|  |  |
| --- | --- |
| RTK_LOGO.jpg | ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА  Комитет по образованию  Санкт-Петербургское государственное бюджетное  профессиональное образовательное учреждение  «Радиотехнический колледж» |

|  |
| --- |
| К защите допущен  заместитель директора по УПР  \_\_\_\_\_\_\_\_\_\_Т.А. Бессчетнова  «\_\_\_\_» \_\_\_\_\_\_\_ 2023 г. |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема: «Автоматизированная система учёта баз практик»

|  |
| --- |
| Специальность 09.02.07 Информационные системы и программирование  Выполнил: обучающийся группы № (подпись)  Руководитель: преподаватель (подпись) |

Санкт-Петербург 2023 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc128739985)

[1. Системный анализ предметной области. 4](#_Toc128739986)

[3. Описание этапов проектирования системы. 6](#_Toc128739987)

[3.1. Проектирование структуры ИнтС. 6](#_Toc128739988)

[3.2. Проектирование базы данных. 9](#_Toc128739989)

[4. Выбор средств реализации проекта. 11](#_Toc128739990)

[5. Описание этапов реализации ИнтС. 13](#_Toc128739991)

[5.1. Даталогическое проектирование баз данных. 13](#_Toc128739992)

[5.2. Поэтапное описание реализации компонентов ИтнС. 13](#_Toc128739993)

[ЗАКЛЮЧЕНИЕ 25](#_Toc128739994)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 26](#_Toc128739995)

[Приложение А (обязательное) Программные коды 27](#_Toc128739996)

## ВВЕДЕНИЕ

Темой дипломной работы является проектирование и разработка информационной системы (ИС) на базе Радиотехнического колледжа (РТК), которая позволяет записывать, хранить и искать производственную практику. Информационная система позволит упростить процесс поиска мест производственной практики для студентов, работодатели смогут приглашать к себе студентов и отмечать их уровень работы, а также облегчит работу колледжа и предприятий, которые предлагают свои места.

Предполагается, что студенты, перед прохождением производственной практики будут заходить на платформу и подбирать для себя нужное предприятие. А предприятия и компании, которые заинтересованы в привлечении молодых специалистов-студентов, будут размещать свои вакансии.

Производственная практика является важным периодом в ходе обучения будущих специалистов. За время прохождения производственной практики студенты учатся применять и отрабатывать свои знания и навыки, полученные во время обучения, в условиях реальной жизни. Практическая часть учебного процесса является заключительной частью учебной практики, проходящей в учебном заведении. Производственная практика, как правило, проходит на различных предприятиях в условиях реального производства.

**Актуальность**

Актуальность выбранной темы заключается в проблеме поиска места прохождения производственной практики. Зачастую в учебных заведениях возникает проблема устройства студентов на прохождение практики.

Также в связи с невероятно быстрым развитием технологий большая часть бумажной работы начала вытесняться более удобными и менее трудоемкими способами решения тех, или иных проблем, связанных с бумажной волокитой, будь это заполнение документов, написание текстов, или, как это продемонстрировано в данной работе, – формирование списков на практики.

**Цель работы**

Целью выпускной квалификационной работы является разработка автоматизированной системы учёта баз практик «Practice».

**Задачи**

* провести анализ предметной области, конкурентов и целевой аудитории;
* сформулировать требования к веб-сайту;
* обосновать выбор технических и программных средств для разработки веб-сайта;
* разработать дизайн веб-сайта;
* спроектировать и разработать веб-сайт;
* осуществить тестирование разработанной информационной системы.

**Структура дипломного проекта**

Структура дипломного проекта обусловлена актуальностью, предметом и задачами исследования. Работа включает введение, три главы, заключение, список использованных источников и приложений.

Во введении раскрыта актуальность выбранной темы, заданы предмет и объект исследования, также установлены цели и задачи, выявлена теоретическая и практическая значимость.

Первая глава посвящена определению предметной области и анализу теоретической информации по заявленной теме исследования. В первом параграфе рассматривается предметная область проекта, понятия информационной системы и производственной практики. Во втором параграфе представлены аналоги проекта, а также их описание и характеристика.

1. Системный анализ предметной области.

Интернет-магазин часов – это веб-сайт, позволяющий клиенту приобрести интересующие его товары. Веб-сайт должен предоставлять пользователю информацию о продаваемых товарах, о категориях товаров, о правилах покупки, контактную информацию интернет-магазина. Главная задача веб-сайта – осуществление продажи товаров клиентам. Пользователь, желающий приобрести тот или иной товар, добавляет его в корзину товаров, оплачивает заказ безналичным способом с помощью электронных терминалов. Данный интернет-магазин ориентирован на продажу фирменных, как электронных, так и механических часов.

Аудитория: молодежь и люди среднего возраста, преимущественно мужского пола.

В магазине будут представлены следующие категории для продажи товаров:

* женские часы,
* мужские часы,
* детские часы.

Предполагается расширение ассортимента за счет эксклюзивных товаров, не представленных в магазинах-конкурентах.

В описании товаров должны присутствовать следующие свойства: название; цена; страна производителя; количество, год выпуска, модель товара и категория товара.

Прежде чем совершить покупку через веб-сайт, пользователь должен пройти регистрацию и зайти на сайт со своего созданного профиля. Пользователь может оформить заказ понравившегося товара, нажав на соответствующую кнопку на странице данного товара и добавив его в корзину.

На странице с товаром пользователь может ознакомиться с характеристиками товара, количеством товара на складе. В случае если для оформления заказа товаров не хватает, пользователь получает уведомление о невозможности оформления заказа в данный момент.

После выбора всех понравившихся товаров в корзину, пользователь может перейти к оформлению заказа. Регистрация сделки производится при помощи электронной платежной системы.

По окончанию оформления заказа, пользователь получает уведомление, на указанный ранее, электронный адрес.

**2. Постановка задачи.**

Можно выделить несколько задач работы:

* проектирование структуры проекта,
* проектирование базы данных,
* выбор средств реализации проекта,
* требования к техническому обеспечению,
* реализация системы,
* создание баз данных,
* реализация компонентов системы.

Информационная система интернет-магазина часов должна обеспечивать:

* возможность удобного структурированного хранения информации в объеме существующих документов;
* возможность удобного и быстрого ввода и занесения необходимой информации;
* возможность быстрого вывода и архивирования информации;
* возможность сетевого доступа к информационной системе всех заинтересованных в получении информации пользователей;
* ограничение доступа к имеющейся в информационной системе конфиденциальной информации через разграничение прав доступа пользователей (требование законодательства о защите персональных данных);
* отдельное хранение нормативных документов с возможностью быстрого обращения к ним и последующего открытия;
* быстрое нахождение нужной информации о приобретаемом товаре.

3. Описание этапов проектирования системы.

### 3.1. Проектирование структуры ИнтС.

В системе должна быть реализована регистрация и авторизация пользователя. Эти две подсистемы служат для обеспечения парольной защиты и разграничения прав доступа в систему. База данных хранит информацию о логинах и паролях пользователей. Подсистемы обеспечивают возможность ввода данных пользователя, проверку введенных данных, информирование о неверном вводе пароля, блокировку входа в систему при неправильном вводе данных, возможность изменения, добавления и удаления данных пользователя.

Структура системы включает в себя ссылки из навигационной панели (рисунок 1).

Структура

Где нас найти?

О нас

Каталог

Регистрация

Авторизация

Рисунок 1 – Структура системы

Все пользователи системы подразделяются на три группы:

* 1. Администраторы,
  2. Посетитель сайта,
  3. Клиент.

Функционал администратора:

* добавление, удаление, редактирование данных о товаре,
* добавление, удаление категорий товаров,
* просмотр всех заказов и смена их статусов.

Функционал посетителя сайта:

* просмотр товаров и рубрик сайта.

Функционал клиента:

* добавление товаров в корзину,
* редактирование корзины,
* создание заказа,
* просмотр своих заказов.

Проектируемая мной система должна включать в себя такой основной функционал, как оформление заказов, доступный зарегистрированному пользователю. Заказы можно просматривать, редактировать и удалять. Таким образом, пользователь может просматривать свои заказы в личном кабинете.

Подсистема администратора состоит из трёх модулей: работа с товарами, работа с категориями, работа с заказами. Вход в подсистему происходит при авторизации пользователя с логином «Admin». Подсистема служит для организации удобного взаимодействия пользователя с правами администратора с соответствующими каталогами и подсистемой авторизации.

Модуль работы с товарами позволяет добавлять, редактировать, удалять и открывать для просмотра товары.

Модуль работы с категориями позволяет добавлять, удалять и редактировать категории.

Модуль работы с заказами позволяет отменять и подтверждать новый заказ и открывать для просмотра товары.

**3.1.1 Рубрики сайта.**

Всем пользователям доступны рубрики: "О нас", "Каталог", "Где нас найти?". Также в меню нужно сделать кнопку вход и регистрация.

На странице «О нас» представлен слайдер ("Новинки компании") с пятью последними добавленными товарами (выводится фото товара и его название).

