

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 25

ПРЕПОДАВАТЕЛЬ

доцент, кандидат тех. наук		Е. М. Линский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ ПО КУРСОВОЙ РАБОТЕ НА ТЕМУ:  
“WEB-каталог для интернет магазина”

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	2055		Грицацев И.А.
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2022

# Web-каталог для интернет-магазина

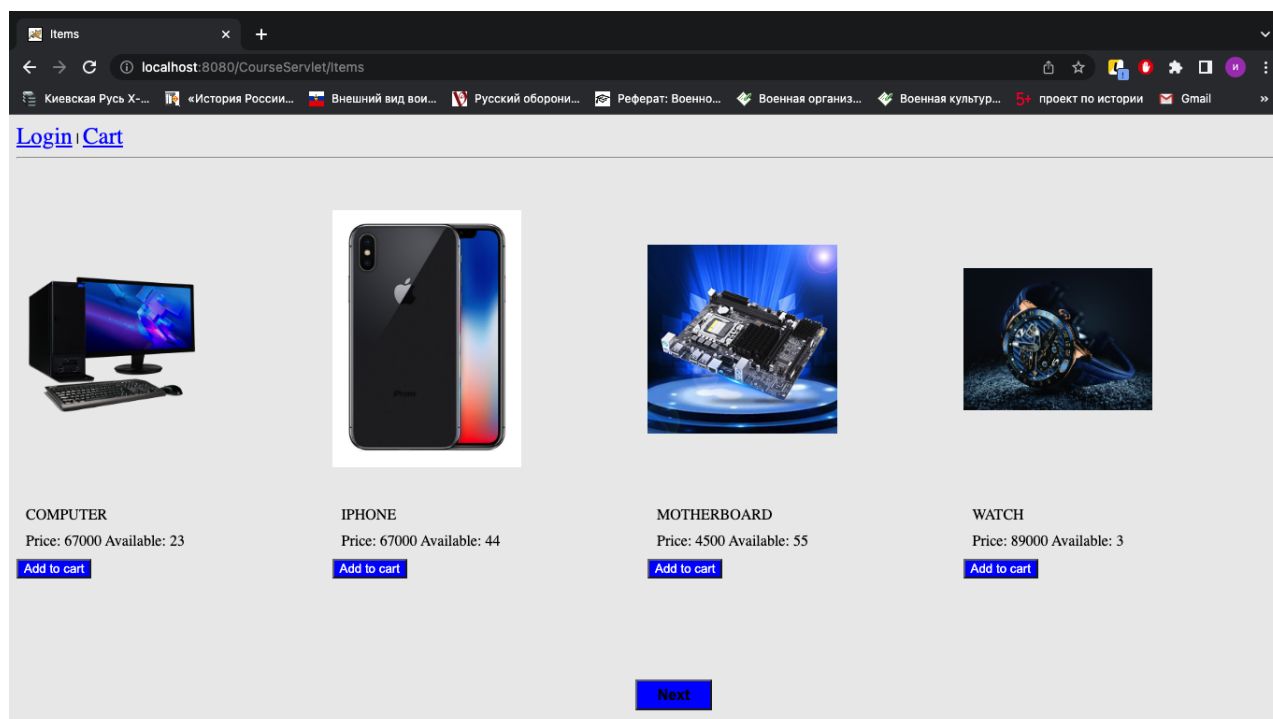
30 октября 2022 г.

