

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И. Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Салихов Нариман Бауржанович

Разработка web-приложения «Интернет-магазин по продаже техники»

ДИПЛОМНЫЙ ПРОЕКТ

Специальность 5В070300 – «Информационные системы»

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И. Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой
КБОиХИ,

канд.техн.наук,
ассистент-профессора

 Н.А. Сейлова

" 14 " 05 20 19 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Разработка web-приложения «Интернет-магазин по продаже
техники»»

по специальности 5В070300 – «Информационные системы»

Выполнил

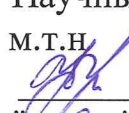
Н.Б. Салихов

Рецензент,
заместитель руководителя Национальной
научной лаборатории коллективного
пользования информационными и
космическими технологиями НАО КазНТУ
имени К.И.Сатпаева, PhD

 Ж.А. Бостанбеков
" 05 " 20 19 г.

Научный руководитель,

М.Т.Н.

 Х. И. Юбузова

" 18 " 05 20 19 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И. Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – «Информационные системы»

УТВЕРЖДАЮ

Заведующий кафедрой,
КБОиХИ,

канд.техн.наук,
ассистент-профессора

 Н.А. Сейлова
" 24 " 05 2019 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Салихову Нариману Бауржановичу

Тема: Разработка web-приложения «Интернет-магазин по продаже техники».

Утверждена приказом Ректором Университета № 1162-б от 16.10.2018 г.

Срок сдачи законченной работы «19» 05 2019 г.

Исходные данные к дипломному проекту: результат литературного обзора современного состояния по данной теме, сбор теоретического материала.

Краткое содержание дипломного проекта:

- а) анализ и обработка современных веб-приложений;
- б) модели и службы функционирования веб-приложения;
- в) информационное и программное обеспечение.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 16 слайдов презентации работы

Рекомендуемая основная литература: из 12 наименований

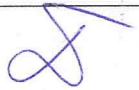
ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Анализ и обзор современных веб-приложений	10.01.2019 – 08.03.2019	
Модели и службы функционирования веб-приложения	05.02.2019 – 10.03.2019	
Информационное и программное обеспечение	11.03.2019 – 28.04.2019	

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов работы

Наименование разделов	Научный руководитель, консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	М.Т.Н М.Б. Бауржан	8.05.19	
Нормоконтролер	Сениор-лектор, PhD О. В. Киселева	8.05.19	

Научный руководитель  м.т.н Х. И. Юбузова

Задание принял к исполнению обучающийся  Н. Б. Салихов

Дата "15" 01 20 19 г.

РЕЦЕНЗИЯ

на дипломный проект

Салихова Наримана Бауыржанұлы

Специальности 5B070300 Информационные системы

На тему: **Разработка Web-приложения «Интернет-магазин по продаже техники»**

Выполнено:

- а) графическая часть на 16 листах
б) пояснительная записка на 30 страницах

ЗАМЕЧАНИЯ К РАБОТЕ

В представленном на рецензирование дипломном проекте разработано Web-приложение «Интернет-магазин по продаже техники»

Пояснительная записка дипломного проекта состоит из 3 разделов.

В первом разделе проведен аналитический обзор теоретических аспектов, анализ и обзор современных технологий по разработке веб-приложений. Во втором разделе разработана функциональная модель и обобщенный алгоритм функционирования веб-приложения «Интернет-магазин по продаже техники», выбрана платформа и дополнительное программное обеспечение для реализации приложения. Третий раздел посвящен непосредственно разработке, рассмотрено информационное и программное обеспечение, представлена структура программного обеспечения веб-приложения «Интернет-магазин по продаже техники», описание работы и апробации веб-приложения.

Пароли пользователей хранятся в БД в открытом виде, необходимо хэшировать пароль с добавлением соли. Не продемонстрирована основная функция системы – добавление покупок в корзину.

Оценка работы

Рецензируемый дипломный проект выполнен на отличном уровне, проект отвечает актуальной проблеме современности, удовлетворяет требованиям, предъявляемых к дипломным проектам. Дипломный проект демонстрирует умение дипломника самостоятельно работать с технической документацией, решать задачи проектирования, реализации Web-приложения, применения технологий MVC, IntelliJ IDEA, Git, MySQL, Insomnia и др.

Считаю, что дипломный проект Салихова Наримана на тему: «Разработка Web-приложения «Интернет-магазин по продаже техники»» заслуживает оценки 98 баллов (Отлично), а Салихов Н.Б. – присвоения академической степени бакалавра по специальности 5B070300 – Информационные системы.

Рецензент

Заместитель руководителя Национальной
научной лаборатории коллективного пользования
информационными и космическими технологиями
ИАО КазНУ им. К.И. Сатпаева, PhD
А. Бостанбеков



2019 г.

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломный проект

Салихова Наримана Бауыржанұлы

5B070300 Информационные системы

Тема: Разработка Web-приложения «Интернет-магазин по продаже техники»

Разработка Web-приложения является актуальной и востребованной сферой деятельности, т.к. приложение для любой компании, организации или фирмы в сети Интернет представляет собой достаточно дешевый и массовый способ рекламы, дает возможность потенциальным и существующим клиентам легко получать необходимую информацию об услугах компании, деловых интересах и ее товарах, что может в дальнейшем оказать содействие в привлечении новых заказчиков и партнеров по бизнесу, а следовательно, способствует увеличению объема продаж и рентабельности предприятия.

Задачами представленного дипломного проекта являются: исследование проблем по созданию Web-приложений, создание веб-ресурса по продаже различных товаров, обеспечивающее своевременное информирование пользователей о скидках и новых поступлениях, новостные данные в любом формате.

Для решения поставленных задач дипломник провел аналитический обзор и анализ существующих технологий по созданию Web-приложений, определил требования, которым должно отвечать разрабатываемое веб-приложение, разработал функциональную модель веб-приложения «Интернет-магазин по продаже техники», структуру веб-приложения, обосновал выбор СУБД и разработал структуру базы данных для приложения.

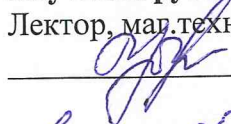
В процессе работы над проектом дипломант проявил творческий подход и показал себя грамотным специалистом, владеющим современными технологиями. Для разработки приложения применены методы разработки веб приложений с использованием языков Java и JavaScript, технологии MVC, IntelliJ IDEA, Git, MySQL, Insomnia и ряда других.

Дипломник Салихов Н. Б. справился с поставленными задачами в полном объеме, показал отличное владение навыками самостоятельной работы с технической документацией, самостоятельность в выборе предметной области, инструментария и проектирования Web-приложений.

Считаю, что дипломный проект на тему «Разработка Web-приложения «Интернет-магазин по продаже техники»» выполнен Салихов Н. Б. на отличном уровне и может быть допущен к защите.

Научный руководитель

Лектор, маг. техн. наук

 Юбузова Х.И.

« 8 »  2019 г.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Салихов Нариман

Название: Разработка Web-приложения «Интернет-магазин по продаже техники»

Координатор: Халича Юбузова

Коэффициент подобия 1:2,7

Коэффициент подобия 2:0

Тревога:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- ☒ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- ☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- ☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....
.....
.....
.....
.....

Дата 14.05.19г

Подпись заведующего кафедрой /

начальника структурного подразделения




Окончательное решение в отношении допуска к защите, включая обоснование:

Допускается к защите!

Дата 14.05.1992

Подпись заведующего кафедрой /

начальника структурного подразделения


КБДехи

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Салихов Нариман

Название: Разработка Web-приложения «Интернет-магазин по продаже техники»

Координатор: Халича Юбузова

Коэффициент подобия 1:2,7

Коэффициент подобия 2:0

Тревога:0

После анализа Отчета подобия констатирую следующее:

- ☒ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- ☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- ☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

обнаруженные замеченные выявлены
факты не подлежат признание
платежа

8.05.19

Дата



Подпись Научного руководителя

Краткий отчет



Университет:	Satbayev University
Название:	Разработка Web-приложения «Интернет-магазин по продаже техники»
Автор:	Салихов Нариман
Координатор:	Халича Юбузова
Дата отчета:	2019-05-04 16:42:39
Коэффициент подобию № 1:	2,7%
Коэффициент подобию № 2:	0,0%
Длина фразы для коэффициента подобию № 2:	25
Количество слов:	6 852
Число знаков:	51 745
Адреса пропущенные при проверке:	
Количество завершенных проверок:	36

>>

Самые длинные фрагменты, определенные, как подобные

№	Название, имя автора или адрес гиперссылки (Название базы данных)	Автор	Количество одинаковых слов
1	Данные этого документа не доступны		21
2	URL_ https://riptutorial.com/Download/php-ru.pdf		20
3	URL_ https://riptutorial.com/Download/php-ru.pdf		17
4	System informatyczny wspomagający rejestrowanie czasu pracy w przedsiębiorstwie Politechnika Lubelska (Politechnika Lubelska)	Patryk Drozd, Michał Derkacz	11
5	URL_ https://www.logicbig.com/how-to/code-snippets/jcode-java-servlet-extending-httpervlet.html		9
6	URL_ https://studfiles.net/preview/1702375/		8
7	URL_ https://riptutorial.com/Download/php-ru.pdf		8
8	URL_ https://riptutorial.com/Download/php-ru.pdf		8
9	URL_ https://github.com/blackav/ejudge/blob/master/NEWS.RUS		7
10	Данные этого документа не доступны		6

>>

Документы, в которых найдено подобные фрагменты: из



>>

Документы, содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

>>

Документы, содержащие подобные фрагменты: Из внешних баз данных

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобию № 2

№	Название (Название базы данных)	Автор	Количество одинаковых слов (количество фрагментов)
1	<i>Данные этого документа не доступны</i>		27 (2)
2	System informatyczny wspomagający rejestrowanie czasu pracy w przedsiębiorstwie <i>Politechnika Lubelska (Politechnika Lubelska)</i>	Patryk Drozd, Michał Derkacz	11 (1)
3	Internetowa aplikacja bazodanowa „Serwis komputerowy” <i>Państwowa Wyższa Szkoła Zawodowa im. Stanisława Pigońa w Krośnie (Instytut Politechniczny)</i>	Piotr Filak	6 (1)

>>

Документы, содержащие подобные фрагменты: Из интернета

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобию № 2

№	Источник гиперссылки	Количество одинаковых слов (количество фрагментов)
1	URL_ https://riptutorial.com/Download/php-ru.pdf	79 (9)
2	URL_ https://scienceforum.ru/2018/3227/3594	25 (5)
3	URL_ https://studfiles.net/preview/1702375/	18 (3)
4	URL_ https://github.com/blackav/ejudge/blob/master/NEWS.RUS	13 (2)
5	URL_ https://www.logicbig.com/how-to/code-snippets/jcode-java-servlet-extending-httpervlet.html	9 (1)

АНДАТПА

Бұл дипломдық жоба «техниканы сату бойынша Интернет-дүкен» тақырыбына арналған. Интернет-дүкен тауарлардың әртүрлі санаттарын ұсынады. Бұл дипломдық жобада MVC технология қолданылады. IDE интеграцияланған даму ортасы (Integrated Development Environment) IntelliJ Idea осы жобада қолданылады және Oracle компаниясы тегін тарататын Java тіліндегі қосымшалар әзірлеушісі жиынтығы қолданылады. Әзірлеу кезінде Java Enterprise Edition(java EE) корпоративтік қосымшаларды құру платформасы пайдаланылды. Front-end пайдаланушын jsp (JavaServer Pages) қолданылады. Jsp статистикалық және динамикалық мазмұнды жасауға мүмкіндік береді. Сондай-ақ, біздің веб-қосымшаның логикасына жауап беретін сервлеттер қолданылады. Бағдарламалық қамтамасыз екі бөліктен тұрады – серверлік және клиенттік.

АННОТАЦИЯ

Данный дипломный проект посвящен разработке Web-приложения «Интернет-магазин по продаже техники». В интернет магазине предоставлены различные категории товаров. В данном дипломном проекте поэтапно изложена разработка веб-приложения MVC. Применяется интегрированная среда разработки IDE (Integrated Development Environment) IntelliJ Idea и комплект разработчика приложений на языке Java бесплатно распространяемый компанией Oracle. При разработке использовался Java Enterprise Edition(java EE) платформа для создания корпоративных приложений. Для клиентской стороны пользовательского интерфейса front-end используется jsp (JavaServer Pages). Jsp позволяет разработчикам создавать статистический и динамический контент. Также используется так называемые сервлеты, которые отвечают за логику нашего веб-приложения. Сервлеты взаимодействуют с клиентской стороной с помощью запросов. Программное обеспечение состоит из двух частей – серверной и клиентской. Для работы базой данных используется система управления базой данных MySQL.