На странице «Каталог» по умолчанию представлены все товары, упорядоченные по новизне (от новых добавленных к более старым). Упорядочить товары можно: по году производства, по наименованию, по цене. Также товары можно отфильтровать по категориям. Каждая карточка товара представлена изображением, названием и ценой. При клике на товар открывается отдельная страница товара. Пользователю всегда показывает товары, которые только есть в наличии.

Каждая страница товаров содержит фото товара, наименование, цену и характеристики (страна-производитель, год выпуска, модель).

Форма регистрации нового пользователя в системе должна содержать следующие

поля:

* name – обязательное поле, разрешенные символы (кириллица, пробел и тире);
* surname – обязательное поле, разрешенные символы (кириллица, пробел и

тире);

* patronymic – не обязательное поле, разрешенные символы (кириллица, пробел
* и тире);
* login – обязательное и уникальное поле, разрешенные символы (латиница,

цифры и тире);

* email – обязательное и уникальное поле, проверка на email;
* password – обязательное поле, не менее 6-ти символов;
* password\_repeat – обязательное поле, должно совпадать с полем password;
* rules - согласие с правилами регистрации.

При ошибках валидации пользователю должны выводиться сообщения без

перезагрузки страницы.

Форма авторизации пользователя в системе должна содержать следующие поля:

* login,
* password.

При ошибках валидации пользователю должны выводиться сообщения.

Чтобы наглядно продемонстрировать интерфейс приложения, были разработаны несколько вариантов будущей системы - последовательность окон (форм), снабженных управляющими элементами.

Форма «Авторизация» содержит выпадающий список логинов, загружающийся при открытии формы, текстовое поле для ввода пароля, флажок для открытия/скрытия вводимых символов и кнопку «Пройти авторизацию». Внешний вид формы представлен на рисунке 2.

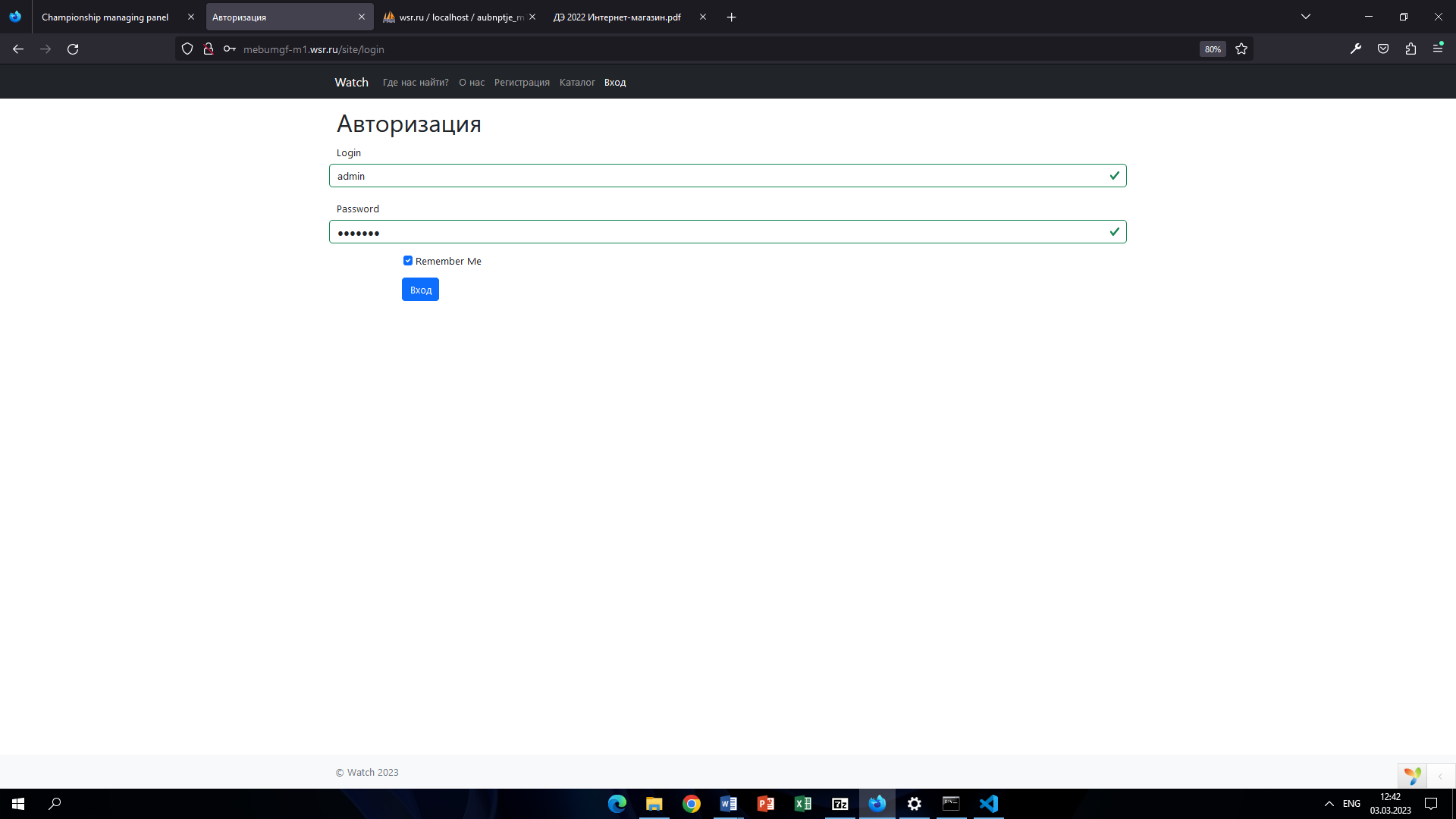


Рисунок 2 – Форма «Авторизация»

Форма «Регистрация» обеспечивает возможность регистрации нового пользователя системы. В форме присутствует чек-бокс «Я студент», позволяющий разграничить права куратора и студента (рисунок 3).

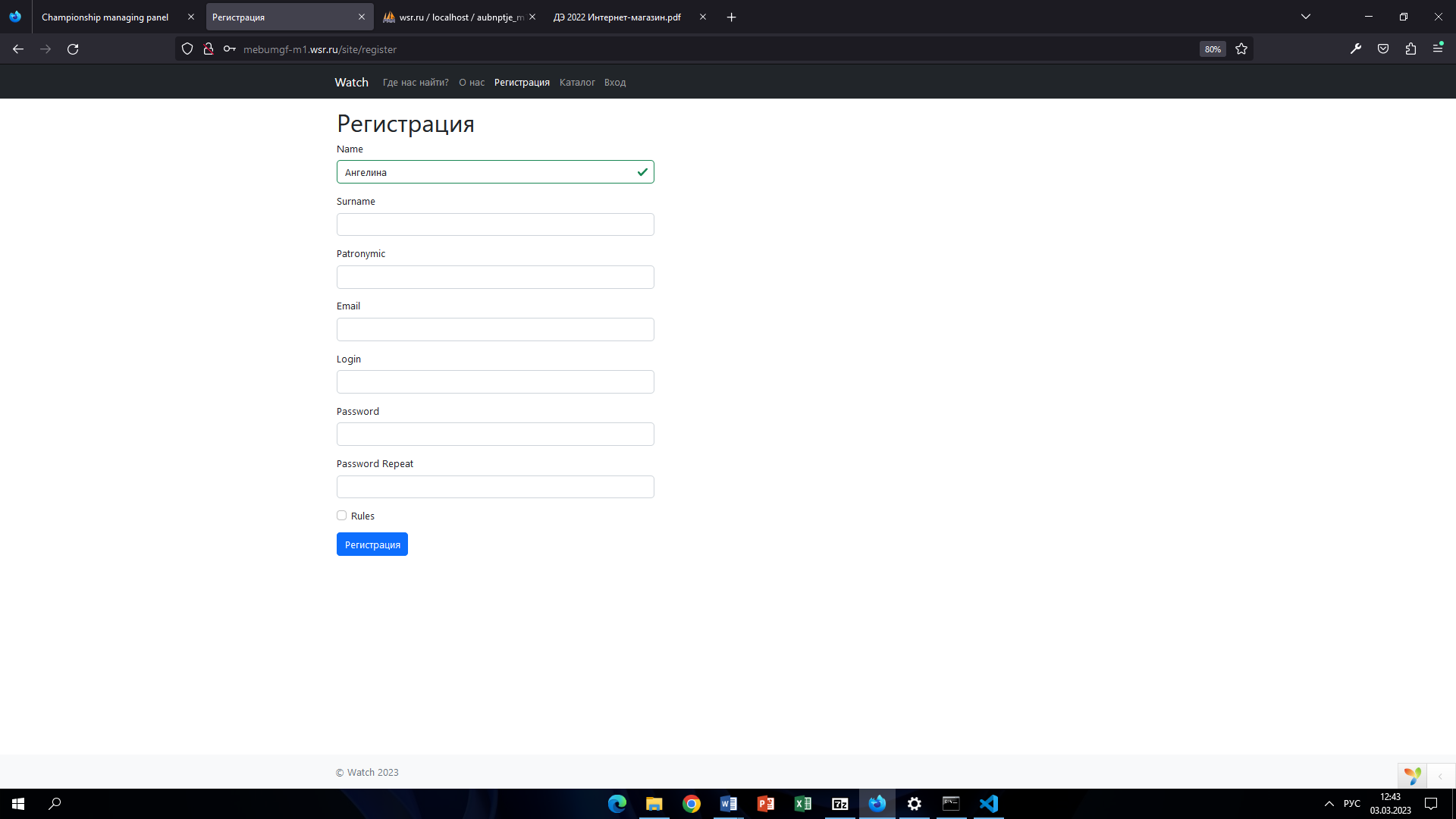


Рисунок 3 – Форма «Регистрация»

