

## 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:** Шин Александра

**Название:** александра шин

**Координатор:** Гульнара Омарова

**Коэффициент подобия 1:**3,1

**Коэффициент подобия 2:**0

**Тревога:**2

**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:

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

16.05.19.  
.....

Дата

.....  
.....

Подпись Научного руководителя

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломный проект  
(наименование вида работы)  
Шин Александра  
(Ф.И.О. обучающегося)  
5B070400 – Вычислительная техника и программное обеспечение

(шифр и наименование специальности)

Тема: «Разработка веб-приложения для связи с компаниями по  
продаже воздухо-вентиляционного оборудования»

Актуальность выбранной дипломной работы заключается в том, что на современном веб рынке достаточно много сайтов, специализирующихся на продажах. В свою очередь каждый такой сайт несет узконаправленный характер. Данная работа позволяет значительно сэкономить рабочее время, позволяет создать честную конкуренцию и объединить компании, которые занимаются продажами вентиляционного оборудования.

Дипломная работа состоит из четырех разделов.

Во введении обусловлена актуальность темы, основные задачи ЭТП.

Первый раздел посвящен общему описанию дипломной работы. Рассмотрены цель работы разработки веб-приложения, определения, термины и сокращения, предметная область.

Во втором разделе изложены требования к разработке веб-приложения. Автор включает в данную главу: техническое задание, разработка модели бизнес-прецедентов и логическую и физическую модели

В третьем разделе студентка описывает, какие технологии были применены при разработке данного веб-приложения.

Четвертый раздел включает в себя заверченный интерфейс проекта.

Из заключения стало видно, что студентка серьезно и ответственно подошла к работе.

В целом данный дипломный проект выполнен с учетом всех требований, предъявляемых к дипломному проекту по специальности 5B070400 - «Вычислительная техника и программное обеспечение», студент Шин А. рекомендован к защите дипломного проекта и заслуживает присвоения академической степени «бакалавра» по специальности 5B070400 - «Вычислительная техника и программное обеспечение».

Научный руководитель

лектор

(должность, уч. степень, звание)

 Омарова Г. А.

(подпись)

«16»  2018г.

## АКТ ВНЕДРЕНИЯ

### веб-приложения для связи с компаниями по продаже воздухо-вентиляционного оборудования

1. Разработанное студенткой 4-го курса кафедры Программной инженерии Шин А.Р. приложение было передано в тестовую эксплуатацию в мае 2019 года для использования в качестве электронно-торговой площадки.
2. Эксплуатационные характеристики объекта внедрения.  
Программный продукт включает в себя:
  - необходимые справочники;
  - базу данных, содержащую необходимые сведения (категории воздухо-вентиляционного оборудования, типы пользователя);
3. Назначение программного продукта: поддержка связи с компаниями, продающими воздухо-вентиляционное оборудование, получение информации о компании, оформление заказа.
4. Показатели экономической и социальной эффективности программного продукта:
  - Экономия рабочего времени;
  - Создание честной конкуренции;
  - Объединение компаний, которые специализируются на продажах воздухо-вентиляционного оборудования.
5. ТОО EES Kazakhstan обязуется не передавать программный код в другие организации.

**Представители ТОО EES KAZAKHSTAN:**

Директор компании ТОО EES KAZAKHSTAN

**Исполнители:**

Студентка группы ВПБ15-1р

Сембаев Е.А.

Шин А.Р.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра "Программная инженерия"

Шин А.Р.

Разработка веб-приложения для связи с компаниями по продаже воздухо-  
вентиляционного оборудования

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное  
обеспечение

Алматы 2019

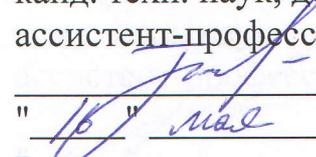
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра "Программная инженерия"

**ДОПУЩЕН К ЗАЩИТЕ**  
Заведующий кафедрой ПИ  
канд. техн. наук, доцент,  
ассистент-профессор

 Р. Юнусов  
" 16 " май 2019  
г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

На тему: " Разработка веб-приложения для связи с компаниями по продаже  
воздухо-вентиляционного оборудования "

по специальности 5В070400 – Вычислительная техника и программное  
обеспечение

Выполнила

Шин А.Р.