THE SUMMARY

This thesis is devoted to the topic « Internet shop for sales of technics ». The online store provides various categories of goods. This thesis work outlines the development of the MVC web application in stages. It uses the IDE (Integrated Development Environment) IntelliJ Idea integrated development environment and the Java application development kit distributed free by Oracle. The development was based on the Java Enterprise Edition (java EE) platform for creating enterprise applications. For the client side of the front-end user interface, use jsp (JavaServer Pages). Jsp allows developers to create statistical and dynamic content. Also used are the so-called servlets, which are responsible for the logic of our web application. Servlets interact with the client side using queries.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 Анализ и обзор современных веб-приложений	8
1.1 Виды веб-сервисов	9
1.2 Анализ существующих технологий создания веб - приложений	10
1.3 Постановка задач по разработке веб-приложения «Интернет-магазин по продаже техники»	15
2 Модели и службы функционирования веб-приложения «Интернет-магазин по продаже техники»	16
2.1 Функциональная модель веб-приложения «Интернет-магазин по продаже техники»	16
2.2 Обобщенный алгоритм функционирования веб-приложения «Интернет-магазин по продаже техники»	16
2.3 Информационное обеспечение веб-приложения «Интернет-магазин по продаже техники»	17
2.3.1 Обоснование выбора СУБД	17
2.3.2 Разработка структуры базы данных	20
3 Информационное и программное обеспечение	21
3.1 Структура программного обеспечения веб-приложения «Интернет-магазин по продаже техники»	21
3.2 Системное программное обеспечение	22
3.3 Инструментальное программное обеспечение	23
3.4 Прикладное программное обеспечение	25
3.5 Функциональное назначение	25
3.6 Работа и апробация веб-приложения «Интернет-магазин по продаже техники»	27
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	30
Приложение А	31
Приложение Б	36
Приложение В	41
Приложение Г	45

ВВЕДЕНИЕ

21 век – век информационных технологий. Во всем мире люди идут в ногу с новыми технологиями. Зачастую людям удобно и выгоднее покупать, приобретать, совершать какие-либо операции, не выходя из дома. Интернет расширил возможности людей до тех уровней, о которых человечество не могло вообразить. Современному человеку интернет предоставляет большие возможности в сфере развития (саморазвития), развлечения, а также в сфере бизнеса. Если раньше, человеку для того, чтобы сделать какие-либо покупки, приходилось тратить много времени на следующие действия, как дорога, очереди и т.п., то услуги, которые предоставляет глобальная сеть, дает людям свободу в действиях. В случае, бизнеса, это способ заработка денег, используя современные методы маркетинговых инструментов Google adWords и Яндекс.Директ. В современных условиях рентабельность и успешность бизнеса в значительной степени зависит от скорости, с которой фирма реагирует на изменения ситуации на рынке. Этот фактор в сочетании с производительностью становится стратегическим активом предприятий различных отраслей экономики. В этой связи применение передовых информационных технологий все чаще становится решающим фактором успеха, главным козырем в борьбе за лидерство.

Одна из таких возможностей – это интернет магазин, который предоставляет клиенту широкий ассортимент различных товаров. Интернет магазин впервые появился в США в 1994 г. Понятие интернет магазин внедрил американский предприниматель Джефф Безос, который является основателем интернет компании Amazon, самой крупной площадкой по продаже различных товаров и услуг. Позже в 1997 году по аналогии интернет магазина США появился интернет магазин в России. Самыми крупными интернет площадками в мире по продажам онлайн товаров на нынешний момент являются такие гиганты как Amazon, AliExpress, Walmart, eBay, брm, Carters. Сейчас люди во всем мире прибегают к услугам интернет магазина, потому что это удобно и эффективно. Если раньше интернет магазин использовался для продажи товаров, не имеющих срока годности, то сейчас через интернет магазины можно практически заказать абсолютно любую вещь. Установление эффективных взаимоотношений с клиентами и повышение конкурентоспособности во многом зависят от внедрения инновационных технологий

С появлением интернет магазина облегчилась жизнь многих людей, так как появилась возможность заказывать товары со всего мира. В этом заслуга программистов новаторов современного мира. В современных условиях эффективность и успешность интернет магазина в наибольшей степени зависит от изменения ситуации на рынке. Но дело не обходиться одним лишь предоставлением товаров, на самом деле идея интернет магазина более многогранна: она предусматривает создание максимально комфортных условий для конечного потребителя.

1 Анализ и обзор современных веб-приложений

Современное веб приложение должно обладать следующими характеристиками.

Безопасность. Считается одним из самых важных критериев для веб-приложения в современном мире. Никто из пользователей не будет использовать услуги веб-приложения, если ему не будет гарантировано безопасность его личных данных. Пользователь не должен беспокоиться, что его конфиденциальные данные получают посторонние люди. Существуют различные виды хакерских атак на веб-ресурсы, перечислим самые распространённые. Sql-инъекций это вид хакерской атаки, при которой через GET, POST запросы или значения cookie передается серверу sql код, который может изменить, вытащить или удалить какие-либо данные с базы данных. DoS атака используется для вывода из строя, какой либо программы или системы, после чего нужно будет перезагрузить программу или систему. Вирусы самые известные атаки для обычного пользователя. Суть заключается во внедрении вредоносной программы в ваш компьютер, которая потом поможет злоумышленнику украсть какие-то конфиденциальные данные. Хотя вирусы могут принести существенный урон для обычного пользователя, но с помощью антивирусов мы легко можем предотвратить различные атаки на наш персональный компьютер.

Масштабируемость. Не менее важный критерий для веб-приложения. Нагрузка на сервера в современном мире просто колоссальная, каждый второй человек имеет различные устройства (смартфон, ноутбук, планшет), которые дают ему доступ в глобальную сеть 24 часа в сутки 7 дней в неделю. Для того чтобы пользователи были довольны веб-ресурсом нужно позаботиться о такой характеристике, как масштабируемость, которая дает нам возможность увеличивать ресурсы как программного обеспечения, так и аппаратной части. Существует два вида масштабирования: горизонтальное и вертикальное. Горизонтальное масштабирование подразумевает увеличение количества так называемых хостов, которые улучшают производительность приложения. Вертикальное масштабирование увеличивает производительность веб-приложения за счет добавление таких ресурсов как память, процессор, оперативная память и т.д.

Гибкость. Веб-приложение должно быть гибким, чтобы при дополнении или изменении какой-либо части исходного кода структура приложения не поменялась. Также мы должны учитывать такой фактор, как развитие веб-технологий, чтобы разработчик всегда мог внедрить эти технологии в свой проект. Разработка подразумевает всегда командную работу, при которой гибкость проекта зависит от используемых инструментов веб разработчика.

Отказоустойчивость и высокий уровень доступности. Также немало важным критерием является отказоустойчивость и высокий уровень доступности веб-приложения. Веб-ресурс всегда должен быть доступным для пользователей вне зависимости от ошибок и времени.

Любое веб-приложение представляет собой набор статических и динамических веб-страниц. Статическая веб-страница — это страница, которая всегда отображается перед пользователем в неизменном виде. Веб-сервер отправляет страницу по запросу веб-браузера без каких-либо изменений. В противоположность этому, сервер вносит изменения в динамическую веб-страницу перед отправкой ее браузеру. По причине того что страница меняется, она называется динамической. Например, можно создать страницу, на которой будут отображены результаты программы оздоровления. При этом некоторая информация (например, имя сотрудника и его результаты) будет определяться в момент запроса страницы сотрудником. Статический веб-сайт содержит набор соответствующих HTML-страниц и файлов, размещенных на компьютере, на котором установлен веб-сервер.

1.1 Виды веб-сервисов

Существуют различные виды веб-сервисов. Каждый из них решает свою определенную задачу.

Корпоративный портал. Корпоративные порталы часто используют организации для оптимизации бизнес процесса. Сотрудники организации могут использовать портал для получения доступа к приложениям или нужным данным. Корпоративный портал можно ассоциировать с интранетом, как единую информационную площадку для всех сотрудников, которые могут взаимодействовать между собой. Преимуществ у корпоративных порталов очень много, пройдемся по некоторым. Сотрудники организации могут создавать различные форумы для коммуникации между собой, делиться различными новостями и материалами, также создавать различные файлы и документы, которые используются при выполнении важных задач. Повышение мобильности и работоспособности сотрудников.

CRM (Customer Relationship Management). Система облегчает взаимодействие с потребителями. Система управления клиентами собирает необходимые сведения о потребителях, историю взаимодействия с ними, для повышения показателей продаж и ускорение бизнес-процесса. Сбор информации о клиентах, партнерах, поставщиков и последующий анализ данных помогают бизнесу функционировать в наиболее эффективном режиме.

Можно выделить основные принципы CRM систем: оптимизация бизнес процессов, единое хранилище данных, использование таких каналов как звонки, съезды, социальные сети, почту и т.п. Целью любой CRM системы является повышение увлеченности потребителя за счет собранных данных о клиенте.

ERP системы. Планирование ресурсов предприятия. Данное программное обеспечение помогает максимально эффективно использовать ресурсы предприятия. ERP система помогает синхронизировать и управлять процессами в предприятии, а также улучшает взаимодействие между отделами компаний и экономит трудовые ресурсы.

ITRP (IT Resources Planning) — класс Web-приложений, предназначен для поддержки управления корпоративными ИТ-ресурсами и сервисами.

OSS (Operation Support Systems) — вид Web-приложений, предназначен для обеспечения работы операторов распределенных вычислительных сетей. OSS обеспечивает управление сетью, производительность, ликвидацию сбоев в работе, создание и учет сервисов, планирование сетевых ресурсов, мониторинг процессов, контроль за безопасностью, качество услуг и уровень обслуживания клиентов, сбором статистических данных. Разновидностью OSS является система поддержки бизнеса — BSS (Business Support Systems). К ним относятся биллинговые системы, системы управления взаимоотношениями с клиентами, управления сетями, заказами, качеством услуг.

Специализированные веб-сайты. Эти сайты имеют свою тематику и целевую аудиторию. К специализированным веб-сайтам можно отнести: блог, интернет-магазин, социальную сеть, форум, игровые сайты и т.п. Социальные сети используются для общения с людьми в интернете. Также каждый пользователь может создавать профиль с личной информацией о себе. Выкладывать различные фотографии, видео и новости. Социальные сети используются как для бизнеса, так и для развлечений.

Интернет магазины облегчили жизнь многих людей. Если раньше человеку приходилось тратить много времени и усилий на выполнение, каких-либо действий, то с появлением интернет магазинов возможности человека расширились. Клиентам предоставляется широкий ассортимент товаров, которые они могут получить практически в любой точке мира.

1.2 Анализ существующих технологий создания веб – приложений

Для разработки своего веб-приложения, я использовал высокоуровневый язык программирования Java. Язык java является самым популярным языком в сфере разработки кроссплатформенных корпоративных приложений. Он является строго типизированным языком, это дает возможность в дальнейшем избежать ошибок, связанных с типом данных, в отличие от таких языков как php, которые не являются строго типизированными языками. В качестве среды разработки я использовал intelliJ idea, которая является самой популярной средой разработки для языка java. Эта среда позволяет максимально комфортно писать код для вашего веб-приложения. Также в ней есть так называемые горячие клавиши, которые позволяют максимально облегчить процесс разработки.

В ходе разработки программист не раз встречается с рядом ошибок, но intelliJ idea помогает обнаруживать эти ошибки и устранять их на начальном этапе без последствий. Для этого есть специальная программа-отладчик под названием debug, которая позволяет останавливать код в том месте, где вы хотите проверить корректность выполнения программы. Если раньше при разработке приложения, программисту приходилось постоянно сохранять исходный код, потом через командную строку вызывать компилятор для

компилирования исходного кода, то среда разработки `intellig idea` избавляет от ненужных действий. Эта среда позволяет структурировать ваш проект для вашего дальнейшего удобства, при дополнении или изменении в исходном коде. Также среда позволяет использовать такой мощный инструмент разработки как `GIT`, без этого инструмента вы не сможете совместно реализовать проект в команде с другими разработчиками. Программа `Git` позволяет вам создать так называемую новую ветку, которая дает вам возможность изменять и дополнять исходный код без опасения изменения структуры вашего проекта. И вы всегда сможете, увидеть какое именно разработчик внес дополнение или как изменил исходный код. У вас появляется возможность слияния нескольких проектов и выявления наилучшего исходного кода, для написания в вашем проекте. Также вы можете клонировать любой проект с помощью командной строки и команды `git clone`.

В качестве веб-сервиса используется `apache tomcat`. Он является контейнером для сервлетов. Он содержит множество библиотек, которые нужны для конфигурации нашего веб-приложения. В моем проекте я использую версию `Tomcat 9.0.10`, которая является наиболее популярной из предложенных версий. Так как я не использую фреймворк, то мне приходится вручную загружать в проект веб-сервис `Tomcat`.

Скриптовые языки. Большинство современных Web-браузеров способно интерпретировать код на скриптовых языках, таких как `VBScript` и `JavaScript`. Код на этих языках внедряется в Web-страницу и интерпретируется браузером. Типичный пример применения скриптовых языков — проверка корректности данных, вводимых пользователем в соответствующие поля HTML-формы, непосредственно в процессе ввода или после него, без обращения к Web-серверу. Подобные примеры применения скриптовых языков можно обнаружить при заполнении некоторых анкет и получении сообщений о том, что не заполнены обязательные поля (справедливости ради отметим, что далеко не все анкеты реализованы подобным образом).