### 3.2. Проектирование базы данных.

База данных - совокупность связанных данных, сохраняемая в двумерных таблицах информационной системы. Программное обеспечение информационной системы, обеспечивающей создание, ведение и совместное использование баз данных, называется системой управления базами данных (СУБД). База данных представляет собой удачно организованный набор поименованных таблиц. Каждая таблица - массив из однородных элементов, которые называются записями. Запись может содержать в себе одно или несколько именованных полей. Число и имена полей задаются при создании таблицы. Каждое поле имеет определенный тип. Для работы с компьютерной системой я выбрала одну из самых популярных СУБД - MySQL, потому что она проста и универсальна для большинства приложений. Для работы с ней используется язык структурированных запросов SQL (Structure Query Language), который позволяет выполнять различные операции с базами данных: создавать таблицы, помещать, обновлять и удалять из них данные, производить запросы из таблиц и т.д.

Выполним проектирование базы данных системы, необходимое для структурирования элементов и составных частей при ее разработке. На этапе проектирования системы было распределено несколько ролей и выделены модули системы. Вследствие этого, база данных будет включать в себя упомянутые выше сущности, их связи и атрибуты.

Для наглядности, в сервисе [Draw io](https://app.diagrams.net/) мной была создана диаграмма «Сущность-связь» спроектированной базы данных, представленная на рисунке 4.

Графически сущности на диаграммах “сущность-связь” представляются в виде прямоугольников, изображение атрибутов сущности представляются в виде овальных блоков, а для графического представления связи используется ромб. Показатель кардинальности связи (характеристика однозначности) обозначает степень взаимосвязи сущностей и описывает количество возможных связей для каждой из сущностей-участниц.

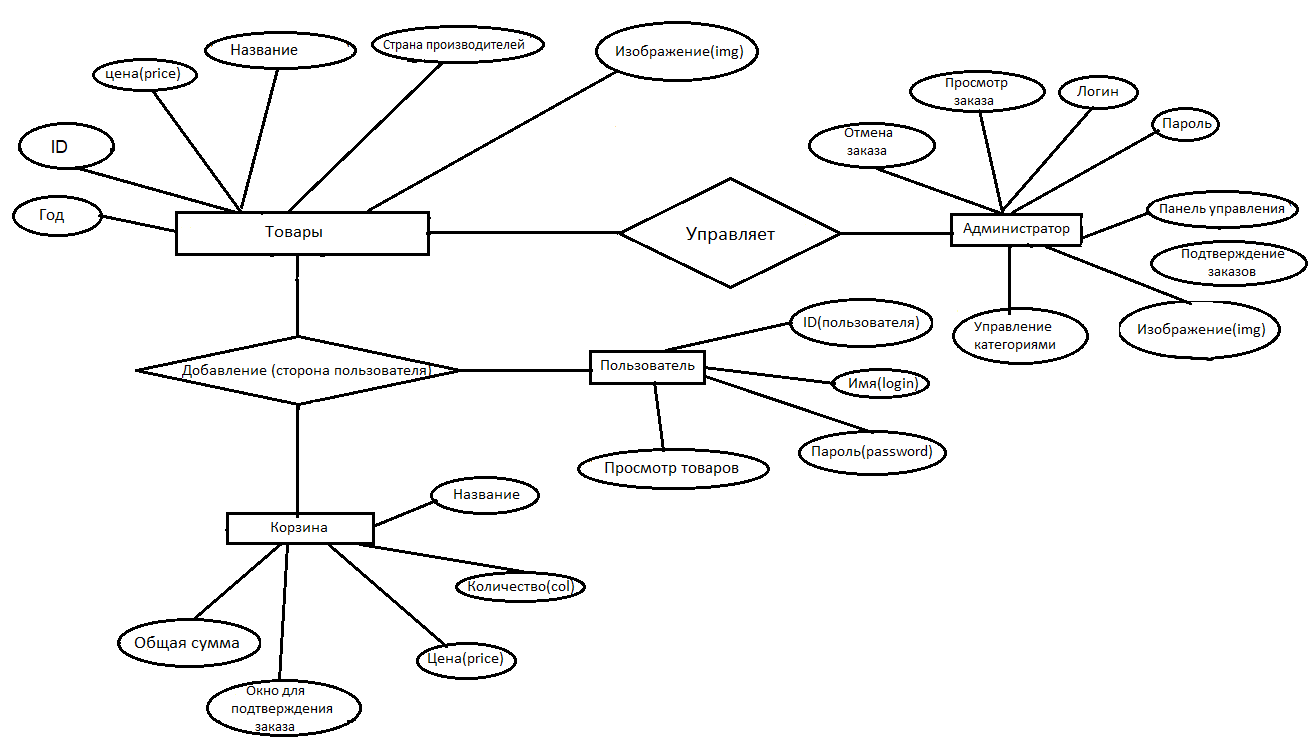


Рисунок 4 - Диаграмма «Сущность-связь»

4. Выбор средств реализации проекта

Тщательно подойдя к выбору технологий для разработки моего проекта, я кратко описала преимущества выбранных платформ.

Язык программирования Php:

* простота использования, простой синтаксис;
* легкость изучения (доступные учебные пособия) и читабельность кода;
* гибкость, т.е. способность смешиваться с другими языками;
* работа с фреймворком yii2;
* универсальность (подходит для различных проектов).

Фреймворк Yii2:

* простой и быстрый (создание сайта из компонентов);
* функциональность (административная панель, формы, миграции и др.);
* безопасность;
* работа с крупными проектами и большой загруженностью;
* существует два заготовленных шаблона проекта для разработчиков с разным уровнем подготовки;
* имеет веб-инструмент для генерации кода, что обеспечивает быстроту его написания;
* имеет замечательную документацию, гайды по старту.

База данных MySQL:

* высокая скорость и производительность,
* простота использования,
* безопасность и простота установки,
* бесплатное обслуживание и открытый доступ.

HTML, CSS и JS фреймворк Bootstrap:

* высокая скорость создания сайта;
* корректная работа сайта во всех поддерживаемых браузерах и ос;
* готовые компоненты и шаблоны;
* элементарные навыки верстки;
* однородность дизайна;
* легкость изучения (огромное количество обучающих материалов).

Среда разработки Visual Studio Code:

* помощь при написании кода (функции автодополнения и анализа кода, подсветка и исправление ошибок);
* встроенные инструменты (инструмент запуска тестов, терминал, инструменты для работы с базами данных);
* поддержка различных веб-фреймворков.

Локальный сервер OpenServer:

* не требует установки на компьютер — его можно запустить с флешки или внешнего жёсткого диска;
* быстро запускается и останавливается;
* сервер автоматически начинает работу при запуске программы;
* можно выбрать режим управления доменами;
* возможна работа через командную строку;
* можно переключать модули http, mysql, php;
* достаточно одного клика для доступа к доменам;
* интерфейс поддерживает несколько языков.

5. Описание этапов реализации ИнтС.

### 5.1. Даталогическое проектирование баз данных.

Разработка базы данных системы является непростой и трудоемкой задачей, так как именно она считается одним из самых важных элементов для информационной системы. При разработке я учитывала проведенный ранее предпроектный анализ.

Разработанная база данных изображена на рисунке 5.

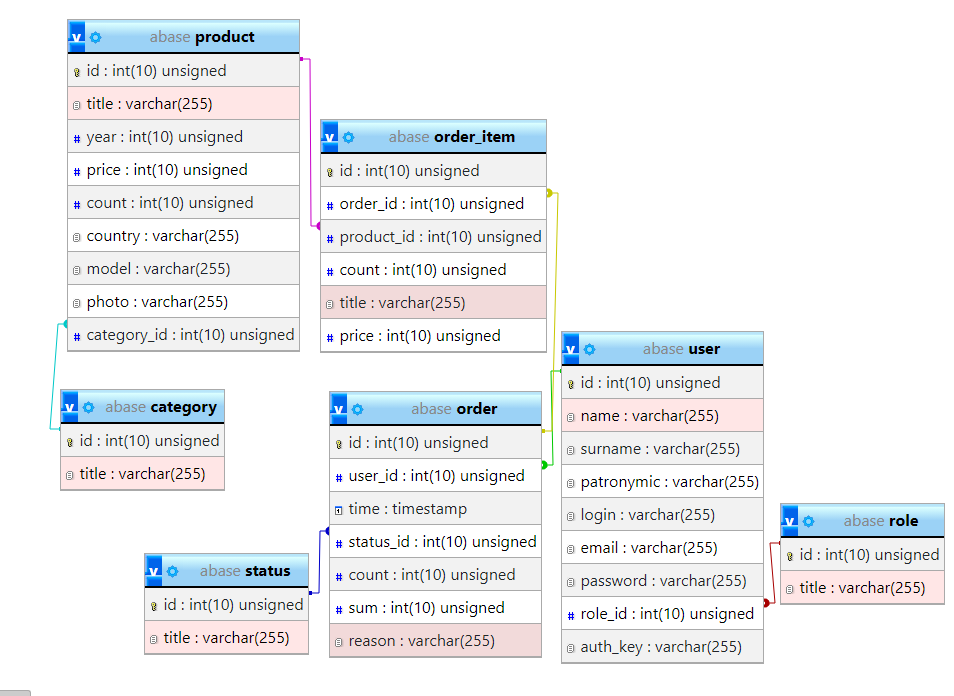


Рисунок 5 – EER диаграмма «база данных»

База данных выполняет следующие функции:

* хранение информации о товарах,
* хранение архива заказов по всем клиентам,
* роли данного сайта,
* оформление заказов,
* в базе можно отслеживать статус заказа.