Научный руководитель  
лектор

 Г.А.Омарова  
" 16 " май 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

**УТВЕРЖДАЮ**

Заведующий кафедрой ПИ  
канд. техн. наук, доцент,  
ассистент-профессор

Р. Юнусов

" 16 " мая 2019

г.

**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Обучающемуся Шин Александре Робертовне

Тема: Разработка веб-приложения для связи с компаниями по продаже воздухо-вентиляционного оборудования

Утверждена приказом проректора по академической работе № 1162 - б  
от "16" октября 2018 г.

Срок сдачи законченного проекта "17" мая 2019 г.

Исходные данные к дипломному проекту: техническая документация по применению технологии, техническое задание, описание БД для хранения информации в виде ER-диаграммы.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) разработка классовой модели реляционной базы данных MySQL согласно представленной диаграмме;

б) реализация многозвенной архитектуры веб-приложения в соответствии с концепцией MVC;

в) проектирование и разработка пользовательского интерфейса;

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

Рекомендуемая основная литература: из 45 наименований.

## ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области	17.10.18	-
2. Разработка технического задания на проектирование программного комплекса	1.11.18	-
3. Реализация классовой структуры модели базы	27.11.18	-
4. Разработка пользовательского интерфейса, реализация контроллеров для исполнения функций пользователя	23.12.18 - 2.02.19	-
5. Разработка структуры приложения с реализацией логической цепочки иерархических вызовов Тестирование приложения	8.03.19	-
6. Написание пояснительной записки к дипломному проекту	6.05.19	-

### Подписи

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

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	З.М.Өмірбекова ассистент	6.05.19.	
Нормоконтролер	Г.А.Омарова лектор	6.05.19.	

Научный руководитель \_\_\_\_\_

 Г.А.Омарова

Задание принял к исполнению обучающийся \_\_\_\_\_

 А.Р.Шин

Дата

" 6 " мая 2019 г.

## АННОТАЦИЯ

Данный дипломный проект состоит из введения, основных четырех разделов и заключения.

Во введении обусловлена актуальность темы, основные задачи ЭТП.

Первый раздел посвящен общему описанию дипломной работы. Рассмотрены цель работы разработки веб-приложения, определения, термины и сокращения, предметная область.

Во втором разделе изложены требования к разработке веб-приложения. В данную главу включены: техническое задание, разработка модели бизнес-прецедентов и логическую и физическую модели

В третьем разделе описываются, какие технологии были применены при разработке данного веб-приложения.

Четвертый раздел включает в себя заверченный интерфейс проекта.

В заключении описан результат дипломного проекта.

## ANNOTATION

This diploma project consists of an introduction, the main four sections and conclusion.

In the introduction due to the relevance of the topic, the main objectives of the EM.

The first section is devoted to the general description of the thesis. The purpose of the work of developing a web application, definitions, terms and abbreviations, the subject area.

The second section outlines the requirements for developing a web application. This chapter includes: technical specifications, development of a business precedent model and logical and physical models.

The third section describes what technologies have been applied in the development of this web application.

The fourth section includes a complete project interface.

In the conclusion describes the result of the graduation project.

## АҢДАТПА

Бұл диплом жобасы кіріспеден, негізгі төрт бөлімнен және қорытындыдан тұрады.

Кіріспе бөлімінде тақырыптың өзектілігі, ЭТЖ негізгі міндеттері баяндалған.

Бірінші бөлім дипломның жалпы сипаттамасына арналған. Веб-қосымшаларды, анықтамаларды, терминдерді және қысқартуларды, тақырыптық саланы дамыту жұмысының мақсаты сипатталған.

Екінші бөлім веб-қосымшаны әзірлеуге қойылатын талаптарды сипаттайды. Бұл тарауға мыналар кіреді: техникалық сипаттамалар, бизнес прецедент үлгісін және логикалық және физикалық үлгілерді дамыту.

Үшінші бөлім осы веб-қосымшаның дамуында қандай технологиялар қолданылғанын сипаттайды.

Төртінші бөлім толық жобалық интерфейсін қамтиды.

Қорытындыда бітіру жобасының нәтижесі сипатталады.