Java-апплеты. Практически все современные браузеры способны отображать и выполнять Java-апплеты — специальные Java-приложения, которые пользователь получает в составе Web-страницы. Эти приложения нередко включаются в состав Web-страниц с целью добавления функциональности, которую сложно или невозможно реализовать с помощью скриптовых языков. Апплеты могут выполняться на всех платформах, для которых доступна виртуальная Java-машина.

Апплеты обычно создаются в соответствии с правилами, оговаривающими период их жизни и способы взаимодействия со своим окружением. Однако апплет способен считывать значения параметров (например, цвета, шрифтов, файлов с графическими изображениями, используемыми при выполнении апплета) с содержащей его Web-страницы и в соответствии с этими параметрами изменять свое поведение. Кроме того, параметры апплета можно менять динамически из кода на скриптовых языках, содержащихся в составе той же страницы. Отметим, что, поскольку апплеты

реализуют выполнение кода на компьютере клиента, они в определенной степени являются потенциально опасным содержимым. Именно поэтому все современные браузеры обладают доступными пользователю средствами ограничения возможностей выполнения апплетов.

Приложения Macromedia Flash являются сегодня наиболее популярным расширением функциональности Web-браузеров — с их помощью многие Web-дизайнеры придают своим сайтам интерактивность и оригинальность. Модель безопасности приложений Flash основана на том, что Macromedia Flash Player, как и виртуальная Java-машина, выполняет приложения в ограниченном адресном пространстве, при этом выполняемые приложения не имеют доступа к файловой системе (кроме одного конкретного каталога, используемого Macromedia Flash Player для служебных целей) и другим ресурсам компьютера пользователя; исключение делается для микрофонов и видеокамер, однако пользователь должен дать разрешение на передачу данных, полученных с этих устройств. Доступ к сетевым ресурсам ограничивается доменом, с которого было получено приложение. Отметим, что приложения Flash также могут управляться с помощью кода JavaScript, присутствующего на той же странице. Сам Macromedia Flash Player для Microsoft Internet Explorer является элементом управления ActiveX и использует возможности элементов управления ActiveX для доступа к свойствам приложений Flash из скриптовых языков.

Отметим, что помимо вышеперечисленных наиболее популярных средств расширения функциональности браузеров имеется и ряд других средств, реализованных обычно в виде так называемых модулей расширения (plug-in). Поскольку модули расширения также представляют собой исполняемый код, современные браузеры обладают средствами ограничения возможностей, связанных с их загрузкой и выполнением.

Технологии создания серверных частей Web-приложений. CGI Common Gateway Interface — это стандартный интерфейс, позволяющий выполнять серверные приложения, вызываемые через URL. Входной информацией для таких приложений служит содержимое HTTP-заголовка либо тело запроса, в зависимости от применяемого протокола. CGI-приложения генерируют HTML-код, который возвращается браузеру. Отметим, что в свое время широко использовался и термин «CGI-скрипт», происхождение которого объясняется тем, что подобные приложения писались на скриптовых языках типа Perl, выполняющихся, тем не менее, не в браузере, а на сервере. CGI-приложения можно создавать с помощью практически любого средства разработки, генерирующего консольные приложения для операционной системы, под управлением которой функционирует Web-сервер.

Основная проблема всех CGI-приложений заключается в том, что при каждом клиентском запросе сервер загружает это приложение в отдельное адресное пространство, а затем инициирует его выполнение и выгрузку. Эта особенность ограничивает производительность приложений и возможность одновременной обработки большого количества клиентских запросов. ISAPI и Apache DSO. Проблему ограниченной производительности Web-приложений,

которые выполняются в отдельном адресном пространстве, можно решить, создав приложение в виде библиотеки, загружающейся в адресное пространство Web-сервера и при необходимости остающейся там для обработки последующих запросов от других клиентов; естественно, в этом случае Web-сервер должен поддерживать загрузку таких библиотек. Подобные приложения для Microsoft Internet Information Service носят название ISAPI (Internet Server Application Program Interface), а для весьма популярного Web-сервера Apache такие библиотеки называются Apache DSO (Dynamic Shared Objects). Отметим, однако, что при создании как CGI-, так и ISAPI-приложений было довольно сложно отделить задачи Web-дизайна от задач, связанных с реализацией функциональности и логики приложений, подобные приложения генерируют Web-страницы целиком, поэтому все данные, связанные с дизайном этих страниц, должны в общем случае содержаться внутри исполняемого файла.

ASP, JSP, PHP. Очередной шаг в развитии технологий создания Интернет-приложений — появление средств, позволяющих отделить задачи Web-дизайна от задач, связанных с реализацией функциональности приложений. Первой из таких технологий стала Active Server Pages (ASP), построенная на основе ISAPI-фильтра. Основная идея ASP заключается в создании Web-страниц с внедренными в них фрагментами кода на скриптовых языках. Однако, в отличие от рассмотренных выше средств применения скриптовых языков для расширения функциональности браузеров, указанные фрагменты кода интерпретируются не браузером, а сервером (точнее, предназначенной для этого ISAPI-библиотекой), и результат выполнения этих фрагментов кода замещает сам фрагмент кода в той версии страницы, которая передается в пользовательский браузер. Вскоре после ASP появились и другие технологии, реализующие идею размещения внутри Web-страницы кода, выполняемого Web-сервером. Наиболее известной из них сегодня является технология JSP (Java Server Pages), основная идея которой — однократная компиляция Java-кода (сервлета) при первом обращении к нему, выполнение методов этого сервлета и помещение результатов выполнения этих методов в набор данных, отправляемых в браузер. Еще одной популярной технологией подобного типа является PHP (Personal Home Pages), которая использует CGI-приложения, интерпретирующие внедренный в HTML-страницу код на скриптовом языке. Новейшей версией технологии Active Server Pages является ASP.NET, ключевая в архитектуре Microsoft.NET Framework. Основное отличие этой технологии от ASP с точки зрения архитектуры приложений заключается в том, что код, присутствующий на Web-странице, не интерпретируется, а компилируется и кэшируется, что, естественно, способствует повышению производительности приложений. С помощью ASP.NET можно создавать Web-приложения и Web-сервисы, которые не только позволяют реализовать динамическую генерацию HTML-страниц, но и интегрируются с серверными компонентами и могут использоваться для решения широкого круга бизнес-задач, возникающих перед разработчиками современных Web-приложений.

В общем случае клиентом Web-сервера может быть не только персональный компьютер, оснащенный обычными Web-клиентами (например, Web-браузером), но и мобильные устройства, отличающиеся ограниченным размером экрана, малым объемом памяти, а нередко и невозможностью отображения графики. Для этих устройств существуют свои протоколы передачи данных (Wireless Access Protocol, WAP) и соответствующие языки разметки (WML, Wireless MarkupLanguage, СHTML, Compact HTML и т.п.). При этом необходимо передавать данные на мобильное устройство в соответствующем формате, для чего нередко создаются специальные сайты (например, поддерживающие WAP и WML). Более удобным представляется создание приложений, которые способны генерировать тот или иной код в зависимости от типа клиента. Именно такой подход и реализован в Microsoft ASP.NET.

С ростом объема используемых данных и числа посетителей Web-сайтов возрастают требования к надежности, производительности и масштабируемости Web-приложений. Для удовлетворения этим требованиям бизнес-логика, реализованная в Web-приложении, а также сервисы обработки данных и реализации транзакций, отделяются от интерфейса приложений и переносятся на сервер приложений в виде бизнес-объектов. Серверы приложений и соответствующие бизнес-объекты могут быть различного типа (наиболее распространенными из них сегодня являются серверы, поддерживающие спецификацию Java2 Enterprise Edition, и серверы, базирующиеся на технологиях COM и Microsoft.NET). Впрочем, рассмотрение серверов приложений выходит за рамки данной статьи. Отметим, что бизнес-объекты часто предоставляют доступ к данным корпоративных информационных систем либо реализуют какую-либо часть их функциональности, осуществляя функции интеграции Web-приложений с другими приложениями, применяемыми на предприятии.

Web-сервисы. Говоря о серверных Web-технологиях, нельзя обойти вниманием такую важную, как Web-сервисы XML. На Web-сервисы XML в настоящее время нередко возлагается решение многих задач, связанных с интеграцией приложений, в том числе созданных на разных платформах. Создавать Web-сервисы можно и в виде исполняемых файлов, и в виде библиотек, и в виде интерпретируемого кода; существуют также средства представления бизнес объектов в виде Web-сервисов. Методы Web-сервисов можно вызывать из обычных приложений, Web-приложений и других Web-сервисов, и, за редким исключением, конечные пользователи непосредственно с Web-сервисами дела не имеют. Тем не менее, в последнее время отмечается массовое появление приложений, использующих Web-сервисы, в том числе и приложений, предназначенных для конечных пользователей. В языке XML все имена должны начинаться с буквы, символа подчёркивания () или двоеточия (:) и продолжаться только допустимыми для имён символами, а именно они могут содержать только буквы, входящие в секцию букв кодировки Unicode, арабские цифры, дефисы, знаки подчёркивания, точки и двоеточия. Однако

имена не могут начинаться со строки `xml` в любом регистре. Имена, начинающиеся с этих символов, зарезервированы для использования консорциумом W3C. Нужно помнить, что так как буквы не ограничены исключительно символами ASCII, то в именах можно использовать слова из родного языка.

JPA – это технология, обеспечивающая объектно-реляционное отображение простых JAVA объектов и предоставляющая API для сохранения, получения и управления такими объектами. JPA – это спецификация (документ, утвержденный как стандарт, описывающий все аспекты технологии), часть EJB3 спецификации. Сам JPA не умеет ни сохранять, ни управлять объектами.

1.3 Постановка задач по разработке веб-приложения «Интернет-магазин по продаже техники»

Основной задачей дипломного проекта является разработка веб-приложения «Интернет-магазин по продаже техники».

В данной дипломном проекте используются последние технологии в сфере разработки веб-приложения, которые позволяют создать масштабируемое, безопасное и гибкое приложения с минимальными затратами.

Требования при разработке веб-приложения «Интернет-магазин по продаже техники»:

- приложение должно обладать удобным и понятным интерфейсом для пользователей;
- исходный код приложения должен комментироваться на этапе разработки для будущих изменений или дополнений;
- приложение должно предусматривать контроль вводимой информации;
- для удобства и контроля версии должен использоваться git;
- приложение должно включать хранение и обработку данных о различных товарах;
- на веб-ресурсе должна быть информация о владельце данного сайта;
- в приложении должна присутствовать административная часть, где происходит контроль за всеми происходящими действиями в приложения;
- приложение должно сообщать пользователю о некорректных действиях при работе с сервисом, с выдачей сообщений об ошибках;
- приложение должно предоставлять полную информацию пользователю о товаре, фото с разных углов, срок годности, стоимость и способ доставки;
- приложение должно обладать возможностью удаления профиля;
- приложение должно иметь окна регистрации и авторизации пользователя, для того чтобы пользователь мог добавлять в корзину нужные ему товары.

2 Модели и службы функционирования веб-приложения «Интернет-магазин по продаже техники»

2.1 Функциональная модель веб-приложения «Интернет-магазин по продаже техники»

Функциональная модель веб-приложения «Интернет-магазин по продаже техники» представляет собой взаимодействие двух модулей – администратора и пользователя, приложение включает в себя следующие функции:

Создание и редактирование учетных записей. Функция позволяет добавлять и удалять учетные записи клиентов. Для ввода данных необходимо внести адрес электронной почты пользователя, который будет использоваться в дальнейшем как логин учетной записи, и временный пароль, который пользователю будет необходимо изменить при первой авторизации.

Добавление товаров в базу данных, для дальнейшего отображения клиенту. Изменения цены и количества товаров. Дополнение новых категорий товаров, удаление ненужных товаров и т.п.

2.2 Обобщенный алгоритм функционирования системы

Целью разработки веб-приложения «Интернет-магазин по продаже техники» является создание веб-ресурса для продажи различных товаров, своевременное информирование пользователей о скидках и новых поступлениях, новостные данные в любом формате. Структура веб-приложения «Интернет-магазин по продаже техники» предоставлена на рисунке Г.1.

Разработка состоит из двух модулей:

- модуль «Администратор»;
- модуль «Пользователь».

В модуль «Администратор» входят и другие подмодули:

- аутентификация пользователя – проверка на существование пользователя в базе данных;
- авторизация пользователя – проверка прав пользователя на доступ и выявление у него ролей;
- добавление пользователя – регистрация пользователей с различными ролями (Администратор или Пользователь);
- подмодуль добавление товаров – позволяет добавлять новые товары или изменять описание существующих товаров;
- подмодуль добавление категории – подмодуль для добавления новых категорий или изменения уже существующих;
- подмодуль обработка заказов – подмодуль для обработки определенного заказа клиента.

Модуль «Клиент» включает в себя следующие подмодули:

- аутентификация клиента – проверка на существование пользователя в базе данных;

- подмодуль корзина – для добавления товаров в корзину пользователя;
- подмодуль просмотр товаров – позволяет просмотреть список товаров, которые предоставлены на сайте;
- подмодуль панель инструментов – для удобной навигации пользователя по сайту;
- заказ товаров – пользователь может заказать любой товар в ассортименте;
- подмодуль детали – позволяет узнать всю необходимую информацию о конкретном товаре.