### 5.2. Поэтапное описание реализации компонентов ИтнС.

Сами страницы подразделены на логические и тематические блоки в зависимости от их наполнения. Для быстрой вёрстки HTML-кода веб-приложения был выбран HTML, CSS и JS фреймворк Bootstrap. Благодаря наличию у него большого количества готовых шаблонов, можно создавать сайт с красивым, приятным, профессиональным дизайном за достаточно короткое время.

**5.2.1 Подготовка.**

После развертывания проекта на сервере, в корне создается файл .htaccess. В нем устанавливается правило для перенаправления запросов, содержащих /web в URL, как показано на рисунке 6.

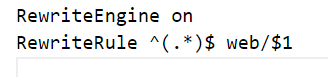


Рисунок 6 – Правила в .htaccess

Далее в файле config/db.php указываются данные для подключения к базе данных – ее название, логин и пароль для авторизации.

В config/web.php необходимо указать различные параметры сайта:

* name – название, указываемое в шапке сайта;
* language – язык сайта;
* cookieValidationKey – ключ для проверки дополнительная безопасность проверки cookie, которая защищает файлы cookie от изменения на стороне клиента;
* baseUrl – базовый путь.

Также раскомментируется блок urlManager в конце файла для активации ЧПУ.

Для получения доступа к возможностям отладки редактируются allowedIPs – туда прописываются разрешенные для доступа адреса, либо просто символ \* (рисунок 7).

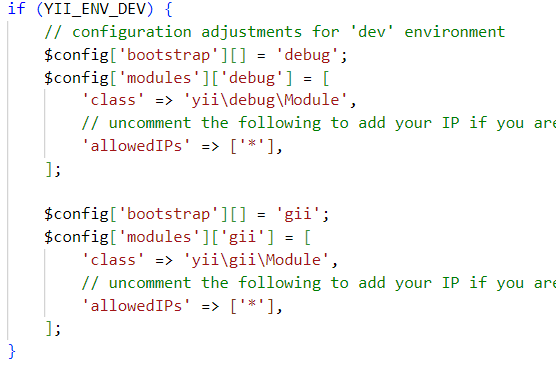


Рисунок 7 – Настройки отладки

**5.2.2 Регистрация.**

Создается копия класса ContactForm под названием RegisterForm.

В ней прописываются свойства и правила валидации полей формы в соответствии с их назначением.

Далее необходимо добавить метод registerUser (рисунок 8), создающий и сохраняющий модель пользователя с указанными данными.

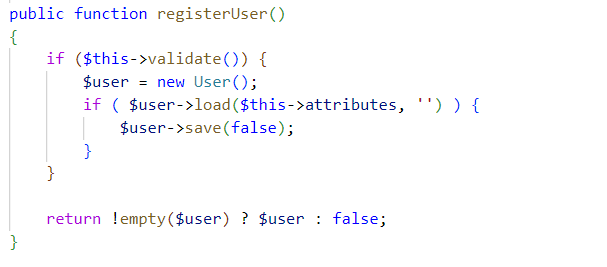


Рисунок 8 – Метод registerUser

В классе SiteController создается метод actionRegister (код на рисунке 9), в котором происходит рендеринг страницы регистрации и вызов метода registerUser с последующей авторизацией и переадресацией на главную страницу.



Рисунок 9 – Метод actionRegister

В папке views/site копируется contact.php. Там прописываются все необходимые поля и их типы в соответствии с заданием. Также устанавливается свойство enableAjaxValidation для поля email. Оно используется для проверки данных без перезагрузки страницы.

В базу данных добавляются роли пользователей. Создается статический метод getRoleId в классе Role для получения id роли по ее названию, как показано на рисунке 10.

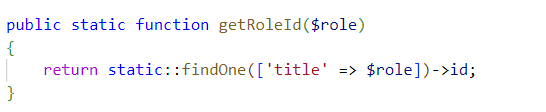


Рисунок 10 – Метод getRoleId

Класс User дорабатывается для использования интерфейса IdentityInterface. В него включаются все требуемые методы из документации, а также добавляются следующие методы:

* beforeSave – создание хэша пароля, ключа аутентификации, установка роли, непосредственно перед сохранением модели;
* findByEmail – поиск экземпляра класса User по его адресу электронной почты с помощью статического метода findOne;
* getIsAdmin – определение, является ли текущий пользователь администратором;
* validatePassword – сравнение пароля с его хэшем в базе данных.

После этого необходимо добавить ссылку на страницу регистрации в файле шаблона приложения, по аналогии с уже имеющимися ссылками.

**5.2.3 Авторизация.**

Класс LoginForm дорабатывается в соответствии с заданием – свойство username переименовывается в login, добавляется вывод flash-сообщения об ошибке в случае, если введенные данные неверны (рисунок 11).

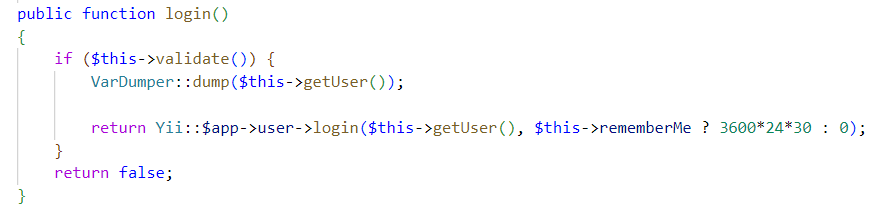


Рисунок 11 – Метод login

Из view удаляется все, кроме самой формы и кнопки входа.

**5.2.3 Панель администрирования.**

В Gii создаётся модуль admin. Далее генерируются CRUD для всех моделей, кроме user и role. Все CRUD должны находиться внутри модуля. Для корректной работы необходимо включить Pjax при генерации. Пример заполненной формы показан на рисунке 12.

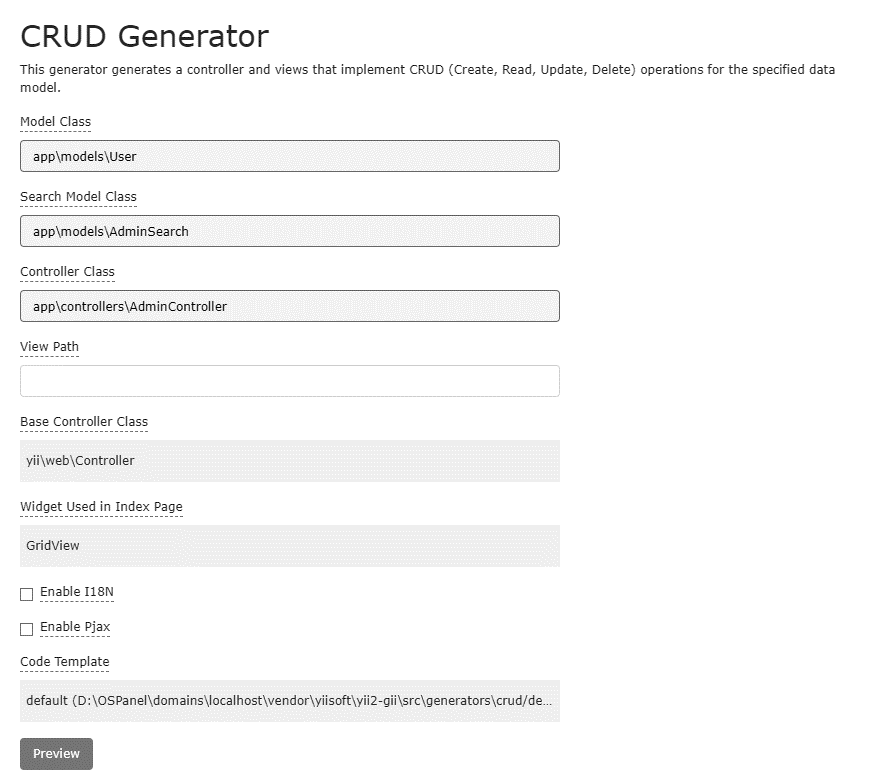


Рисунок 12 – Генерация CRUD

Для каждого CRUD настраивается вывод данных – добавляются выпадающие списки, настраивается фильтрация и сортировка. Для ввода дат используется библиотека.