## СОДЕРЖАНИЕ

	Введение	9
1	Общее описание	10
1.1	Цель разработки веб-приложения	10
1.2	Определения, термины и сокращения	10
1.3	Предметная область	11
2	Требования к разработке	15
2.1	Техническое задание	15
2.2	Разработка модели бизнес-прецедентов	18
2.3	Логическая и физическая модели	19
3	Технологии	21
3.1	Технологии для создания программного обеспечения	21
4	Интерфейс приложения	29
4.1	Интерфейс	29
	Заключение	40
	Список использованной литературы	41
	Приложение А	44

## ВВЕДЕНИЕ

В современном мире роль интернета достаточно сложно переоценить. На данный момент около 30% населения земли используют интернет ежедневно, а это немного немало 1 500 000 человек. С каждым днем количество тех, кто начинает использовать его – растет. Но одним из больших плюсов интернета – это появление на его просторах: интернет магазинов или иначе торговых площадок, которые помогают сэкономить время и деньги. Ведь появление интернет магазинов, в каком то роде, облегчило жизнь современного человека, который вечно занят и не может найти время для совершения покупок. Интернет площадки просто переполнены интересными, яркими предложениями, создающими искусственную конкуренцию. Но большинство интернет магазинов – однонаправлены, то есть специализируются только на одном виде товаров. Так, редко можно встретить торговую площадку по продажам воздухо-вентиляционного оборудования, которая поможет объединить компании, специализирующихся на данном виде товаров.

Электронная торговая площадка – аппаратно программный комплекс, который позволяет объединить в одном торговом и информационном пространстве продавцов и покупателей. ЭТП можно назвать любой веб ресурс с помощью которого осуществляются сделки купли продажи.

Актуальность выбранной дипломной работы заключается в том, что на современном веб рынке достаточно много сайтов, специализирующихся на продажах. В свою очередь каждый такой сайт несет узконаправленный характер. Например: магазины одежды, продажа машин, техники и т.д. Но для воздухо-вентиляционного оборудования, аналогов на казахстанском рынке выявлено не было. Данная работа позволяет значительно сэкономить рабочее время, позволяет создать честную конкуренцию и объединить компании, которые занимаются продажами вентиляционного оборудования.

Основными задачами электронных торговых площадок являются:

- информационная задача, которая знакомит с компаниями, работающими на ЭТП;
- задача защиты - криптографическая защита информации;
- торговая задача – это задача, которая позволяет осуществить комплекс действий для того, чтобы эффективно продать собственный товар;

Данный дипломный проект представляет реализацию решения, которое позволяет создать простую торговую площадку и создать чистую конкуренцию. Но недостатком системы можно назвать то, что приложение не до конца автоматизировано и понадобится рука человека. Но решение может быть улучшено в соответствие с требованиями заказчика.

## 1 Общее описание

### 1.1 Цель разработки веб-приложения

Веб приложение для связи с компаниями по продаже воздухо-вентиляционного оборудования представляет собой интернет площадку, которая позволяет объединить продавца и покупателя в едином информационном пространстве. Отличие от обычных торговых площадок – узконаправленность, так как данное веб приложение предназначено только для компаний, которые продают воздухо-вентиляционное оборудование. Основное функциональное назначение – это возможность связи с продавцом(компанией), предоставляющая свой товар, потребитель сам может предложить цену и дальше вести переговоры. Информационная функция приложения позволяет получить сведения о компании, которая предоставляет нужный товар. Данное приложение позволяет значительно экономить рабочее время, позволяет создать честную конкуренцию и объединить компании, которые занимаются продажами вентиляционного оборудования.

Компания вносит свои данные в систему и происходит регистрация. В свою очередь администратор отслеживает информацию, которая была заполнена. То есть контроль контента сайта осуществляется администраторами. После регистрации можно добавить товары, которые, данная организация, собирается продавать. В окне у покупателя, высвечивается полная информация о товаре и компании с возможностью связи.

### 1.2 Определения, термины и сокращения

В таблице 1 сформулированы все сокращения и определения, используемые в предметной области при разработке проекта, а также специфичные термины, которые связаны с программной реализацией приложения и технологиями, применяемые при разработке.