2.3 Информационное обеспечение веб-приложения «Интернет-магазин по продаже техники»

2.3.1 Обоснование выбора СУБД

На сегодняшний день есть множество СУБД, которые могут обрабатывать большие объемы данных.

Классификация БД происходит по следующим критериям:

- сфера использования;
- инфраструктура;
- характеристики обработки;
- количество полученных данных;
- уровни классифицирования хранящихся данных;
- качества исследуемых объектов.

Работать с БД без СУБД, которые могут преобразовать большие объемы данных в поддающуюся влиянию информацию, было бы невозможным. Чтобы обработать данные больших объемов, сотруднику понадобится много времени, помимо этого могут возникнуть проблемы с внесением данных, так как сотрудник может забыть какую-то важную информацию. [1].

Правильная организация базы данных обеспечивает более быстрые и гибкие возможности выборки данных. Она существенно упрощает реализацию средств поиска и сортировки, а проблемы прав доступа к информации решаются при помощи средств контроля за привилегиями, присутствующими во многих системах управления базами данных (СУБД). Кроме того, упрощаются процессы репликации и архивации данных.

Язык SQL заложен в основу практически всех существующих СУБД. Чтобы перейти к рассмотрению примеров работы с базами данных, необходимо хотя бы в общих чертах представлять, как работает SQL.

SQL обычно описывается как стандартный язык, используемый для взаимодействия с реляционными базами данных. Однако SQL не является языком программирования, как C, C++ или PHP. Скорее, это интерфейсное средство для выполнения различных операций с базами данных, предоставляющее в распоряжение пользователя стандартный набор команд. У MySQL есть внутренняя структура – ассортимент движков и внешняя - sql.

MySQL содержит следующие модульные системы:

- серверная инициализация;
- менеджер соединений;
- потоковый менеджер;
- обработчик команд;
- аутентификация;
- парсинг;
- кеширование;
- оптимизация;
- графический менеджер;
- ассортимент движков;
- логистика;
- дубликация;
- API и его ядро.

Алгоритм выполнения действий модульных систем такой.

В первую очередь начинает работу первая модульная система. Она начинает работу глобальных слоев, выделяет определенные объемы памяти, считывает конфигурационные пакеты данных и параметры командных строк и транслирует контроль над всей работой менеджеру соединения [2].

Когда клиент отправляет свой запрос, к работе присоединяется потоковый менеджер, который разрабатывает поток индивидуально для клиента, далее происходит процесс аутентификации клиента.

Далее подходит очередь парсингового модуля, затем происходит процесс кеширования.

После информация может отправиться в модульную систему оптимизации или дубликации. В итоге вся информация обратно отправляется клиенту.

В реляционных СУБД данные организуются в виде набора взаимосвязанных таблиц. Связи между таблицами реализуются в виде ссылок на данные других таблиц. Таблицу можно представить, как двухмерный массив, в котором расположение каждого элемента характеризуется определенными значениями строки и столбца.

Как выполняются операции с реляционной базой данных? Для этого в SQL есть специальный набор часто используемых команд, таких как SELECT, INSERT, UPDATE и DELETE. Например, если вам нужно получить адрес электронной почты клиента, используя ID 2003, выполните следующую команду SQL:

```
SELECT email FROM customers WHERE id = 2003
```

В итоге, команда выглядит так:

```
SELECT имя поля FROM имя таблицы [WHERE условие]
```

Скобки указывают, что последняя часть команды является необязательной. Например, чтобы получить адреса электронной почты всех клиентов из таблицы клиентов, выполните следующий запрос:

```
SELECT email FROM customers
```

Например, предположим, что пользователь добавляет новую запись в таблицу продуктов. Самая простая команда вставки:

```
INSERT into products VALUES (100, Фззду);
```

Если вам нужно удалить эти данные позже, используйте следующую команду:

```
DELETE FROM products WHERE prod_id = 100
```

Поддержка баз данных в Java включает совместимость со всеми известными базами данных (такими как Oracle) и многими базами данных, которые вы не слышали. Базы данных могут поддерживаться Java, если пользователям необходимо использовать строгую СУБД для хранения распределенной информации в Интернете. Поддержка базы данных в Java представлена набором стандартных функций, которые позволяют подключаться к базе данных, обрабатывать запросы и отключаться.

MySQL MySQL (надежная СУБД на основе SQL, разработанная и обновленная TS.X DataKonsultAB (Стокгольм, Швеция)) С 1995 года MySQL стала одной из самых популярных СУБД в мире. Его политика надежности и гибкости [3].

Обладая отличными функциями и широким спектром простых в использовании стандартных функций интерфейса, MySQL стал наиболее распространенным инструментом для использования баз данных в Java.

MySQL распространяется в соответствии с условиями GNU General License (GPL). Для получения дополнительной информации о текущей стратегии лицензирования MySQL можно посмотреть на веб-сайте MySQL.

Специальные функции JDBC доступны для определенных типов СУБД. Но что, если пользователю необходимо подключиться к MySQL, Microsoft SQL Server и IBM DB2 в одном приложении? Аналогичные проблемы возникают при разработке приложений, которые не должны полагаться на СУБД. Эти приложения применяются к существующей инфраструктуре базы данных клиентов. JDBC - это платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета java.sql, входящего в состав Java SE.

Если мы хотим подключиться к базе данных нужно использовать JDBC. Стандарт JDBC, разработанный Sun Microsystems, стал открытым стандартом. Поддержка JDBC интегрирована с дистрибутивом Java. В редких случаях определенные настройки могут быть редкими или даже нулевыми [4].

Язык программирования Java очень прост при разработке интернет приложений как на стороне клиента, так и на стороне сервера. Этот язык имеет много интересных функций.

Веб-приложение использует phpMyAdmin для реализации взаимодействия пользователя с базой данных MySQL. phpMyAdmin - это веб-приложение с открытым исходным кодом, написанное на PHP, веб-интерфейс для управления базами данных MySQL. Используя PHPMyAdmin, вы можете управлять своим сервером MySQL через браузер, выполнять команды SQL и отображать содержимое таблиц и баз данных. Это приложение предоставляет

очень распространенный и простой в использовании интерфейс для веб-разработчиков, поскольку оно управляет СУБД MySQL без прямого доступа к командам SQL [5].

PHPMyAdmin широко используется сегодня. При разработке своих продуктов MySQL учла все нововведения в СУБД. Большинство Казахстанских поставщиков используют это приложение в качестве панели управления, чтобы предоставить клиентам возможность управлять своими собственными базами данных собственности. Нужно отметить, что практически все хостеры устанавливают именно PHPMyAdmin в свои панели управления для работы с базой данных. Это приложение является одним из самых популярных среди других приложений в этом классе.

2.3.2 Разработка структуры базы данных

База данных разработана в СУБД MySQL и называется internet_shop в ней имеются 4 таблицы.

Ниже представлены разработанные структуры таблиц базы данных:

- таблица 1 – данные о пользователях;
- таблица 2 – данные о товарах;
- таблица 3 – категории товаров;
- таблица 4 – корзина пользователя;

Таблица 1 – users

Имя поля	Тип поля	Описание
id	Int(11)	Идентификатор пользователя
email	Varchar(200)	Логин
password	Varchar(200)	Пароль
full_name	Varchar(200)	Полное имя пользователя
role	int(1)	Роли

Таблица 2 – items

Имя поля	Тип поля	Описание
id	Int(11)	Идентификатор товара
name	Int(11)	Название товара
category_id	Int(11)	Категория товара
price	Int(10)	Цена
amount	Int(10)	Количество товаров
added_date	timestamp	Дата добавления

Таблица 3 – categories

Имя поля	Тип поля	Описание
----------	----------	----------

id	Int(11)	Идентификатор категории
name	Varchar(200)	Название категории

Таблица 4 – baskets

Имя поля	Тип поля	Описание
id	Int(6)	Идентификатор корзины
item_id	Int(6)	Идентификатор товара
user_id	Int(6)	Идентификатор пользователя

На рисунке Г.2 представлена схема разработанной базы данных

3 Информационное и программное обеспечение

3.1 Структура программного обеспечения веб-приложения «Интернет-магазин по продаже техники»

Структура программного обеспечения веб-приложения «Интернет-магазин по продаже техники» представлена на рисунке Г.3. Структура ПО делится на три части: системное программное обеспечение, инструментальное программное обеспечение, прикладное программное обеспечение.

Системное программное обеспечение. При разработке веб-приложения «Интернет-магазин для продажи техники» использовалась операционная система windows 7 и сервисные программы.

Инструментальное программное обеспечение. При разработке веб-приложения «Интернет-магазин для продажи техники» использовались высокоуровневый язык программирования java и javax.servlet-api-3.1.0.jar для использования сервлетов, которые отвечают за логику нашего веб приложения (принимают POST и GET запросы). Также мы использовали jsp-api-2.2.jar для статического и динамического отображения данных. Commons-codec-1.10.jar содержит набор библиотек утилит Java самого разного назначения. Утилиты проекта Apache Commons лежат в основе таких проектов как Apache Tomcat, Struts, Hibernate и др.

Прикладное программное обеспечение. При разработке веб-приложения «Интернет-магазин для продажи техники» использовались такие программы, как intellij idea, git, insomnia, СУБД MySQL. IntelliJ idea интегрированная среда разработки для написания исходного кода на языке java. Эта среда является самой популярной средой для написания кода на языке java. Git это распределенная система управления версиями. Это значит, что у каждого участника проекта есть своя копия репозитория, которая находится в папке “.git”, которая расположена в корне проекта. Именно в этой папке хранятся все коммиты и другие объекты Git. Когда вы работаете с Git, он в свою очередь работает с этой папкой. Insomnia даёт возможность разработчику проверить работоспособность приложения посылая REST запросы. Insomnia распространяется бесплатно.

3.2 Системное программное обеспечение

Операционная система Microsoft Windows 7. Операционная система (ОС) является основной и необходимой составляющей компьютера, без которой компьютер не может работать в принципе. Это комплекс программ, обеспечивающих взаимодействие всех аппаратных и программных частей компьютера между собой и взаимодействие пользователя и компьютера. ОС обеспечивает связь между пользователем, программами и аппаратными устройствами, обеспечивает совместное функционирование всех устройств компьютера и предоставляет пользователю доступ к его ресурсам. ОС разные, но их назначение и функции одинаковые. ОС принимает на себя сигналы-команды, которые посылают другие программы, и «переводит» их на понятный машине язык. ОС управляет всеми подключенными к компьютеру устройствами, обеспечивая доступ к ним другим программам. Каждая ОС состоит как минимум из трех обязательных частей: ядро, драйверы, интерфейс. Ядро - переводит команды с языка программ на язык «машинных кодов», понятный компьютеру. Драйверы - программы, руководители устройствами. Интерфейс - оболочка, с помощью которой пользователь общается с компьютером. В состав операционной системы входит специальная программа - командный процессор, которая запрашивает у пользователя команды и выполняет их. Пользователь может дать, например, команду выполнения какой-либо операции над файлами (копирование, удаление, переименование), команду вывода документа на печать и т.д.

Для упрощения работы пользователя в состав современных операционных систем, и в частности в состав Windows, входят программные модули, создающие графический пользовательский интерфейс. В операционных системах с графическим интерфейсом пользователь может вводить команды с помощью мыши, тогда как в режиме командной строки необходимо вводить команды с клавиатуры. Операционная система содержит также сервисные программы, или утилиты. Такие программы позволяют обслуживать диски (проверять, сжимать, дефрагментировать и т. д.), выполнять операции с файлами (архивировать и т. д.), работать в компьютерных сетях и т.д. Для удобства пользователя в операционной системе обычно используется справочная система. Она предназначена для оперативного получения необходимой информации о функционировании как операционной системы в целом, так и о работе ее отдельных модулей. При разработке приложения использовалась операционная система Windows 7. Windows 7 является самой популярной операционной системой в мире с 2016 года. Эта операционная система поддерживает Unicode 5.1. В Windows 7 была также улучшена совместимость со старыми приложениями, некоторые из которых было невозможно запустить на Windows Vista. Также в Windows 7 появился режим Windows XP Mode, позволяющий запускать старые приложения в виртуальной машине Windows XP, что обеспечивает практически полную поддержку старых приложений.

3.3 Инструментальное программное обеспечение

Для программной реализации был выбран высокоуровневый язык программирования java. Для того чтобы использовать этот язык программирования нужно его скачать с официального сайта oracle.com. Так называемый jdk комплект разработчика, вместе с ним поставляется компилятор java и JVM виртуальная машина java, с помощью JVM запускаются все java программы. Также в комплект входят стандартные библиотеки языка java.