## 1 Постановка задачи:

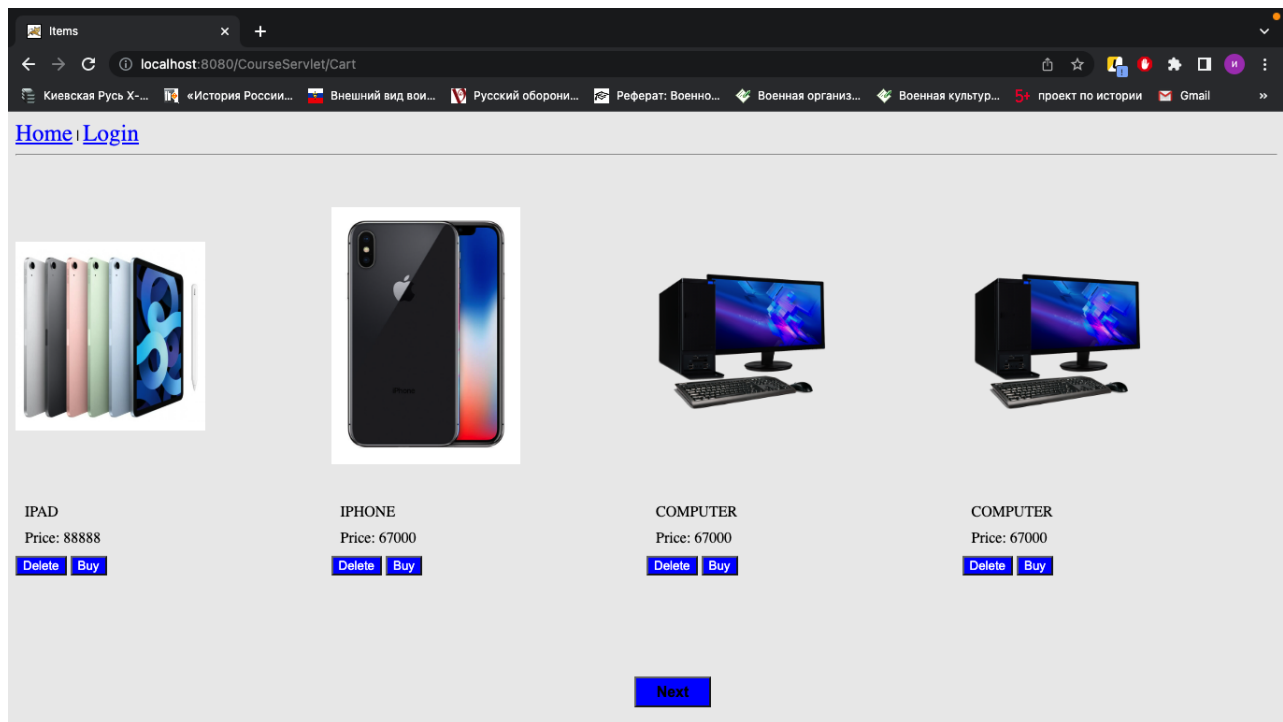
Задачей курсовой работы было создание Web-каталога для интернет магазина, который дает возможность просматривать товары, выложенные с учетной записи продавца, в каждой из карточек товара отобразить изображение товара, название, цену и его количество. Также необходимо было реализовать возможность добавлять товар в корзину, в следствие чего количество товара должно уменьшаться. При авторизации в качестве продавца должна быть предоставлена возможность добавлять новые товары и удалять их. Веб-каталог должен быть разделен на страницы, в случае когда товаров больше, чем может вместить одна страница, а также осуществлять навигацию по этим страницам.

## 2 Руководство пользователя:

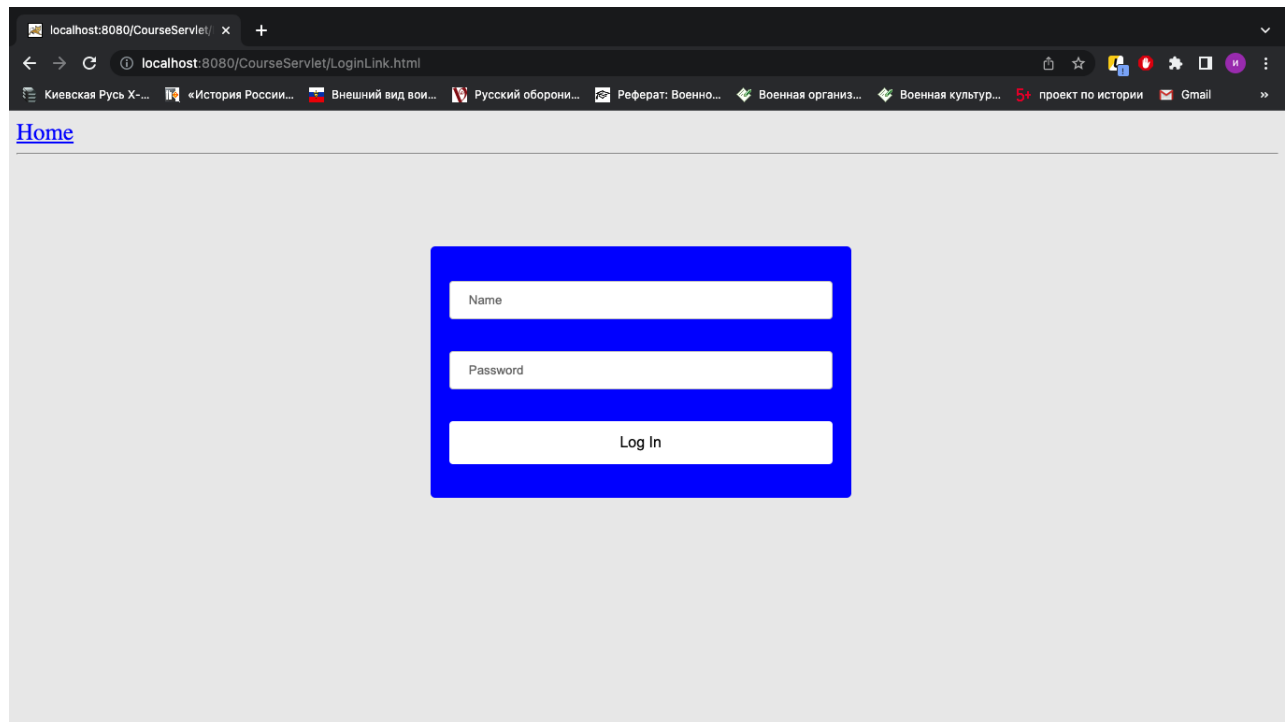
Перейдя по адресу магазина, вы попадаете на главную страницу, на которой расположены карточки товаров, выставленных продавцом, каждая из карточек хранит фотографию, название, цену товара, его количество и кнопку для добавления товара в корзину. При добавлении товара в корзину, количество товара на главной странице уменьшается на 1. В верхней части страницы расположены ссылки для перехода на страницу авторизации в качестве продавца и перехода в корзину. В нижней части находятся кнопки для навигации по страницам каталога, если их больше одной.



