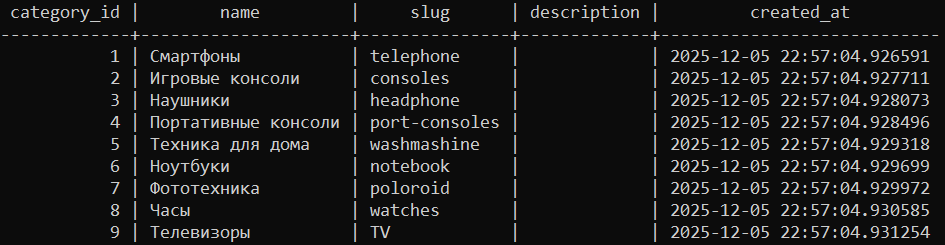


Рисунок 42 – Таблица categories

В соответствии с требованиями к функционалу интернет-магазина, в базе данных shop\_lunev были созданы пять ключевых таблиц. Каждая таблица имеет первичный ключ (PRIMARY KEY) и использует внешние ключи (FOREIGN KEY) для обеспечения реляционной целостности.

**5 Процесс развертывания веб-сайта**

Цель

Описать выбранный веб-хостинг

Описать процесс развертывания веб-сайта

Прикрепить доменное имя и скриншот из веб-браузера с демонстрацией доменного имени в поисковой строке

**ЗАКЛЮЧЕНИЕ**

Описать выводы по проделанной работе

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Не менее 5 источников (оформить в соответствии с ГОСТ 7.32-2017 см. курс в Moodle).