На сегодняшний момент язык Java является одним из самых распространенных и популярных языков программирования. Первая версия языка появилась еще в 1996 году в недрах компании Sun Microsystems, впоследствии поглощенной компанией Oracle. Java задумывался как универсальный язык программирования, который можно применять для различного рода задач. И к настоящему времени язык Java проделал большой путь, было издано множество различных версий. Текущей версией является Java 12, которая вышла в марте 2019 года. А Java превратилась из просто универсального языка в целую платформу и экосистему, которая объединяет различные технологии, используемые для целого ряда задач: от создания десктопных приложений до написания крупных веб-порталов и сервисов. Кроме того, язык Java активно применяется для создания программного обеспечения для целого ряда устройств: обычных ПК, планшетов, смартфонов и мобильных телефонов и даже бытовой техники. Достаточно вспомнить популярность мобильной ОС Android, большинство программ для которой пишутся именно на Java. Для написания мобильных приложений используется android studio.

Особенности Java. Ключевой особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы. А затем этот байт-код выполняется виртуальной машиной JVM (Java Virtual Machine). В этом плане Java отличается от стандартных интерпретируемых языков как PHP или Perl, код которых сразу же выполняется интерпретатором. В то же время Java не является и чисто компилируемым языком, как C или C++. Подобная архитектура обеспечивает кроссплатформенность и аппаратную переносимость программ на Java, благодаря чему подобные программы без перекомпиляции могут выполняться на различных платформах - Windows, Linux, Mac OS и т.д. Для каждой из платформ может быть своя реализация виртуальной машины JVM, но каждая из них может выполнять один и тот же код.

Java является языком с Си-подобным синтаксисом и близок в этом отношении к C/C++ и C#. Еще одной ключевой особенностью Java является то, что она поддерживает автоматическую сборку мусора. А это значит, что вам не надо освобождать вручную память от ранее использовавшихся объектов, как в C++, так как сборщик мусора это сделает автоматически за вас. Java является объектно-ориентированным языком. Он поддерживает полиморфизм, наследование, статическую типизацию. Объектно-ориентированный подход

позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений.

Установка Java. Для работы программ на языке Java на целевой машине должна быть установлена JRE (Java Runtime Environment). JRE представляет минимальную реализацию виртуальной машины, а также библиотеку классов. Поэтому, если мы хотим запускать программы, то нам надо установить JRE. Для каждой конкретной платформы имеется своя версия JRE. Однако, так как мы собираемся не только запускать программы, но и разрабатывать их, нам потребуется специальный комплект для разработки JDK (Java Development Kit). Apache tomcat, является веб-контейнером для сервлетов и jsp файлов. Так как у нас веб-приложение мы используем java EE, это мощный инструмент для создания клиент-серверных приложений.

Если вам нужно произвести какие-то операции с наборами объектов, вы не разрабатываете для этого свою собственную хеш-таблицу; вы используете API коллекций (интерфейс программирования приложений). Точно так же, если вам требуется простое веб-приложение или безопасная, интероперабельная распределенная прикладная система, оперирующая транзакциями, вы не захотите разрабатывать низкоуровневые API-интерфейсы, а будете использовать платформу Java EE. Так же как платформа Java Standard Edition (Java SE) предоставляет API для работы с коллекциями, Java EE предоставляет стандартный способ работы с транзакциями через Java API для транзакций (JTA), с сообщениями через службу сообщений Java (JMS) и с сохраняемостью через интерфейс JPA. Java EE располагает рядом спецификаций, предназначенных для корпоративных приложений. Она может рассматриваться как продолжение платформы Java SE для более удобной разработки распределенных, надежных, мощных и высокодоступных приложений.

Версия Java EE 7 стала важной вехой в развитии платформы. Она не только продолжает традиции Java EE 6, предлагая более простую модель разработки, но и добавляет свежие спецификации, обогащая наработанный функционал новыми возможностями. Кроме того, контекст и внедрение зависимостей (CDI) становится точкой интеграции между всеми новыми спецификациями. Релиз Java EE 7 почти совпадает с 13-й годовщиной выпуска корпоративной платформы. Она объединяет преимущества языка Java и опыт последних 13 лет. Java EE выигрывает как за счет динамизма сообществ свободных разработчиков, так и за счет строгой стандартизации группы Java Community Process (JCP). На сегодняшний день Java EE — это хорошо документированная платформа с опытными разработчиками, большим сообществом пользователей и множеством разворачиваемых приложений, работающих на серверах компаний. Java EE объединяет несколько интерфейсов API, которые могут использоваться для построения стандартных компонентно-ориентированных многозвенных приложений. Эти компоненты разворачиваются в различных контейнерах, предлагая серию служб. Архитектура Java EE состоит из набора спецификаций, реализуемых различными контейнерами. Контейнерами называются средства среды времени выполнения Java EE,

предоставляющие размещенным на них компонентам определенные службы, например, управление жизненным циклом разработки, внедрение зависимости, параллельный доступ и т. д. Такие компоненты используют точно определенные контракты для сообщения с инфраструктурой Java EE и с другими компонентами. Перед развертыванием они должны упаковываться стандартным способом (повторяя структуру определенного каталога, который может быть сжат в архивный файл). Java EE представляет собой расширенный набор функций платформы Java SE, что означает, что API-интерфейсы Java SE могут использоваться любыми компонентами Java EE.

3.4 Прикладное программное обеспечение

Для своего веб приложения я использую следующие прикладные программы: IntelliJ IDEA, Git, MySQL, Insomnia.

IntelliJ IDEA – это IDE (интегрированная среда разработки) для написания исходного кода на языке Java. Этот редактор является самым популярным инструментом для написания кода на языке Java. Существуют две версии IntelliJ IDEA: это Community и Ultimate. Мы используем версию Ultimate. Скачать IntelliJ IDEA можно с официального сайта [jetBrains.com](https://www.jetbrains.com/idea/). Хотя версия Ultimate является платной, но компания JetBrains предоставляет бесплатное годовое использование для студентов. Уникальные возможности IntelliJ IDEA избавляют программиста от груза рутинной работы, помогают своевременно устранить ошибки и повысить качество кода, поднимая продуктивность разработчика на новую высоту.

Также мы используем Git для контроля версий проекта и хранения его на GitHub. Git позволяет фиксировать любые изменения в исходном коде, он является незаменимым инструментом при командной разработке приложения. Для проверки на GET и POST запросы мы используем Insomnia. Эта программа позволяет проверить серверную часть приложения без клиентской части приложения. Для работы с базой данных мы используем систему управления базой данных MySQL.

3.5 Функциональное назначение

Опишем наши классы, сервлеты и JSP страницы:

КЛАССЫ:

- DBManager – класс для работы с базой данных;
- Baskets – сущность корзина;
- Categories – сущность категории;
- Items – сущность товары;
- Users – сущность для наших пользователей.

СЕРВЛЕТЫ:

- BasketServlet – сервлет для работы с корзиной;
- CategoriesServlet – сервлет для работы с категориями товаров;

- HomeServlet – сервлет отвечающий за начальную страницу приложения;
- ItemsServlet – сервлет отвечающий за товары в интернет магазине;
- LogoutServlet – сервлет отвечающий за выход;
- ReadItemServlet – сервлет отвечающий за чтение товаров с базы данных;
- SignInServlet – сервлет отвечающий за авторизацию пользователей;
- SignUpServlet – сервлет отвечающий за регистрацию пользователей.

JSP СТРАНИЦЫ:

- index.jsp - основная страница сайта;
- admin.jsp – страница админа;
- basket.jsp – страница для корзины;
- categories.jsp – страница для категории товаров;
- edititem.jsp – страница для редактирования товаров;
- items.jsp – страница для товаров;
- profile.jsp – страница для профиля;
- readitem.jsp – страница для просмотра товаров;
- register.jsp – страница регистрации пользователей;
- 404.jsp – ошибка 404 страница не найдена;
- 403.jsp – 403 доступ запрещен;
- sidebar.jsp – боковая панель;
- navbar.jsp – панель навигации;
- footer.jsp – нижняя часть страницы;
- admin_sidebar.jsp – боковая панель администратора.

Опишем основные функциональные назначения классов и сервлетов, которые использовались при разработке веб-приложения «Интернет-магазин по продаже техники».

Начнем с класса DBManager.java, который отвечает за соединение и работу с базой данных. В этом классе используются различные методы для выполнения определенных задач. Перечислим основные методы:

- метод `public void connect()` отвечает за соединение к нашей базе данных;
- метод `public void addUser(User user)` отвечает за добавление новых пользователей в базу данных;
- метод `public Users getUser(Users user)` возвращает нам нужного пользователя;
- метод `public Users getUser(String token)` возвращает пользователя по его уникальному токenu;
- метод `public void addCategories(Categories categories)` добавляет категории в нашу базу данных;
- метод `public ArrayList<Categories> getAllCategories()` возвращает список всех категорий, которые находятся в базе данных `internet_shop` в таблице `categories`;

- метод `public void deleteCategory(Long id)` удаляет категорию из таблицы `categories`;
- метод `public void addItem(Items item)` добавляет товар в таблицу `items`;
- метод `ArrayList<Items> getAllItems()` возвращает список всех товаров из базы данных;
- метод `public void deleteItem(Long id)` удаляет товар по `id`, переданный в параметры метода;
- метод `public Items getItem(Long id)` возвращает товар по идентификатору `id`;
- метод `public void saveItem(Items item)` обновляет указанный товар в параметре метода;
- метод `public void addBasket(Baskets basket)` добавляет нужный товар в корзину определенного пользователя;
- метод `public ArrayList<Baskets> getBasketsByUser(Users user)` возвращает корзину определенного пользователя;
- метод `public void deleteBasket(Long id)` удаляет товар из нашей корзины;

Классы `Baskets.java`, `Categories.java`, `Items.java`, `Users.java` являются сущностями моего веб-приложения, которые хранятся в базе данных в виде таблиц.

Сервлеты `SignInServlet`, `SignUpServlet`, `ReadItemServlet`, `LogoutServlet`, `ItemsServlet`, `HomeServlet`, `CategoriesServlet`, `BasketServlet` отвечают за бизнес-логику моего веб-приложения. В сервлетах есть `doPost` и `doGet` методы, которые принимают запросы с клиентской стороны с помощью класса параметра `HttpServletRequest` и для ответа `HttpServletResponse`. Каждый сервлет срабатывает по определенной ссылке с помощью аннотаций `@WebServlet`.

Вызов и загрузка. Веб-приложение «Интернет-магазин по продаже техники» запускается через один из браузеров (Google Chrome, Opera, Firefox) по адресу `http://localhost:8080`

Используемые технические средства. При разработке ПО использовался персональный компьютер со следующими характеристиками:

- процессор AMD A10-4600M APU with Radeon(tm) HD Graphics 2.30 GHz;
- оперативная память 6,00 ГБ;
- жесткий диск HDD 1000Gb Seagate mobile, 2.5, 128Mb SATA 3|;
- видеокарта NVIDIA GeForce 9800 GTS 512.

3.6 Работа и апробация веб-приложения «Интернет-магазин по продаже техники»

Рассмотрим работу веб-приложения «Интернет-магазин по продаже техники» на контрольном примере. При запуске приложения по адресу `http://localhost:8080` открывается начальная страница авторизации пользователя,

представленная на рисунке Г.4. Для входа в административную панель пользователю необходимо ввести логин и пароль от аккаунта администратора, в случае если пароль не верен, на странице будет выведено соответствующее сообщение, как показано на рисунке Г.5.

Если логин и пароль введены верно, то откроется административная панель, представленная на рисунке Г.6. Административная панель включает в себя разделы:

- admin;
- categories;
- items;
- my accounting.

При выборе раздела «categories» откроется страница категории товаров, представленная на рисунке Г.7. В разделе categories администратор может добавлять, изменять или удалять категории. На рисунке Г.8 представлена страница по редактированию товаров. На этой странице мы можем добавлять, изменять и удалять различные товары.

Для создания аккаунта нужно нажать кнопку Sign Up, которая позволяет перейти на страницу регистрации пользователя, как показано на рисунке Г.9

.

ЗАКЛЮЧЕНИЕ

Для компании очень важно достичь определенных целей в экономическом плане. Нелегко обойтись без современных информационных технологий. Сегодня ситуация на Казахстанском рынке неоднозначна. Отечественные компании изобретают более современные и продуктивные способы получения прибыли, повышая свою конкурентоспособность. Для формирования сильной клиентской базы необходимо учитывать важные нюансы. Другими словами, нам нужно сосредоточиться на потребностях и пожеланиях наших клиентов. Из этого следует вывод, что основа данного метода управления бизнесом - это клиент. Рентабельность и успех бизнеса зависят от скорости, с которой компании реагируют на меняющиеся рыночные условия.