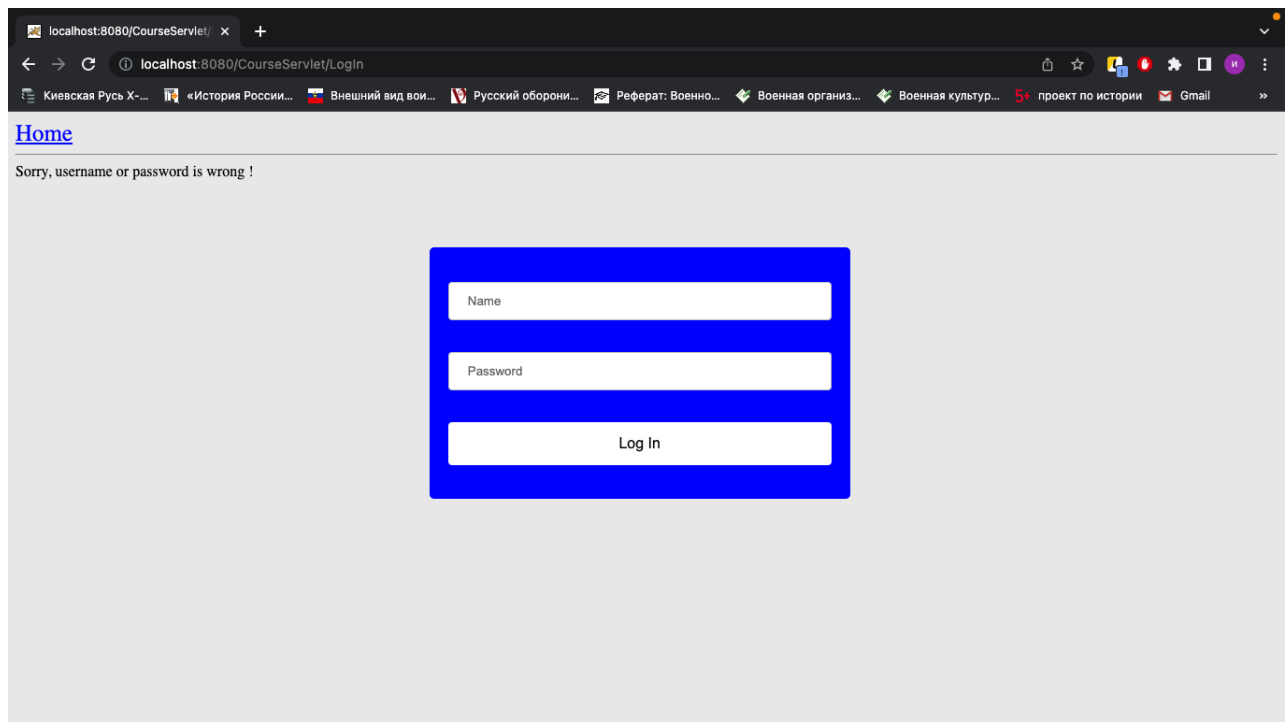
Нажав на ссылку для перехода в корзину, вы увидите карточки всех товаров, которые были в нее добавлены. Каждая карточка в корзине хранит изображение товара, его название, цену и кнопки для удаления товара из корзины и покупки. При нажатии на кнопку удаления товар пропадает из корзины его количество в каталоге возвращается к исходному. При нажатии на кнопку покупки товар исчезнет из корзины. Если это была последняя единица товара в каталоге, то он пропадет из него. В верхней части страницы корзины расположены ссылки для перехода в каталог и на страницу авторизации. Внизу страницы находятся кнопки навигации по страницам корзины, если их больше одной.