С помощью генератора кода gii произвести генерацию CRUD контроллера: (CategoryController) для управления категориями, с использованием модели для поиска (CategorySearch).

**5.2.4 Реализация ограничения доступа к контроллеру администратора.**

Используя поведение behaviors и фильтр доступа AccessControl, настроить доступ к контроллеру панели администратора только зарегистрированного пользователя - администратора. Доступ определять через вызов matchCallback для правил и denyCallback (рисунок 13).

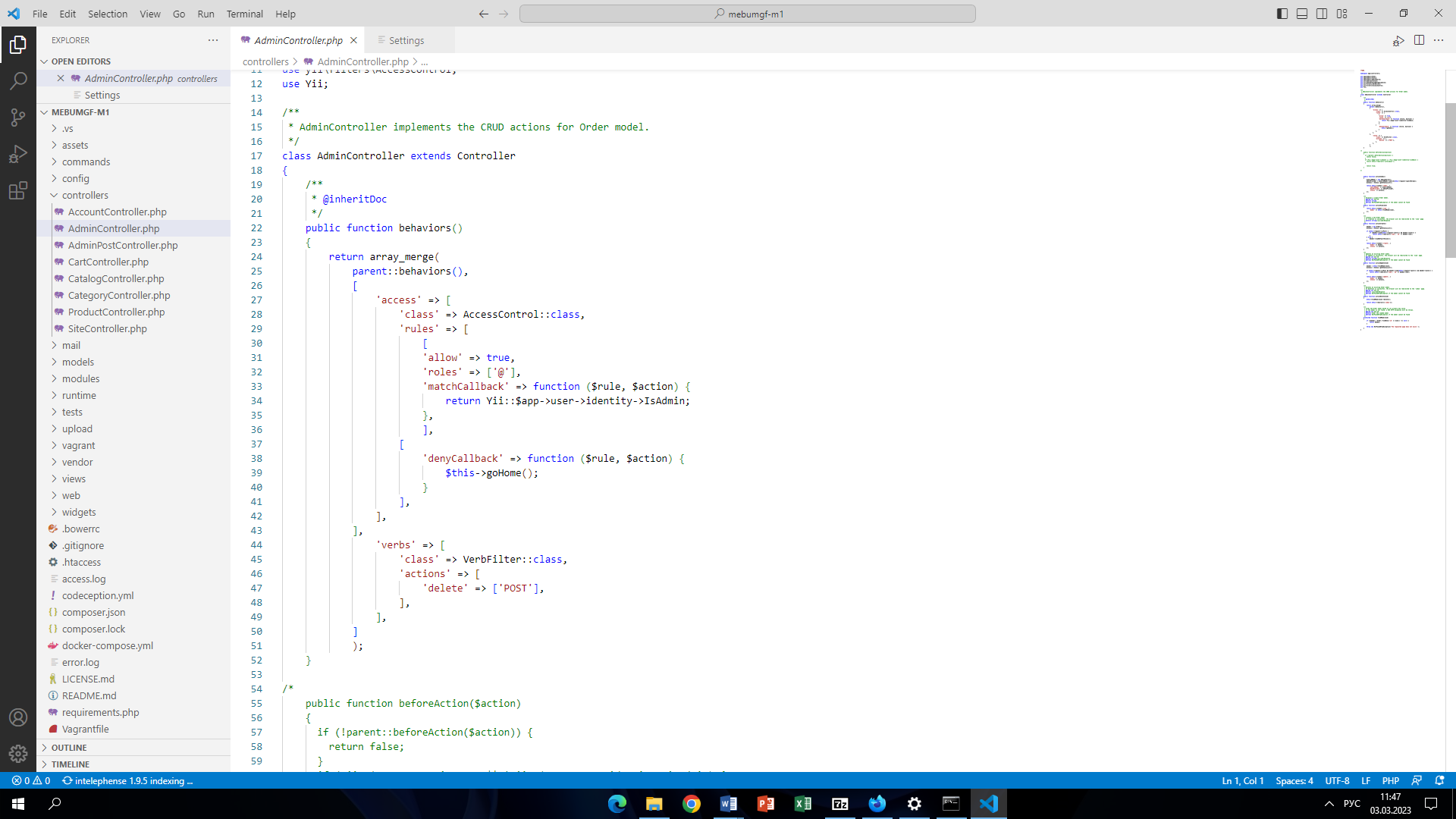


Рисунок 13 – Реализация ограничения доступа к контроллеру администратора

С помощью генератора кода gii произвести генерацию CRUD контроллера:

* CRUD построить на основе модели с товарами, с использованием модели для поиска (ProductSearch);
* Название контроллера установить ProductController.

На главной странице личного кабинета администратора создать две кнопки:

* «Управление категориями» с url - 'admin/category/index',
* «Управление товарами» с url - ' admin/product/index'.

В файле конфигурации приложения (config/web.php), в секции «urlManager», параметра «rules» создать правило для доступа в контроллеры «админки»: 'admin/<controller>/<action>' => '<controller>/<action>'.

**5.2.5 Управление категориями.**

На главной странице контроллера с категориями создать две кнопки (ссылки):

* ведущую на главную страницу ЛК администратора;
* ведущую на главную страницу управления товарами администратором.

Используя поведение behaviors с фильтром доступа AccessControl или метод контроллера beforeAction, настроить доступ к контроллеру управления категориями только зарегистрированного пользователя - администратора. Настройку произвести аналогично доступу к панели администрирования.

Доработать модель Категория, настроив корректное отображение полей модели. Доработать таблицу с категориями на главной странице, настроив корректное отображение столбцов. Настроить форму создания категории, настроив корректное отображение полей ввода. Проверить чтобы на всех страницах использовалась только кириллица.

На страницах представления (вида) использовать библиотеку bootstrap:

* use yii\helpers\Html заменить на use yii\bootstrap5\Html,
* use yii\widgets\ActiveForm заменить на use yii\bootstrap5\ActiveForm.

Внешний вид готовой панели представлен на рисунке 14. В приложении А представлен полный код алгоритма.

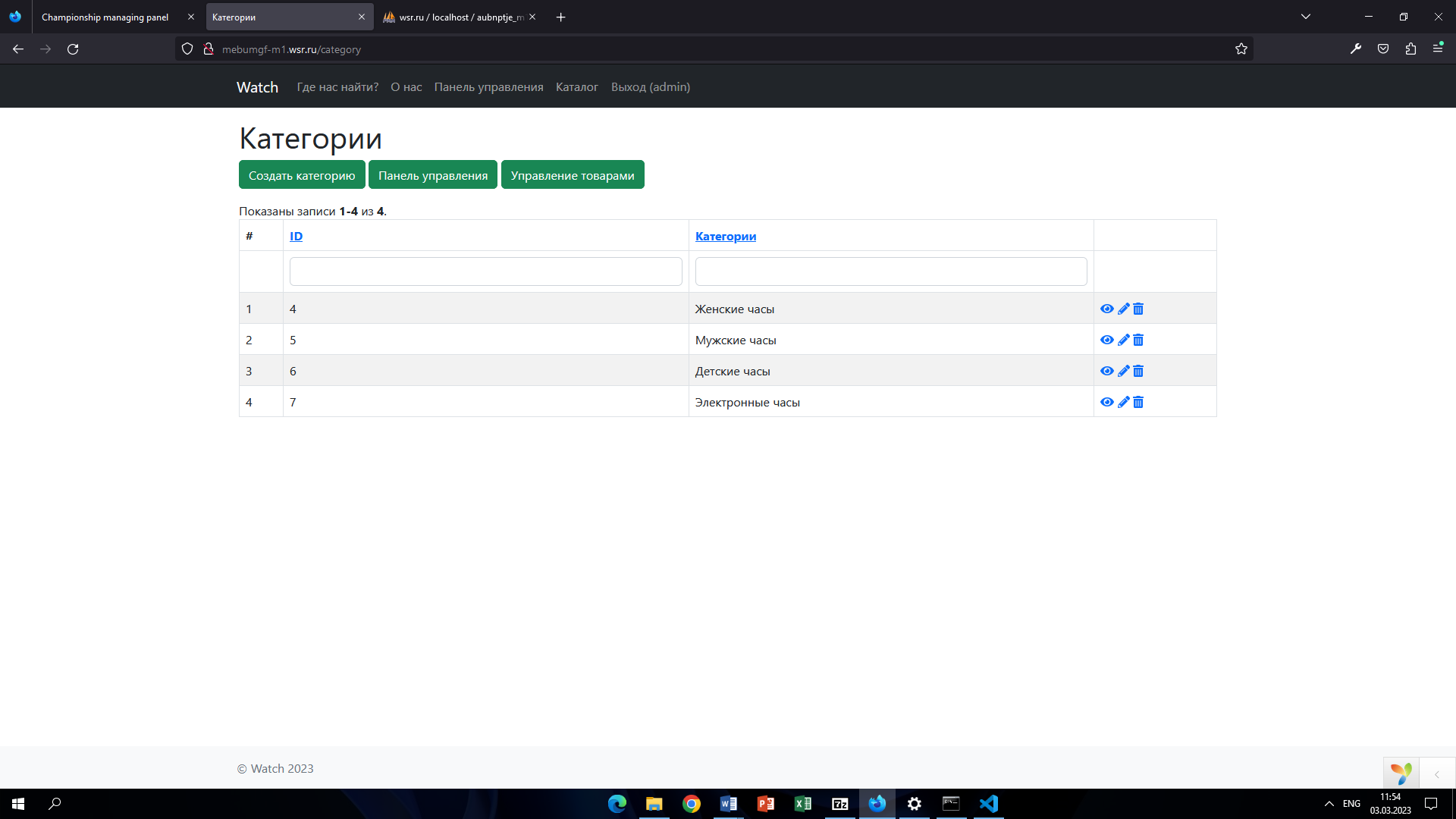


Рисунок 14 – Панель управления категориями

**5.2.6 Управление товарами.**

На главной странице контроллера с категориями создать две кнопки (ссылки):

* ведущую на главную страницу ЛК администратора;
* ведущую на главную страницу управления категориями администратором.