В процессе написания дипломного проекта в соответствии с заданием была создана база данных `internet_shop` и таблицы `users`, `categories`, `items`, `baskets`. Для создания серверной части использовался язык программирования `java`, а в качестве веб-контейнера `Apache Tomcat`. Для реализации функциональной части веб-приложения использовались такие технологии, как `java EE`, `HTML`, `CSS`, `jsp`. Связь приложения и базы данных осуществляется с помощью драйвера `mysql-connector-java-5.1.25-bin.jar`. Для создания и использования базы данных используется СУБД `MySQL`.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Гонсалвес Э. Изучаем java EE7- СПб.: Питер, 2014 – 640.: ил.
- 2 Шилдт, Герберт Ш57 Java 8: руководство для начинающих, 6-е изд.: Пер с англ. – М.: ООО “И.Д.Вильямс”, 2015 -720с.: ил. – Парал. тит.англ.
- 3 Яргер Р.Дж., Риз Дж. Кинг Т. MySQL и mSQL. Базы данных для предприятий и Интернета. - М.: Символ-Плюс, 2000
- 4 Бен Фрейн HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. – СПб.: Питер, 2014. - 304 с.: ил.
- 5 МакГрат, Майк программирование на java для начинающих –Москва: Издательство «Э», 2016. -192с.
- 6 Хайнеман, Джордж, Поллис, Гэри, Селков, Стэнли Алгоритмы. Справочник с примерами на C, C++, java и Python
- 7 Кириллов В.В., Громов Г.Ю. Введение в реляционные базы данных. - СПб.: БХВ-Петербург, 2014. – 464 с.
- 8 Илюшечкин В.М. Основы использования и проектирования баз данных. – М.: Юрайт, 2015. – 224 с.
- 9 Игошина Л.В. Программирование на языке высокого уровня. Рабочая программа дисциплины. - Пенза: ПГУ, 2014. - с.5
- 10 Asterisk: The Definitive Guide, 4th Edition The Future of Telephony Is Now By Russell Bryant, Leif Madsen, Jim Van Meggelen Publisher: O'Reilly Media Final Release Date: May 2013
- 11 Чудинов И.Л., Осипова В.В. Базы данных: Учебное пособие. – Томск: Изд-во Томского политехнического университета, 2016. – 144 с.
- 12 Date C.J. Database Design and Relational Theory. - O'Reilly Media – 2016, P. 135-136.

Приложение А

Jsp файлы для отображения

```
<%@page import="java.util.ArrayList"%>
<%@page import="com.shop.entities.Items" %>
<%@page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
<title></title>
<%@include file="templates/head.jsp"%>
</head>
<body>
<%@include file="templates/navbar.jsp"%>
<section class="mt-5">
<div class="container pt-5">
<div class="row">
<div class="col-3">
<%@include file="templates/sidebar.jsp"%>
</div>
<div class="col-9">
<div class="row">
<% ArrayList<Items> items = (ArrayList<Items>)request.getAttribute("items");
if(items!=null){
for(Item item: items){
%>
<div class="card m-2" style="width:18rem;">
<div class="card-body">
<h5 class="card-title"><%=item.getName()%></h5>
<p class="card-text">
For <b><%=item.getPrice()%> KZT</b> <br>
Amount :<%=item.getAmount()%><br>
Category:<%=item.getCategory().getName()%><br>
Added at:<%=item.getAddedDate()%>
</p>
<a href="/readitem?id=<%=item.getId()%>" class="btn btn-primary">Details</a>
</div>
</div>
<%
}
}
%>
</div></div>
</div></div>
```

Продолжение приложения А

```
</section>
<%@include file="templates/footer.jsp"%>
</body>
</html>
<%@ page import="com.shop.entities.Users" %>
<%@ page import="com.shop.entities.Items" %>
<%@ page import="java.util.ArrayList" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
<title></title>
<%@include file="templates/head.jsp"%> </head>
<body>
<%@include file="templates/navbar.jsp"%>
<section class="mt-5">
<div class="container pt-5">
<div class="row">
<div class="col-3">
<%@include file="templates/sidebar.jsp"%>
</div>
<div class="col-9">
<nav aria-label="breadcrumb">
<ol class="breadcrumb">
<li class="breadcrumb-item"><a href="/" style="text-decoration: none;">Home</a></li>
<li class="breadcrumb-item active" aria-current="page">Items</li>
</ol>
</nav>
<div class="row">
<%
ArrayList<Items> items = (ArrayList<Items>)request.getAttribute("items");
if(items!=null){
for(Items item : items){
%>
<div class="card ml-2 mb-2" style="width: 18rem;">
<div class="card-body">
<h5 class="card-title"><%=item.getName()%></h5>
<p class="card-text">
For <b><%=item.getPrice()%> KZT</b> <br>
Amount : <%=item.getAmount()%> <br>
Category : <%=item.getCategory().getName()%><br>
Added at : <%=item.getAddedDate()%>
```

Продолжение приложения А

```
</p>
<a href="/readitem?id=<%=item.getId()%>" class="btn btn-primary">Details</a>
</div></div>
<%
}
}
%>
</div>
</div>
</div>
</div>
</section>
<%@include file="templates/footer.jsp"%>
</body>
</html>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
<title></title> <%@include file="templates/head.jsp"%>
</head>
<body><%@include file="templates/navbar.jsp"%>
<section class="mt-5">
<div class="container pt-5">
<div class="row">
<div class="col-6 offset-3">
<%
String success = request.getParameter("success");
if(success!=null){
%>
<div class="alert alert-success" role="alert">
Registration completed successfully!
</div>
<%
}
%>
<%
String mismatch = request.getParameter("mismatch");
if(mismatch!=null){
%>
<div class="alert alert-
danger" role="alert"> Password Mismatch!</div><%=}%><%=String exists = request.g
etParameter("exists");
```

Продолжение приложения А

```
if(exists!=null){
%></div>
<div class="alert alert-danger" role="alert">
User exists!
</div>
<%
}
%>
<%
String uncomplete = request.getParameter("uncomplete");
if(uncomplete!=null){
%>
<div class="alert alert-danger" role="alert">
Complete form!
</div>
<%
}
%>
<form action="/signup" method="post">
<div class="form-group row"><label class="col-sm-4 col-form-label">Email</label>
<div class="col-sm-
8"><input type="email" class="formcontrol" placeholder="Email" name="email">
</div>
</div>
<div class="form-group row">
<label class="col-sm-4 col-form-label">Password</label><div class="col-sm-8">
<input type="password" class="form
control" placeholder="Password" name="password"></div></div><div class="form-
group row"><label class="col-sm-4 col-form-label">Confirm Password</label>
<div class="col-sm-8"><input type="password" class="form-
control" placeholder="Confirm Password" name="re-password"></div></div>
<div class="form-group row"><label class="col-sm-4 col-form-label">Email</label>
<div class="col-sm-
8"><input type="text" class="formcontrol" placeholder="Full Name" name="full_na
me">
</div></div><div class="form-group row"><div class="col-sm-
12"><button type="submit" class="btn btn-primary">Sing Up</button>
</div></div></form></div></div></div></section><%@include file="templates/foo
ter.jsp"%>
</body>
</html><%@ page import="com.shop.entities.Users" %>
<%@ page import="java.util.ArrayList" %>
```

Продолжение приложения А

```
<%@ page import="com.shop.entities.Categories" %>
<%@ page import="com.shop.entities.Items" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
<title></title><%@include file="templates/head.jsp"%></head><body><%@include
file="templates/navbar.jsp"%><section class="mt-5">
<div class="container pt-5"><div class="row"><div class="col-3">
<%@include file="templates/sidebar.jsp"%>
</div><div class="col-9"><%Items item = (Items) request.getAttribute("item"); %>
<nav aria-label="breadcrumb"><ol class="breadcrumb"><li class="breadcrumb-
item"><a href="/" style="text-decoration: none;">Home</a></li>
<li class="breadcrumb-item"><a href="/" style="text
decoration: none;">Items</a></li> <li class="breadcrumb-item active" aria-
current="page"><%=item.getName()%></li> </ol></nav><form action="basket" me
thod="post"><input type="hidden" name="operation" value="add"><input type="hid
den" name="item_id" value="<%=item.getId()%>"><div class="row"><div class="c
ol-3">NAME : </div><div class="col-9"><%=item.getName()%></div></div>
<div class="row">
<div class="col-3">PRICE : </div>
<div class="col-9"><%=item.getPrice()%> KZT</div>
</div><div class="row"><div class="col-3">CATEGORY : </div><div class="col-
9"><%=item.getCategory().getName()%></div>
</div>
<div class="row">
<div class="col-3">AMOUNT : </div>
<div class="col-9"><%=item.getAmount()%></div>
</div><div class="row"><div class="col-3">ADDED DATE : </div>
<div class="col-9"><%=item.getAddedDate()%></div></div><%
if(userData!=null){%><div class="row mt-5"><div class="col-12">
<button type="submit" class="btn btn-primary">ADD TO BASKET</button>
</div>
</div>
<%
}
%> </form>
</div>
</div>
</div>
</section>
<%@include file="templates/footer.jsp"%>
</body>
```

Приложение Б

Сервлеты для GET И POST запросов

```
package com.shop.servlets;
import com.shop.db.DBManager;
import com.shop.entities.Users;
import org.apache.commons.codec.digest.DigestUtils;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/signin")
public class SignInServlet extends HttpServlet {
    private DBManager dbManager;
    public void init(){
        dbManager = new DBManager();
        dbManager.connect();
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String message = "";
        String email = request.getParameter("email");
        String password = request.getParameter("password");
        if(email!=null&&!email.trim().equals("")&&password!=null&&!password.trim().equals("")){
            Users user = dbManager.getUser(new Users(null, email, null, null, 0));
            if(user!=null){
                if(user.getPassword().equals(password)){
                    request.getSession().setAttribute("USER_SESSION", user);
                    String remember = request.getParameter("remember");
                    if(remember!=null&&remember.equals("remember")){
                        String rememberUserToken = DigestUtils.sha1Hex(user.getEmail()+user.getPassword()+"bitlab");
                        Cookie cookie = new Cookie("rememberUser", rememberUserToken);
                        cookie.setMaxAge(3*3600*24);
                        response.addCookie(cookie);
                    }
                }else{
                    message = "password";
                }
            }
        }
    }
}
```

Продолжение приложения Б

```
}else{
message = "user";
}
}else{
message = "uncomplete";
}
response.sendRedirect("/?" + message);
}
}

package com.shop.servlets;
import com.shop.db.DBManager;
import com.shop.entities.Users;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/signup")
public class SignUpServlet extends HttpServlet {
private DBManager dbManager;
public void init(){
dbManager = new DBManager();
dbManager.connect();
}
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
String message = "error";
String email = request.getParameter("email");
String password = request.getParameter("password");
String rePassword = request.getParameter("re-password");
String fullName = request.getParameter("full_name");
if(email!=null&&!email.trim().equals(""))&&password!=null&&!password.trim().equals(""))&&rePassword!=null&&!rePassword.trim().equals("")){
if(password.equals(rePassword)){
Users exists = dbManager.getUser(new Users(null, email, null, null, 0));
if(exists==null){
dbManager.addUser(new Users(null, email, password, fullName, 2));
message = "success";
}else{
message = "exists";
}
}
```

Продолжение приложения Б

```
}else{
message = "mismatch";
}
}else{
message = "uncomplete";
}
response.sendRedirect("signup?" + message);
}
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException { request.getRequestDispatcher("register.jsp").forward(request, response);
}
}
package com.shop.servlets;
import com.shop.db.DBManager;
import com.shop.entities.Categories;
import com.shop.entities.Items;
import com.shop.entities.Users;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
@WebServlet("/readitem")
public class ReadItemServlet extends HttpServlet {
private DBManager dbManager;
public void init(){
dbManager = new DBManager();
dbManager.connect();
}
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
String itemId = request.getParameter("id");
Long id = null;
try {
id = Long.parseLong(itemId);
}catch (Exception e){
}
String name = request.getParameter("name");
int amount = Integer.parseInt(request.getParameter("amount"));
```

Продолжение приложения Б

```
int price = Integer.parseInt(request.getParameter("price"));
Long categoryId = Long.parseLong(request.getParameter("category_id"));
dbManager.saveItem(new Items(id, name, price, amount, new Categories(categoryId,
null), null ));
response.sendRedirect("readitem?id="+id);
}
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    Users userData = (Users)request.getAttribute("USER_DATA"); String jspPage = "404.jsp";
    String itemId = request.getParameter("id");
    Long id = null;
    try {
        id = Long.parseLong(itemId);
    } catch (Exception e){
    }
    Items item = null;
    if(id!=null){
        item = dbManager.getItem(id);
    }
    if(item!=null) {
        if (userData != null) {
            if (userData.getRoleId() == 1) {
                ArrayList<Categories> categories = dbManager.getAllCategories();
                request.setAttribute("categories", categories);
                request.setAttribute("item", item);
                request.getRequestDispatcher("edititem.jsp").forward(request, response);
            } else {
                request.setAttribute("item", item);
                request.getRequestDispatcher("readitem.jsp").forward(request, response);
            }
        } else {
            request.setAttribute("item", item);
            request.getRequestDispatcher("readitem.jsp").forward(request, response);
        }
    } else {
        request.getRequestDispatcher("404.jsp").forward(request, response);
    }
}
```

Продолжение приложения Б

```
package com.shop.servlets;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/logout")
public class LogoutServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.getSession().removeAttribute("USER_SESSION");
        Cookie[] cookies = request.getCookies();
        for(Cookie c : cookies){
            if(c.getName().equals("rememberUser")){
                c.setMaxAge(0);
                response.addCookie(c);
                break;
            }
        }
        response.sendRedirect("/");
    }
}

package com.shop.servlets;
import com.shop.db.DBManager;
import com.shop.entities.Categories;
import com.shop.entities.Items;
import com.shop.entities.Users;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
import javax.servlet.*;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.io.IOException;
```