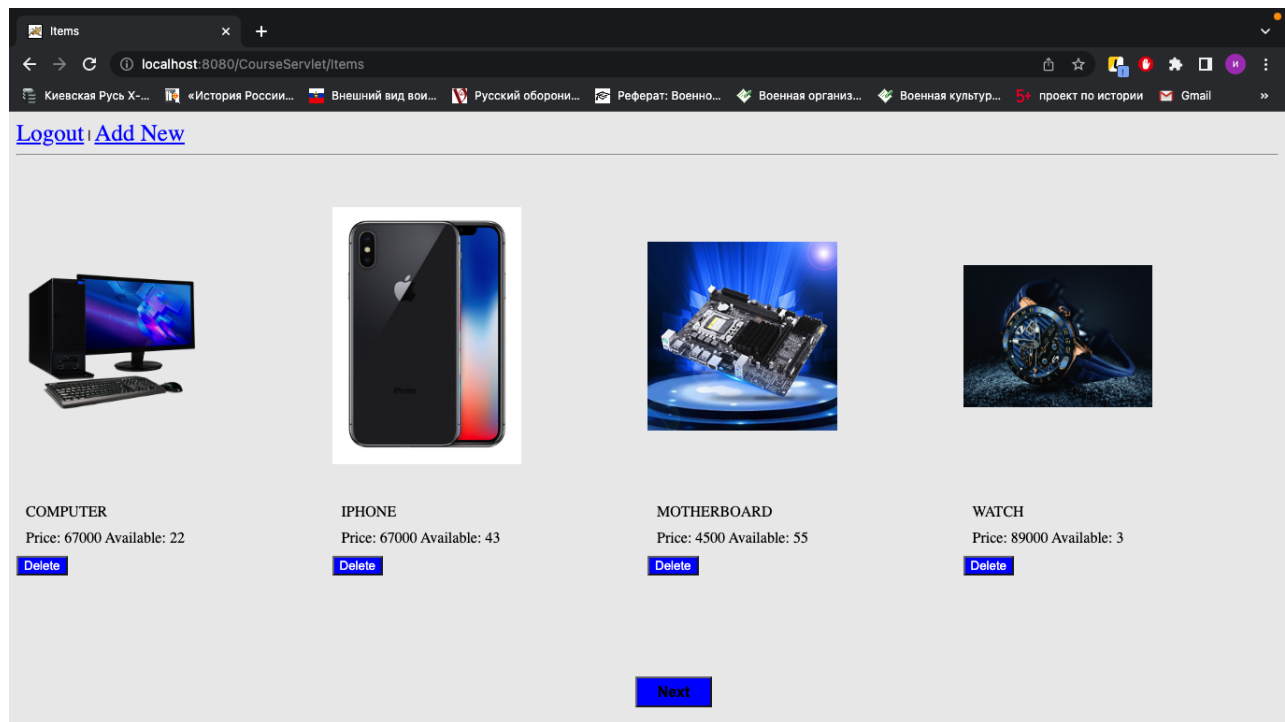
Нажав на ссылку для перехода на страницу авторизации в верхней части страницы, вы увидите страницу с полями для ввода логина и пароля и кнопку для авторизации. В верхней части страницы находится ссылка для возвращения к каталогу. После успешной авторизации вы будете перенаправлены на страницу каталога.



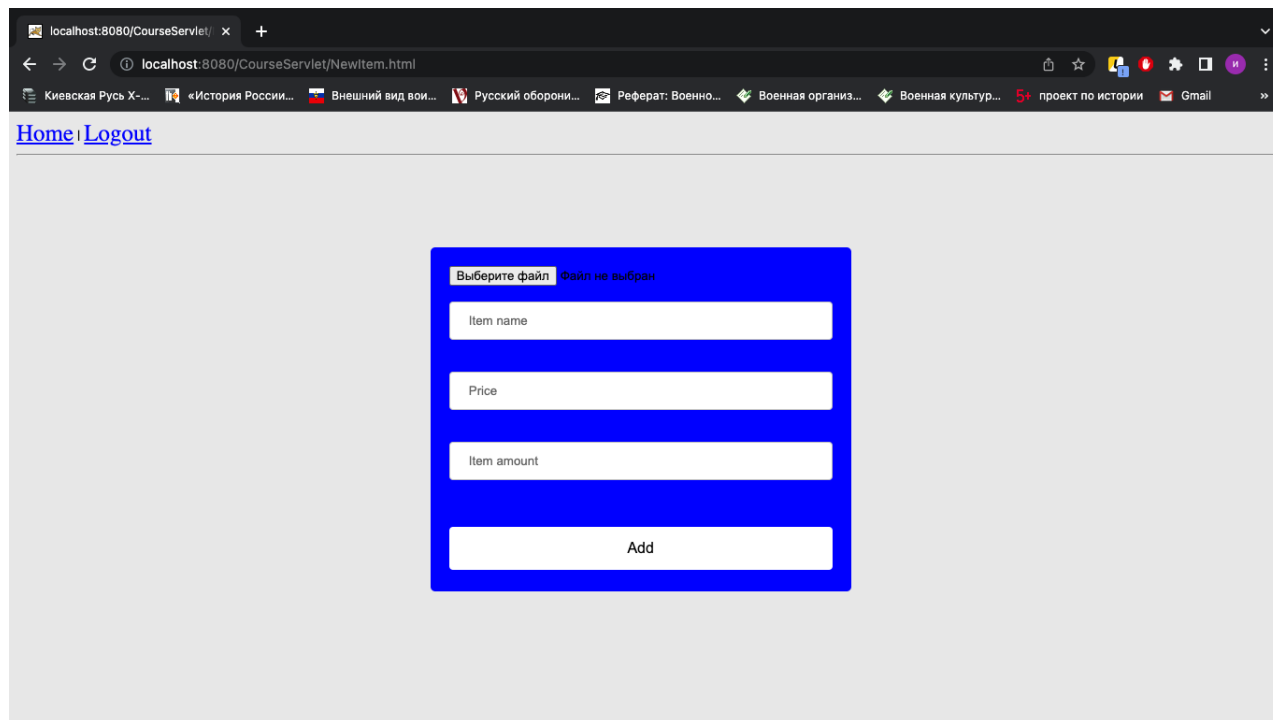
Если же логин и пароль не верны, то появится сообщение о том, что данные не верны и вам будет предложено попробовать снова.