Используя поведение behaviors с фильтром доступа AccessControl или метод контроллера beforeAction, настроить доступ к контроллеру управления товарами только зарегистрированного пользователя - администратора. Настройку произвести аналогично доступу к панели администрирования.

На страницах представления (вида) использовать библиотеку bootstrap:

* use yii\helpers\Html заменить на use yii\bootstrap5\Html,
* use yii\widgets\ActiveForm заменить на use yii\bootstrap5\ActiveForm;

Удалить breadcrumbs («хлебные крошки») на страницах категорий.

Доработать таблицу с товарами на главной странице, настроив корректное отображение столбцов. Настроить форму создания товара, настроив корректное отображение полей ввода. Проверить чтобы на всех страницах использовалась только кириллица.

Внешний вид готовой панели представлен на рисунке 15. В приложении А представлен полный код алгоритма.

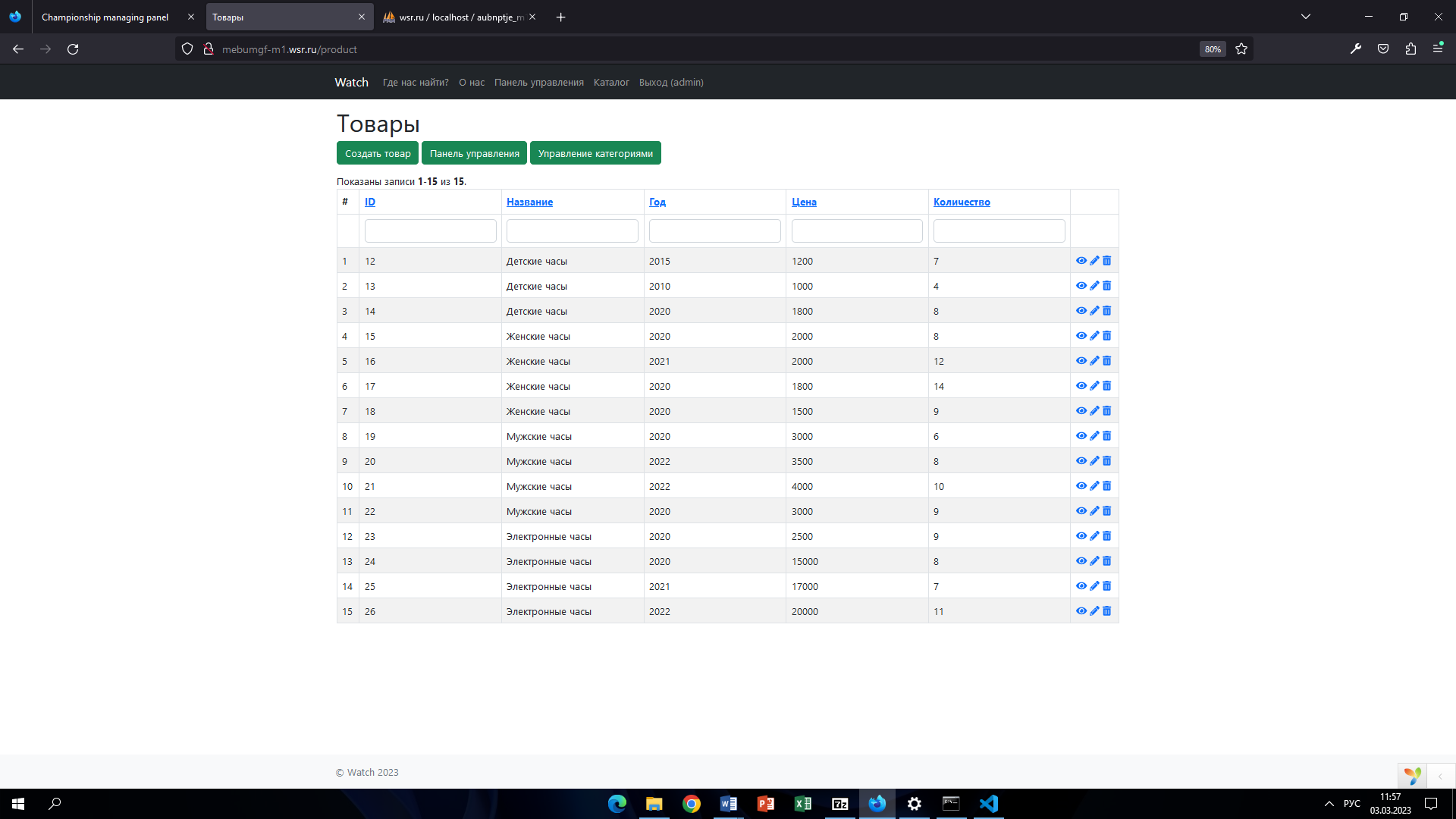


Рисунок 15 – Панель управления товарами

**5.2.7 Настройка информационных страниц.**

Разработка страницы «Где нас найти?»:

* в папке views/site создать файл-представление, в котором разместить разделы и контент согласно заданию;
* в контроллере SiteController создать action в котором вызвать ранее созданный «вид».

Разработка страницы «О нас»:

* в модели Товар, разработать статический метод, возвращающий массив с 5 последних добавленных товаров с информацией о товаре (название, изображение);
* в SiteContoller в методе About создать переменную, содержащую список последних 5 товаров, вызвав ранее созданный метод. Передать созданную переменную в представление;
* в представлении создать переменную содержащую пустой массив;
* заполнение массива, произвести путем перебора массива, переданного в представление, элементы заполняемого массива должны содержать информацию для слада: ключ ‘content’ - изображение товара, а ключ ‘caption’ - его описание;
* добавить на страницу виджет bootstrap5 «Карусель»: yii\bootstrap5\Carousel;
* передать в виджет ранее созданный массив, в качестве списка слайдов в параметр ‘items’;
* настроить внешний вид «карусели» и слайдов на основании компонентов bootsrap5;
* внести изменение в файл конфигурации config/web.php, создав параметр 'defaultRoute' => ***'****site*/about***'.***

Внешний вид готовой страницы представлен на рисунке 16. В приложении А представлен полный код алгоритма.