Приложение В

Классы сущностей для базы данных

```
package com.shop.entities;
public class Users {
    private Long id;
    private String email;
    private String password;
    private String fullName;
    private int roleId; // 1 - ADMIN, 2 - USER
    public Users() {
    }
    public Users(Long id, String email, String password, String fullName, int roleId) {
        this.id = id;
        this.email = email;
        this.password = password;
        this.fullName = fullName;
        this.roleId = roleId;
    }
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getFullName() {
        return fullName;
    }
    public void setFullName(String fullName) {
        this.fullName = fullName;
    }
}
```

Продолжение приложения В

```
public int getRoleId() {
return roleId;
}
public void setRoleId(int roleId) {
this.roleId = roleId;
}
}
package com.shop.entities;
import java.util.Date;
public class Items {
private Long id;
private String name;
private int price;
private int amount;
private Categories category;
private Date addedDate;
public Items() {
}
public Items(Long id, String name, int price, int amount, Categories category, Date a
ddedDate) {
this.id = id;
this.name = name;
this.price = price;
this.amount = amount;
this.category = category;
this.addedDate = addedDate;
}
public Long getId() {
return id;
}
public void setId(Long id) {
this.id = id;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public int getPrice() {
return price;
}
}
```

Продолжение приложения В

```
public void setPrice(int price) {
    this.price = price;
}
public int getAmount() { return amount;
}
public void setAmount(int amount) {
    this.amount = amount;
}
public Categories getCategory() {
    return category;
}
public void setCategory(Categories category) {
    this.category = category;
}
public Date getAddedDate() {
    return addedDate;
}
public void setAddedDate(Date addedDate) {
    this.addedDate = addedDate;
}
}
package com.shop.entities;
public class Categories {
    private Long id;
    private String name;
    public Categories() {
    }
    public Categories(Long id, String name) {
        this.id = id;
        this.name = name;
    }
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() { return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

Продолжение приложения В