При успешной авторизации в качестве продавца, в карточках товаров исчезнет кнопка для добавления товара в корзины, вместо нее появится кнопка для удаления товара из каталога, по ее нажатию товар сразу пропадает из каталога. В верхней же части страницы каталога теперь расположены ссылки для выхода из аккаунта продавца и перехода на страницу для добавления нового товара. Внизу страницы по прежнему находятся кнопки для навигации по страницам каталога, если их больше одной.



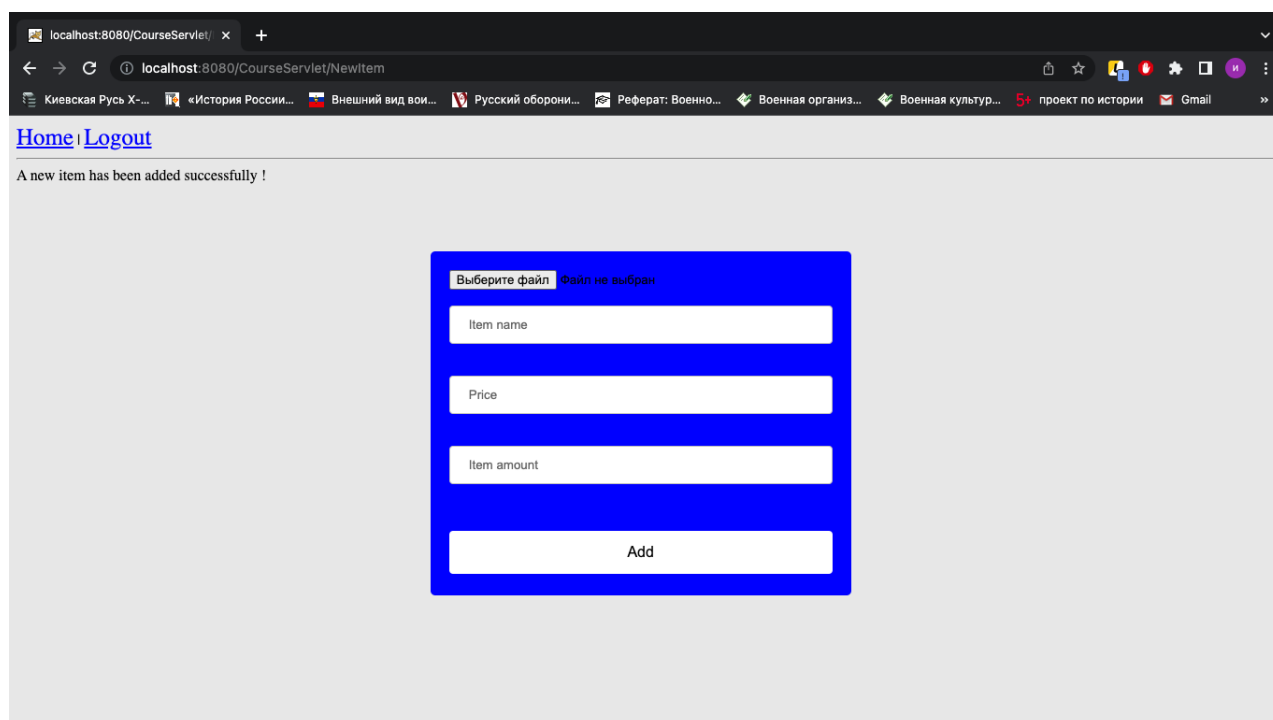
При переходе по ссылке для добавления нового товара отобразится страница с кнопкой для выбора изображения товара с вашего компьютера, поля для ввода названия товара, цены, его количества и кнопка подтверждения. В верхней части страницы расположены ссылки для возвращения к каталогу и выхода из аккаунта продавца.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/CourseServlet/NewItem.html'. The browser's tab bar shows several tabs, including 'Киевская Русь X-...', '«История России...', 'Внешний вид вои...', 'Русский оборони...', 'Реферат: Военно...', 'Военная организ...', 'Военная культур...', '5+ проект по истории', and 'Gmail'. The page content includes a navigation bar with links for 'Home' and 'Logout'. Below the navigation bar, a form for adding a new item is displayed. The form is highlighted with a blue border and contains the following elements:

- A file selection button labeled 'Выберите файл' and a status indicator 'Файл не выбран'.
- An input field labeled 'Item name'.
- An input field labeled 'Price'.
- An input field labeled 'Item amount'.
- An 'Add' button.

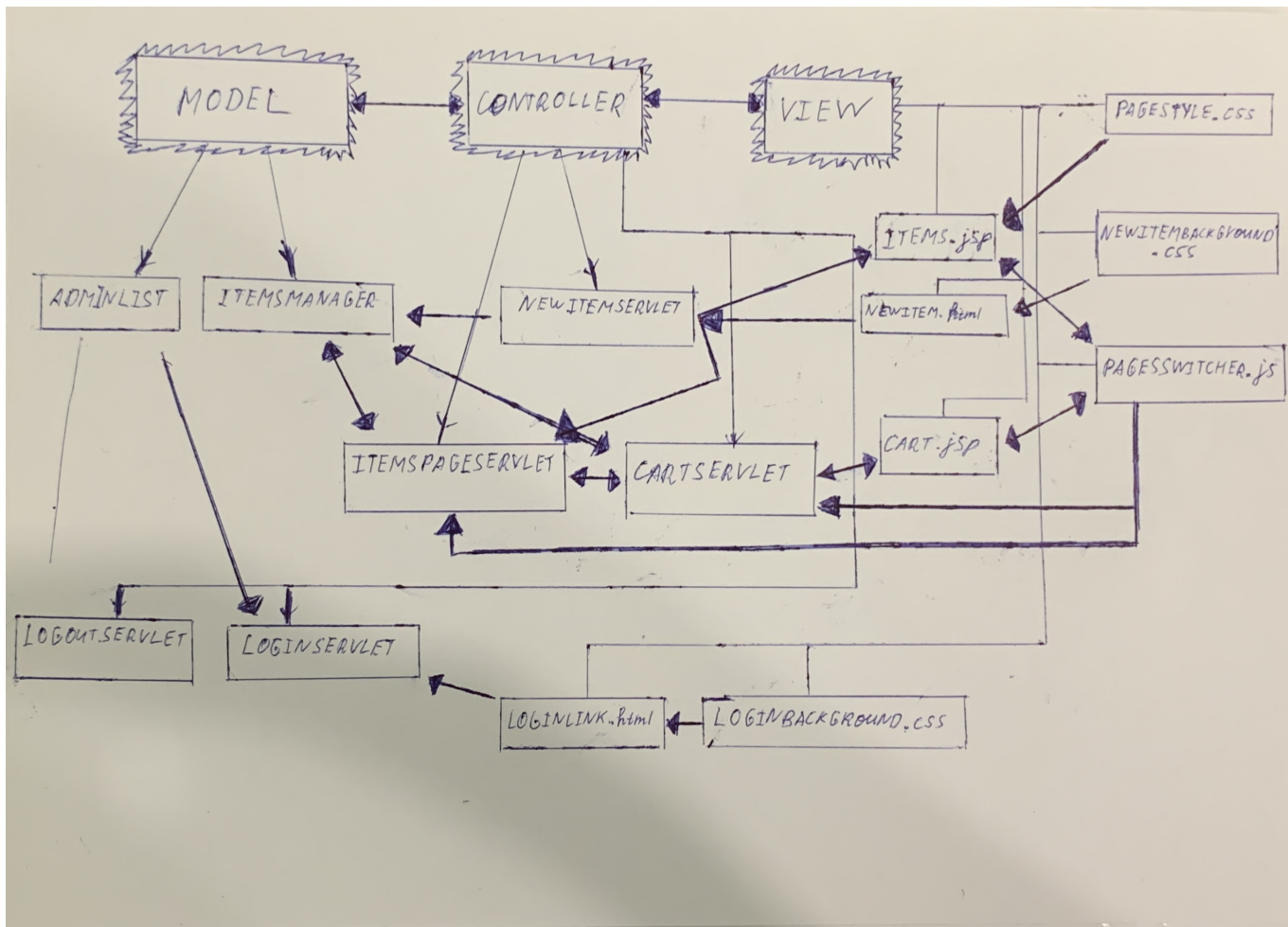
После нажатия кнопки подтверждения появится сообщение об успешном добавлении товара в каталог и будет предложено добавить другой товар.