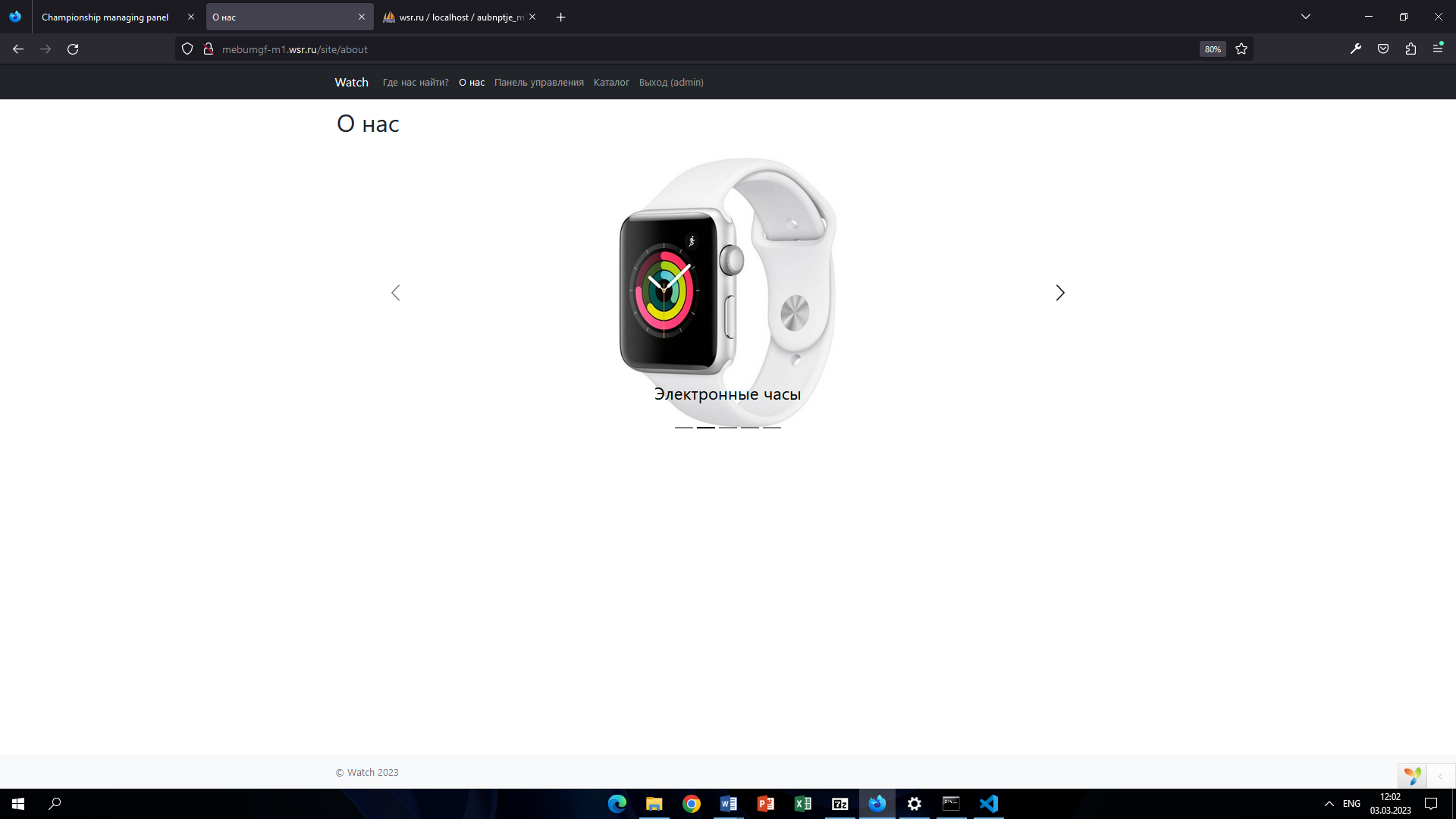


Рисунок 16 – Страница «О нас»

**5.2.8 Создание каталога товаров.**

Для создания и настройки контроллера необходимо:

* с помощью генератора кода gii создать CRUD контроллер Каталога товаров, на основе модели Товар, используя поисковую модель. При генерации контроллера, на главной странице использовать widget listView, а также отметить пункт для включения Pjax;
* удалить в контроллере каталога не используемые методы: create, update и delete;
* в модели поиска доработать SQL запрос, согласно заданию (вывод только существующих товаров).

В файле вида:

* для контейнера Pjax установить параметры: enablePushState => false, timeout => 5000, задать id для контейнера Pjax;
* удалить кнопку создания продукта;
* временно закомментировать вывод формы поиска;
* настроить ListView установив параметры:
* 'layout' => '{pager}<div class="row">{items}</div>{pager}',
* 'pager' => ['class' => \yii\bootstrap5\LinkPager::class];
* задать атрибут id для блока «оборачивающего» Pjax.

В контроллере каталога, методе index, вызвать статический метод, возвращающий индексированный массив с категориями, передать его в представление и далее в форму поиска (временно закомментированную).

В файле виде - шаблоне приложения, добавить пункт Каталог, если это не было сделано ранее со ссылкой на главную страницу каталога.

**5.2.9 Создание сортировки и фильтрации в каталоге товаров.**

Настройка сортировки.

В файле поисковой модели каталога:

* + доработать запрос к бд согласно заданию (выбрать товары только которые есть в наличии);
  + настроить сортировку товаров по умолчанию, для этого, у ActiveDataProvider настроить параметр sorter, указав в нем параметр defaultOrder с названием поля и типом сортировки по умолчанию (рисунок 17).

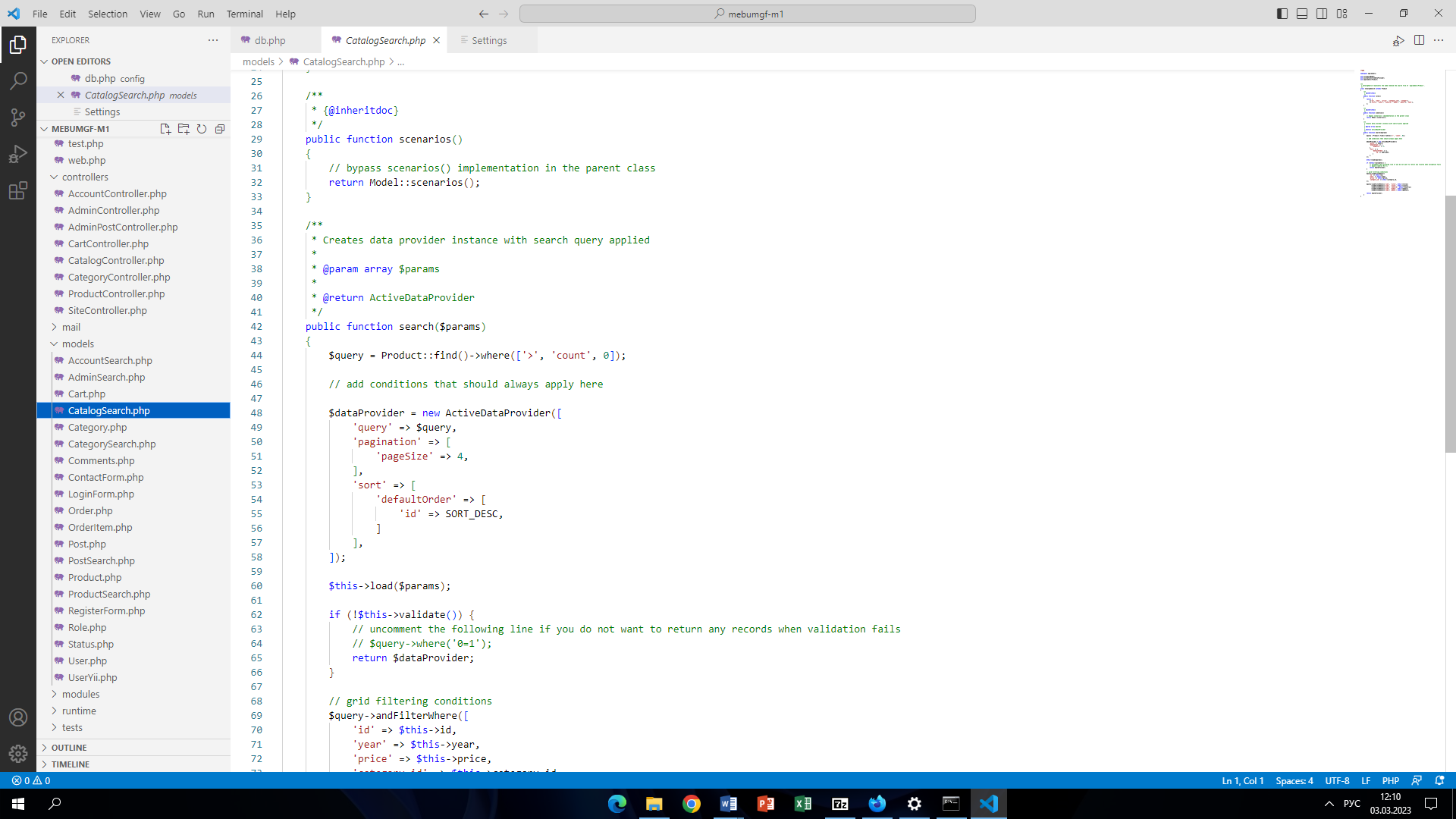


Рисунок 17 – Сортировка товаров

В файле вида каталога:

* + раскомментировать показ формы поиска;
  + дополнительно передать в форму поиска переменную, содержащую DataProvider, используемый в ListView.

Создание сортировки:

* в файле с формой создайте два блока расположенные в одной строке. В левом блоке будет находится сортировка, в правом фильтрация. Код формы поиска перенесите в правый блок;
* в левом блоке укажите поля, по которым необходимо произвести фильтрацию согласно заданию, для это для каждого поля сортировки, необходимо вызвать сортировку у DataProvider по данному полю. Для это необходимо обратиться у DataProvider к методу getSort(), а далее к методу link(‘имя поля для сортировки’);
* оформить внешний вид в соответствии с темой оформления сайта.

Настройка фильтрации.

В контроллере каталога, в методе index, сформировать переменную, содержащую список категорий, для это необходимо вызвать метод из модели Категории, возвращающий проиндексированный массив со списком категорий. Переменную с массивом передать в представление, а далее передать ее в форму поиска.

В форме поиска, в опциях, установить дата-атрибут ‘data-pjax’ => 1.

В форме поиска оставить только поле, связанное с категорией товара. Данное поле реализовать выпадающим списком dropDownList. Для выпадающего списка задать параметр prompt, указав в нем «все категории».

Существующую кнопку сброса на форме убрать. Дополнительно под блоком с сортировкой и фильтрацией разместить кнопку сброса, в виде ссылки на главную страницу каталога.

Внешний вид готового каталога с сортировкой и фильтрацией представлен на рисунке 18. В приложении А представлен полный код алгоритма.