```
package com.shop.entities;
public class Baskets {
    private Long id;
    private Users user;
    private Items item;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public Users getUser() {
        return user;
    }
    public void setUser(Users user) {
        this.user = user;
    }
    public Items getItem() {
        return item;
    }
    public void setItem(Items item) {
        this.item = item;
    }
    public Baskets(Long id, Users user, Items item) {
        this.id = id;
        this.user = user;
        this.item = item;
    }
    public Baskets() {
    }
}
```

Приложение Г

Структура и интерфейс веб-приложения

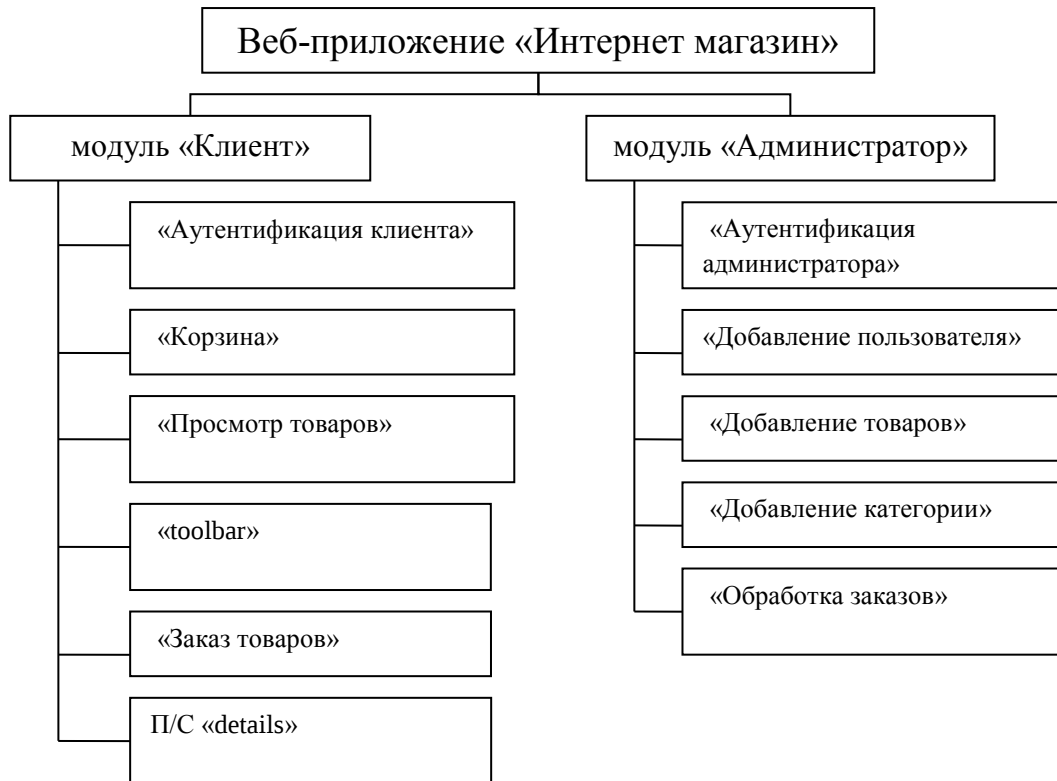


Рисунок Г.1 – Структура веб-приложения «Интернет-магазин по продаже техники»

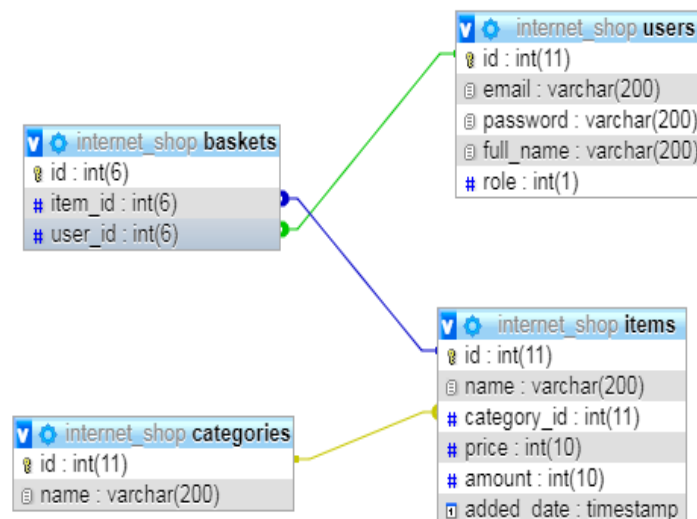


Рисунок Г.2 – Схема разработанной базы данных

Продолжение приложения Г

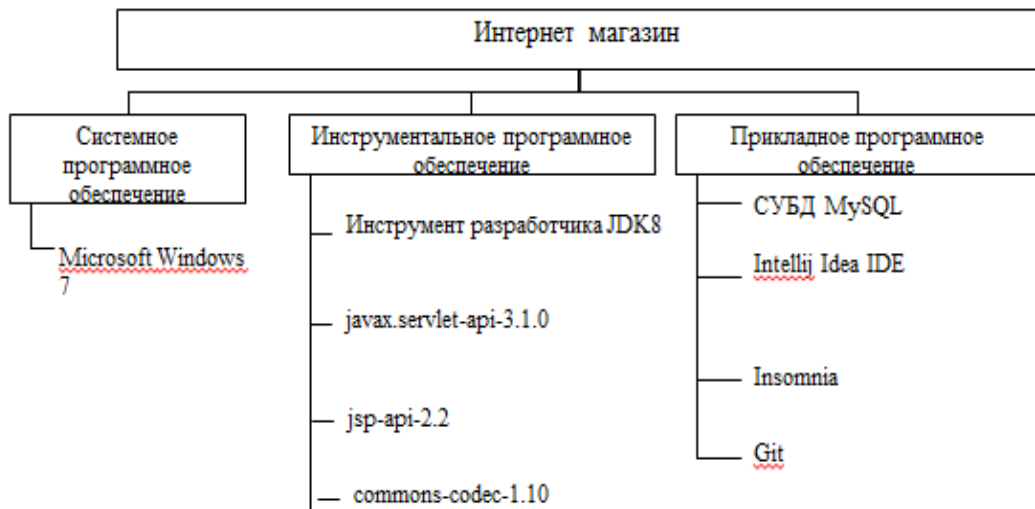


Рисунок Г.3 – Программное обеспечение веб-приложения «Интернет-магазин по продаже техники»

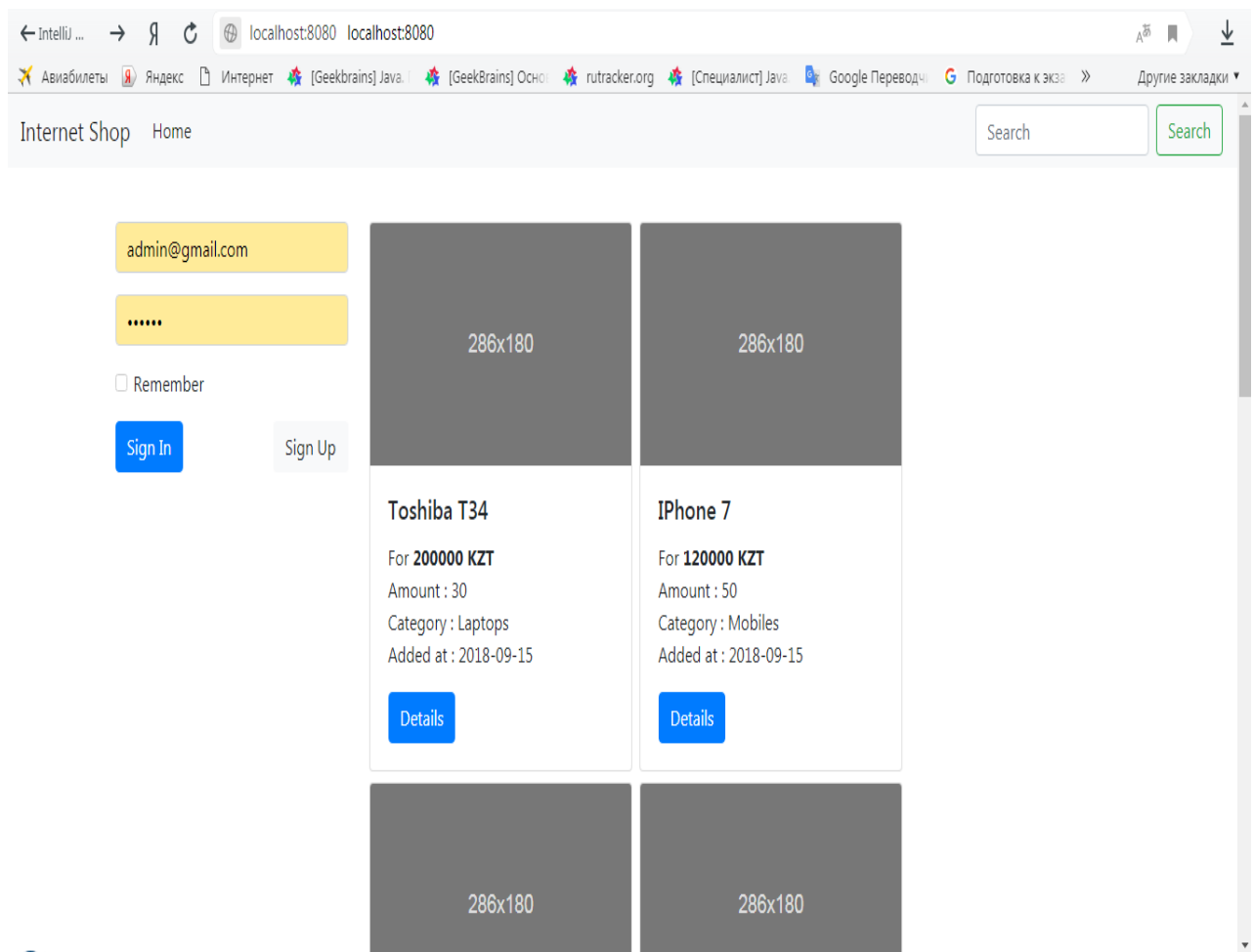


Рисунок Г.4 – Страница авторизации

Продолжение приложения Г

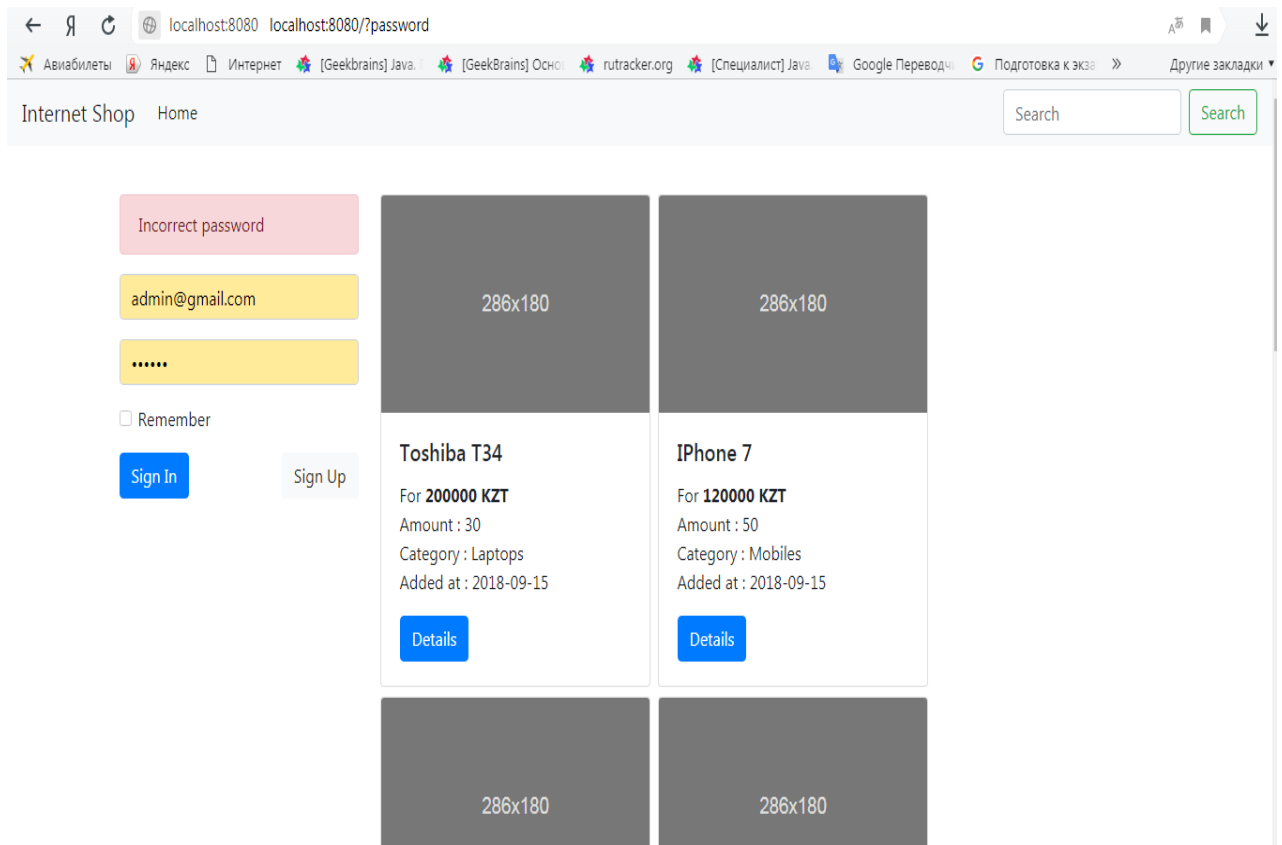


Рисунок Г.5 – Ошибка при авторизации

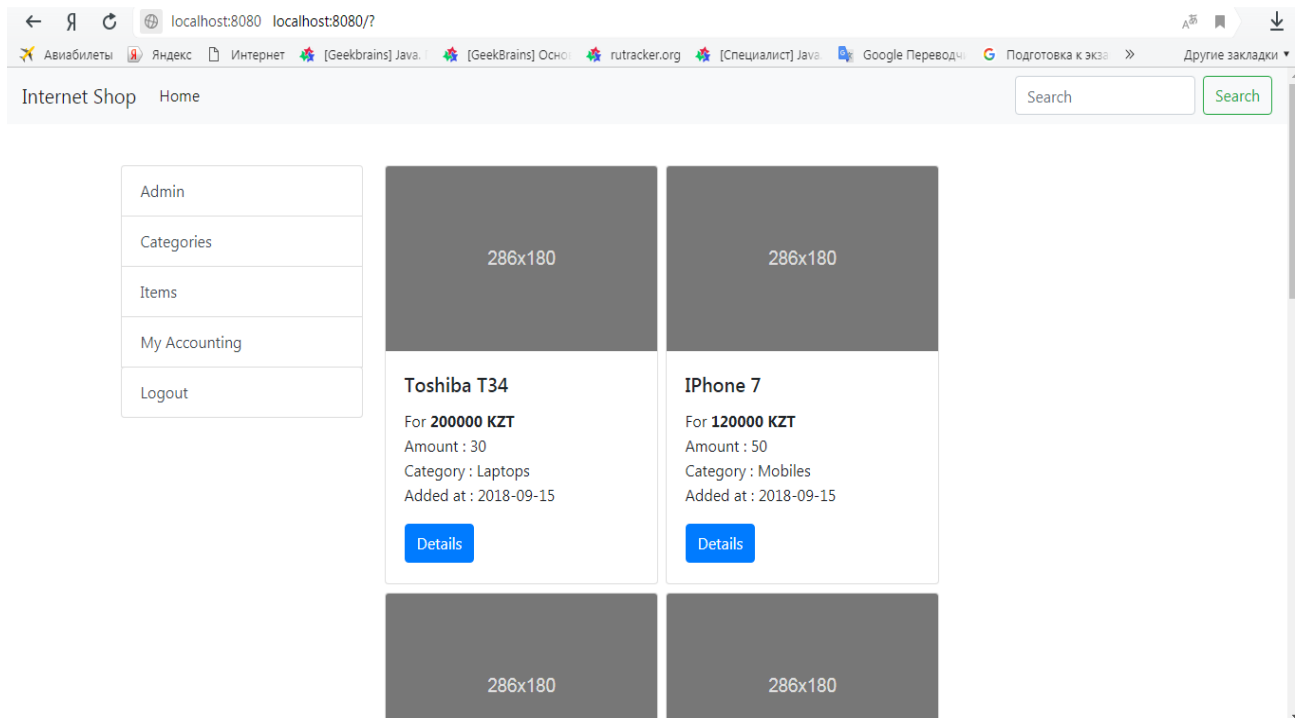


Рисунок Г.6 – Административная панель

Продолжение приложения Г

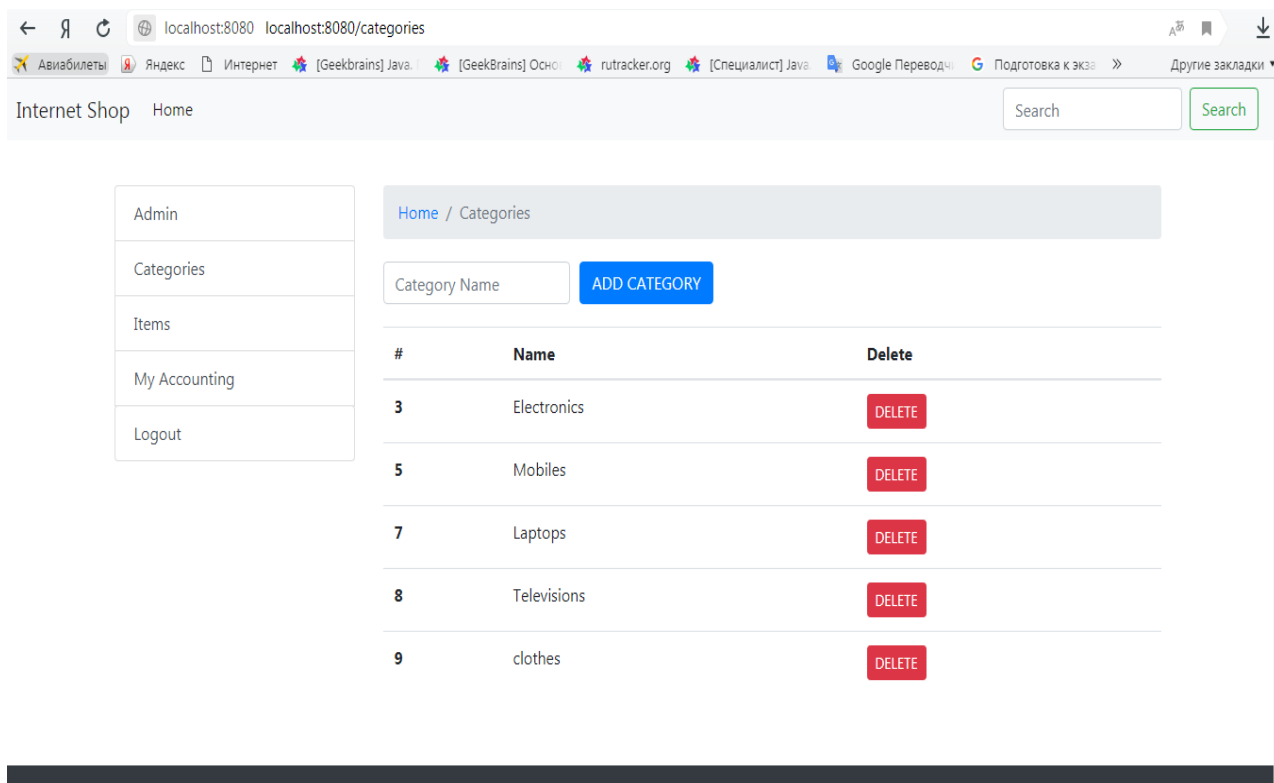


Рисунок Г.7 – Страница категории

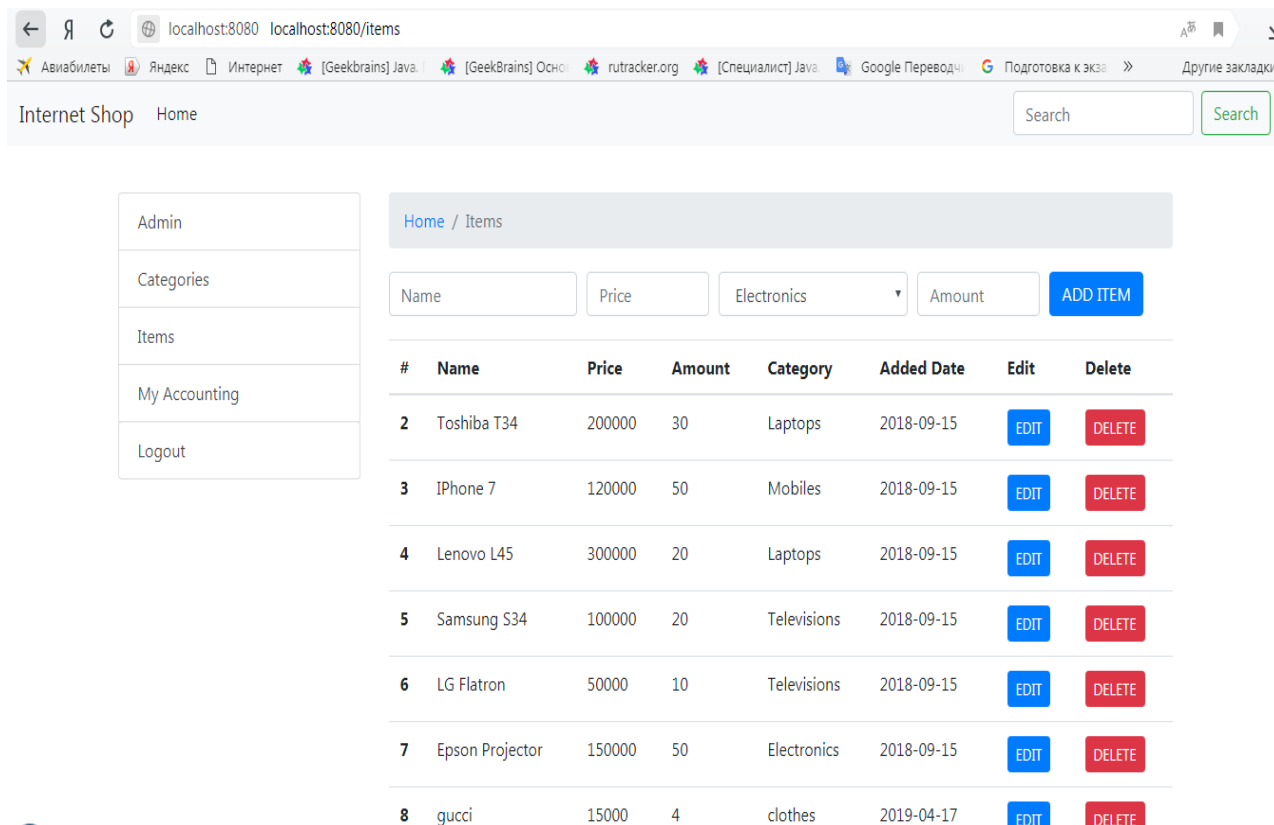


Рисунок Г.8 – Страница товаров

Продолжение приложения Г

The screenshot shows a web browser window with the address bar displaying 'localhost:8080 localhost:8080/signup'. The browser's bookmark bar contains several links, including 'Авиабилеты', 'Яндекс', 'Интернет', '[Geekbrains] Java', '[Geekbrains] Основы', 'rutracker.org', '[Специалист] Java', 'Google Переводчик', 'Подготовка к экзаменам', and 'Другие закладки'. The page header includes the text 'Internet Shop' and a 'Home' link, followed by a search bar with a 'Search' button. The main content area contains a registration form with the following fields and labels:

- Email (input field)
- Password (input field)
- Confirm Password (input field)
- Email (input field, labeled 'Full Name' in the input field)

Below the form is a blue 'Sing Up' button. The footer of the page contains the text 'Copyright 2018 (C). Internet Shop'.

Рисунок Г.9 – Страница регистрации