The screenshot shows the same web browser window as before, but with a success message 'A new item has been added successfully !' displayed above the form. The form itself remains the same, with the following elements:

- A file selection button labeled 'Выберите файл' and a status indicator 'Файл не выбран'.
- An input field labeled 'Item name'.
- An input field labeled 'Price'.
- An input field labeled 'Item amount'.
- An 'Add' button.

### 3 Архитектура программы:



Программа разделена на 3 пакета:

#### 1. Model

- (a) AdminList
- (b) ItemsManager

#### 2. Controller

- (a) CartServlet
- (b) ItemsPageServlet
- (c) LogInServlet
- (d) LogOutServlet
- (e) NewItemServlet

#### 3. View

- (a) Cart.jsp
- (b) Items.jsp
- (c) LoginLink.html
- (d) NewItem.html
- (e) PageStyle.css
- (f) LoginBackground.css
- (g) NewItemBackground.css
- (h) Pageswitcher.js

Controller получает информацию из Model, который взаимодействует с базой данных(текстовыми файлами на сервере) и передает ее во View, затем при взаимодействии пользователя со страницами каталога(добавление, удаление, добавление в корзину, покупка товаров) View передает информацию в Controller, а Controller в свою очередь в Model, который изменяет информацию в базе данных(текстовыми файлами на сервере).

Попав на главную страницу(страница каталога с карточками товаров), пользователь отправляет GET запрос на сервер, который обрабатывает класс ItemsPageServlet из пакета Controller, этот класс в свою очередь обращается к объектам классов ItemsManager и AdminList из пакета Model для получения информации из базы данных(текстовых файлов на сервере) и передает их в качестве атрибутов запроса в Items.jsp(стилизованный при помощи PageStyle.css) из пакета View, на который будет перенаправлен пользователь.

При взаимодействии пользователя с главной страницей(Items.jsp), POST запросы на сервер(обрабатываются классом ItemsPageServlet из пакета Controller, а информация впоследствии заносится в базу данных, используя объект класса ItemsManager из пакета Model) и изменение содержимого страницы происходят при помощи PagesSwitcher.js из пакета View.

Перейдя по ссылке для авторизации, пользователь переходит к LoginLink.html(стилизованный при помощи LoginBackground.css из пакета View) из пакета View, который содержит форму для авторизации, отправляющую POST запросы на сервер, обрабатываемые классом LoginServlet из пакета Controller, который в свою очередь получает информацию для аутентификации при помощи объекта класса AdminList из пакета Model.

При переходе по ссылке для отображения корзины отправляется GET запрос, обрабатываемый классом CartServlet из пакета Controller, который обращается к объекту класса ItemsManager из пакета Model для получения информации о товарах и передает ее в качестве атрибутов запроса в Cart.jsp(стилизованного при помощи PageStyle.css) из пакета View.

Взаимодействие с Cart.jsp сопровождается изменением содержимого страницы и посылкой POST запросов(обрабатываются классом CartServlet из пакета Controller, который заносит информацию в базу данных, используя объект класса ItemsManager из пакета Model) при помощи PagesSwitcher.js из пакета View.

Перейдя по ссылке для добавления нового товара, авторизованный пользователь видитNewItem.html(стилизованный при помощиNewItemBackground.css) из пакета View, который содержит форму для отправки POST запроса с информацией о новом товаре, который обрабатывается классомNewItemServlet из пакета Controller, который в свою очередь обращается к объекту класса ItemsManager из пакета Model для добавления информации о новом товаре в базу данных(текстовые файлы на сервере).

По нажатии на ссылку для выхода из аккаунта продавца отправляется GET запрос, обрабатываемый классом LogoutServlet из пакета Controller.

## 4 Описание наиболее важных классов:

### Список наиболее важных классов:

#### 1. Model

- (a) AdminList
- (b) ItemsManager

#### 2. Controller

- (a) CartServlet
- (b) ItemsPageServlet
- (c) LogInServlet
- (d) LogOutServlet
- (e)NewItemServlet

### Описание наиболее важных методов классов:

#### 1. Model.AdminList

- (a) CheckUser(String name, String password):  
сравнивает логин и пароль, переданные в метод с информацией из базы данных(текстового файла на сервере).
- (b) readFile():  
Считывает информацию из текстового файла на сервере и добавляет ее в HashMap.