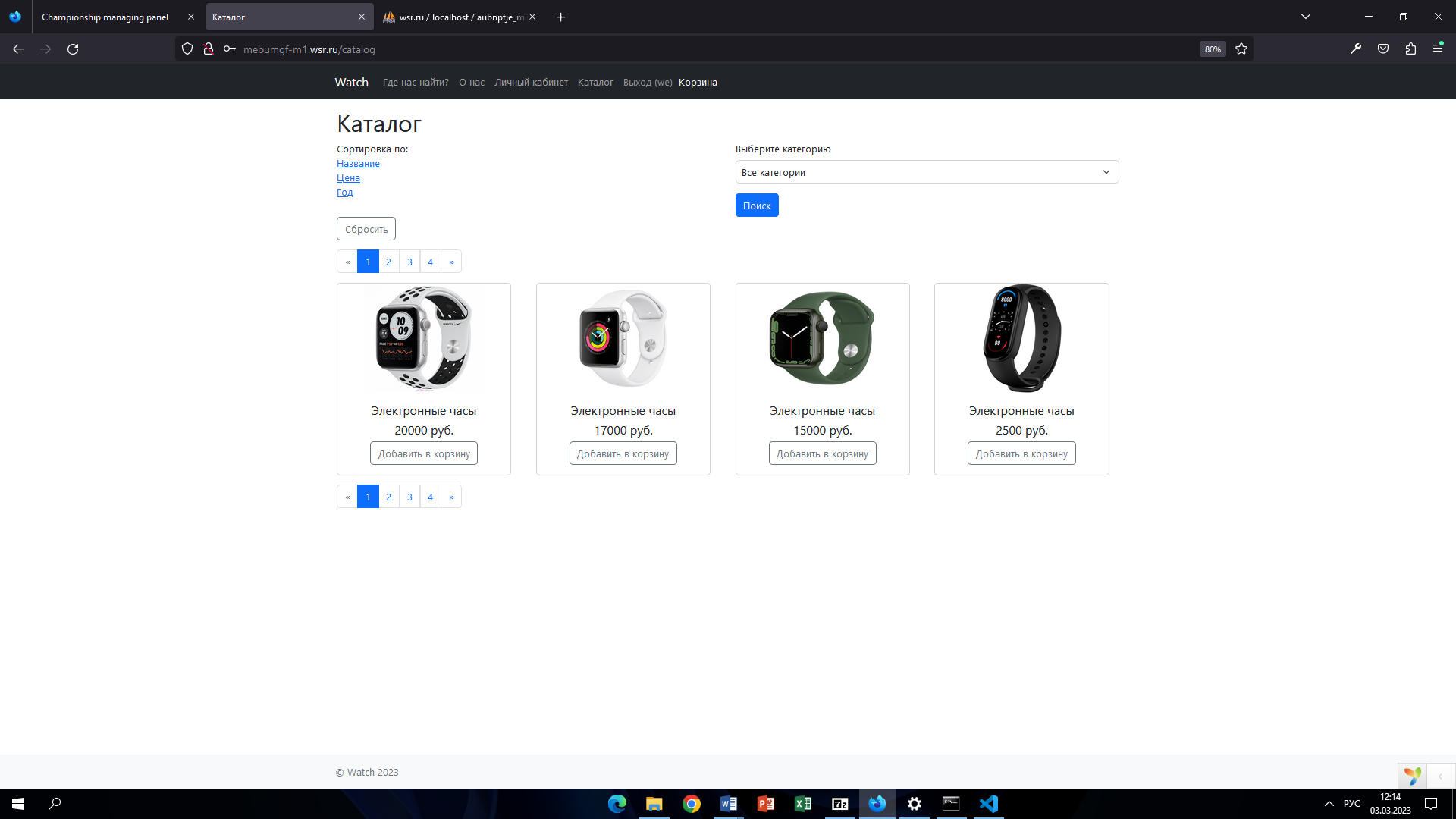


Рисунок 18 – Каталога с сортировкой и фильтрацией

## ЗАКЛЮЧЕНИЕ

В ходе выполнения мной курсовой работы, на основе предпроектного исследования была выявлена актуальность работы, а также особенности создания систем данного вида. Опираясь на результаты этих исследований, была построена структура системы и впоследствии разработан интернет-магазин.

При создании сайта были использованы самые популярные технологии, такие как HTML, CSS, JavaScript, PHP, Yii2. Полученный сайт является адаптивным и имеет четкую логическую структуру, что без особого труда позволит привязать данный веб-сайт к какой-либо системе управления сайтами, использовать и развивать сайт в дальнейшем.

Разработанный веб-сайт имеет понятный и простой пользовательский интерфейс, что обеспечит одобрение сайта со стороны клиентов интернет-магазина. Реализованный сайт позволяет отображать информацию о товарах, о категориях товаров, позволяет клиентам добавлять товары в корзину, оформлять заказ.

Постоянно увеличивается число пользователей интернета, и лиц осуществляющих покупки посредством него. Данный проект является удобной web-системой, которая дает возможность пользователям заказать любую продукцию, не выходя из дома или офиса. Интернет-магазин удобен для людей, которым не хватает времени для походов по магазинам. Данный веб-сайт создан для русскоговорящей аудитории. Сайт можно использовать для продаж товаров Интернет-магазина на территории стран СНГ.

Веб-сайт «Интернет-магазин часов» поможет организации, желающей продавать свои товары через Интернет, расширить свою клиентскую базу и увеличить прибыль.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Справочная документация по языку Php. - URL: [https://www.php.net](https://www.php.net/).
2. Официальный сайт фреймворка Yii. - URL: [https://www.yiiframework.com](https://www.yiiframework.com/).
3. Документация MySQL базы данных. - URL: [https://dev.mysql.com](https://dev.mysql.com/).
4. Документация Bootstrap. - URL: [https://bootstrap-4.ru](https://bootstrap-4.ru/).
5. Документация Visual Studio Code. - URL: [https://code.visualstudio.com](https://code.visualstudio.com/).
6. Документация Open Server Panel. - URL: [https://ospanel.io](https://ospanel.io/).
7. Методы авторизации Yii. Настройка RBAC Manager. - URL: <https://www.yiiframework.com/doc/guide/2.0/ru/security-authorization>.

Приложение А  
 (обязательное)  
Программные коды

Код генерации категорий:

public function actionCreate()

{

$model = new Category();

if ($this->request->isPost) {

if ($model->load($this->request->post()) && $model->save()) {

return $this->redirect(['view', 'id' => $model->id]);

}

} else {

$model->loadDefaultValues();

}

return $this->render('create', [

'model' => $model,

]);

}

Код генерации и редактирования товаров:

public function actionCreate()

{

$model = new Product();

$model -> scenario = Product::CREATE\_PRODUCT;

$category = Category::getCategoryList();

if ($this->request->isPost) {

if ($model->load($this->request->post()) ) {

$model->imageFile = UploadedFile::getInstance($model, 'imageFile');

if($model->upload()) {

if($model->save(false)){

return $this->redirect(['view', 'id' => $model->id]);

}

}

}

}

/\*else {

$model->loadDefaultValues();

} \*/

return $this->render('create', [

'model' => $model,

'category' => $category,

]);

}

public function actionUpdate($id)

{

$model = $this->findModel($id);

$category = Category::getCategoryList();

$file\_name = $model->photo;

if ($this->request->isPost) {

if ($model->load($this->request->post()) ) {

$model->imageFile = UploadedFile::getInstance($model, 'imageFile');

if($model->imageFile){

if( !$model->upload()) {

Yii::$app->session->setFlash('error', 'Ошибка при добавлении товара.');

}

if( file\_exists('upload/' . $file\_name))

unlink( $file\_name);

}

if($model->save(false)) {

Yii::$app->session->setFlash('success', 'Товар успешно отредактирован.');

return $this->redirect(['view', 'id' => $model->id]);

}

else{

Yii::$app->session->setFlash('error', 'Ошибка при добалении товара.');

}

}

}

return $this->render('update', compact('model', 'category'));

}

Код генерации карусели на странице «О нас»:

<div class="site-about">

<h1><?= Html::encode($this->title) ?></h1>

<?php

$items = [];

foreach($products as $product) {

$items[] = ['content' => "<img src='{$product [ 'photo']}' class='d-block text-center mx-auto' style='width:500px; height:500px;' alt=''>",

'caption' => "<h3 class='carousel-caption d-none d-md-block'>{$product['title']}</h3>"

];

}

?>

<?= Carousel::widget ([

'items' => $items,

'options' => ['class' => 'carousel carousel-dark slide',

'data-bs-ride' => 'carousel',

'data-bs-interval' => '5000',

],

]);

?>

</div>

Код генерации карточки товара в каталоге:

<div class="card text-center mb-3" style="width: 18rem;">

<?= Html::a(Html::img( $model->photo, ['alt' => $model->title, 'width' => 200, 'height' => 180]), ['view', 'id' => $model->id]) ?>

<div class="card-body">

<h5 class="card-title"><?= $model->title ?></h5>

<h5 class="card-title"><?= $model->price ?> руб.</h5>

<?php if (!Yii::$app->user->isGuest && !Yii::$app->user->identity->IsAdmin): ?>

<?= Html::a('Добавить в корзину', [''], ['class' => 'btn btn-outline-secondary cart-add', 'data-product-id' => $model->id]) ?>

<?php endif; ?>

</div>

</div>