#### 2. Model.ItemsManager

- (a) edit(int key, int val):  
Меняет количество доступного товара, переданное в качестве параметра в метод (int val) у товара с id, переданным в данный метод(int key).
- (b) Add(String title, int price, int amount):  
Добавляет информацию о новом товаре в HashMap, хранящий названия, цену и количество товара, присваивая товару уникальный идентификатор.
- (c) delete(int key):  
Удаляет информацию о товаре с идентификатором, переданным в метод и изображение товара с сервера.
- (d) readFile():  
Считывает информацию о товарах из текстовых файлов на сервере и добавляет ее в соответствующие HashMap-ы.
- (e) writeFile():  
Записывает в текстовые файлы на сервере информацию из HashMap-ов.
- (f) SaveImage(InputStream fileContent, String name):  
Сохраняет изображение из входящего потока, переданного в качестве аргумента(fileContent) в метод с именем, переданным в метод(name) в директорию с изображениями товаров на сервере.

#### 3. Controller.CartServlet

- (a) doPost(HttpServletRequest request, HttpServletResponse response):  
обрабатывает POST запросы к серверу, отправленные со страницы корзины: Считывает информацию о товарах из текстовых. файлов на сервере при помощи объекта класса ItemsManager, считывает параметры, переданные через POST запрос. Если отправлен запрос на удаление товара из корзины, удаляет из Cookies данные о данном товаре и убирает его из корзины. Если же отправлен POST запрос на покупку товара, то метод удаляет информацию о данном товаре из Cookies и удаляет информацию о товаре из текстовых файлов на сервере и его изображение из директории, если была куплена единица товара.
- (b) doGet(HttpServletRequest request, HttpServletResponse response):  
обрабатывает GET запросы к серверу, отправленные к CartServlet: при помощи объекта класса ItemsManager считывает информацию о товарах из текстовых файлов на сервере и устанавливает ее в качестве атрибутов запроса и направляет пользователя на Cart.jsp.



#### 4. Controller.ItemsPageServlet

- (a) doPost(HttpServletRequest request, HttpServletResponse response):  
обрабатывает POST запросы, отправленные со страницы каталога на сервер: считывает информацию о товарах из текстовых файлов при помощи объекта класса ItemsManager, и считывает параметры, переданные через POST запрос. Если отправлен запрос на добавление товара в корзину, то добавляется Cookies с id товара и интервалом отсутствия активности 30 минут, а также изменяется количество товара в текстовом файле на сервере при помощи объекта класса ItemsManager. Если же отправлен запрос на удаление товара из каталога, то информация о нем удаляется из текстовых файлов на сервере и удаляется изображение этого товара из директории на сервере.
- (b) doGet(HttpServletRequest request, HttpServletResponse response):  
обрабатывает GET запросы, отправленные к ItemsPageServlet: информацию о товарах из текстовых файлов при помощи объекта класса ItemsManager и устанавливает ее в качестве атрибутов запроса и направляет пользователя на Items.jsp.

#### 5. Controller.LogInServlet

- (a) doPost(HttpServletRequest request, HttpServletResponse response):  
обрабатывает POST запросы, при помощи формы в LoginLink.html: параметры из формы, переданные POST запросом, сверяет данные, полученные из параметров запроса с данными из текстового файла, извлеченными при помощи объекта класса AdminList. Если данные совпадают, то инвалидируется текущая сессия и создается новая с заданным параметром "name- "Admin" с интервалом отсутствия активности 30 минут, затем пользователя перенаправляет на ItemsPageServlet. В случае, если данные не совпадают, пользователю выводится форма для повторного ввода данных для аутентификации с сообщением о неверных данных.
- (b) doGet(HttpServletRequest request, HttpServletResponse response):  
обрабатывает GET запросы, отправленные к LogInServlet: направляет пользователя на LoginLink.html с формой для аутентификации.

#### 6. Controller.LogOutServlet

- (a) doGet(HttpServletRequest request, HttpServletResponse response):  
обрабатывает GET запросы, отправленные к LogOutServlet: инвалидирует текущую сессию и перенаправляет пользователя на ItemsPageServlet.

#### 7. Controller.NewItemServlet

- (a) doPost(HttpServletRequest request, HttpServletResponse response):  
обрабатывает POST запросы, отправленные при помощи формы изNewItem.html кNewItemServlet: считывает параметры запроса и добавляет эту информацию в текстовые файлы на сервере и сохраняет изображение товара в директорию на сервере при помощи класса ItemsManager. После успешного добавления товара выводится страница с соответствующим сообщением и формой для добавления следующего товара.
- (b) doGet(HttpServletRequest request, HttpServletResponse response):  
обрабатывает GET запросы, отправленные кNewItemServlet: направляет пользователя наNewItem.html.