**Таблица 1 – Определения, сокращения и их значения**

Сокращение или термин	Определение
MVC	Model – View – Controller
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
DFD	Data flow diagrams

### Продолжение таблицы 1

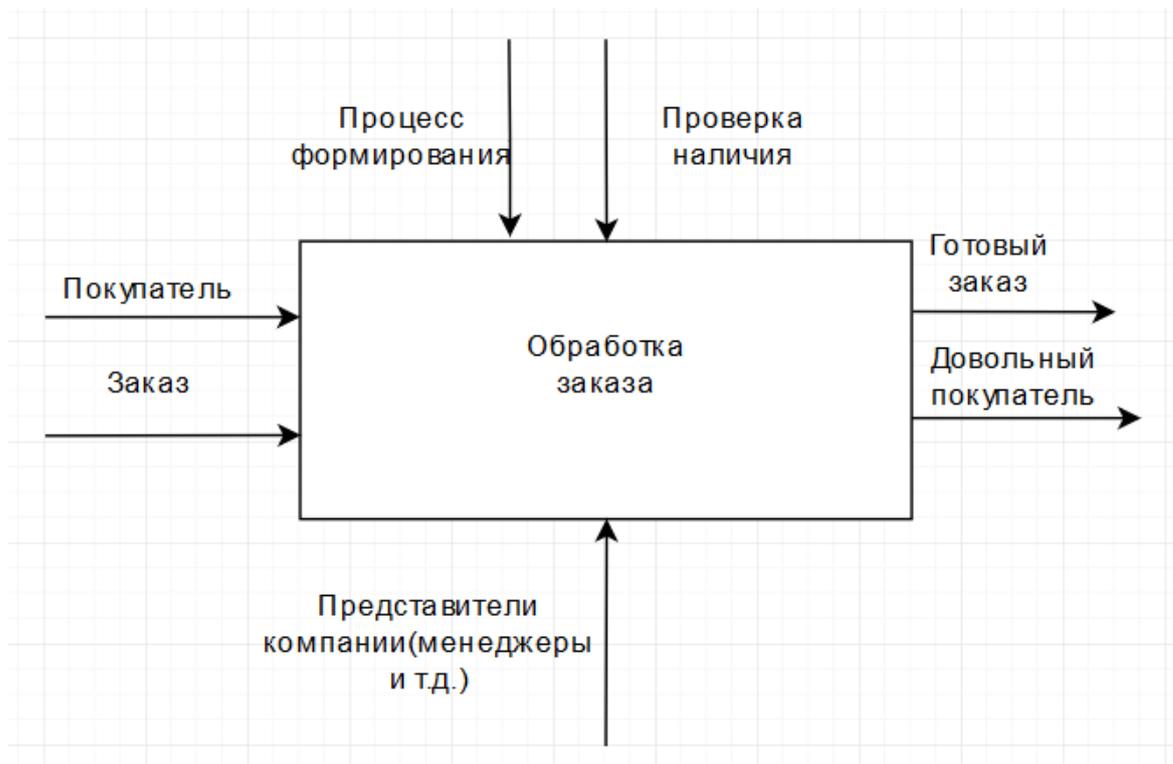
MySQL	Structured query language
ERD	Entity-relationship model
BC	Вентиляционная система
БД	База данных
СУБД	Система управления базами данных
MVP	Model View Presenter
SQL	Structure Query Language
UML	Unified Modeling Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation

### 1.3 Предметная область

В качестве предметной области была выбрана компания по продажам воздухо-вентиляционного оборудования – ТОО EES Kazakhstan. Это коммерческое предприятие, которое продает вентиляционное оборудование, такое как приточные установки, огнезадерживающие клапаны, аксессуары для систем вентиляции (шпилька резьбовая, забивной анкер, скоба, скотч алюминиевый, профиль монтажный, гибкая вставка, уголок оцинкованный, лента уплотнительная, анкер расклинивающий, шина монтажная и т.д.), воздухораспределители, клапаны дымоудаления. Весь ассортимент товара располагается на складах, принадлежащих собственности компании. Каждый заинтересованный клиент может позвонить в офис и узнать о наличии необходимых товаров, а также о их качестве, стране производства, размерах и цене. Для процесса продажи необходимо личное присутствие покупателя и представителя компании.

Вышеизложенное можно представить в виде DFD смотреть рисунок 1.

DF диаграмма включает в себя функциональный блок, стрелки входа и выхода, а также стрелки управления и механизма. В данном случае обработка заказа является функциональным блоком, далее на вход поступает покупатель с заказом, который будут выполнять представители компании – механизмы. Они проверят наличие и начнут процесс формирования. И на выходе получается готовый заказ и довольный покупатель.



**Рисунок 1** – Процесс обработки заказов

Продажи – это бизнес-понятие, которое представляет собой коммерческую деятельность. Под продажами понимается современный подход для организации компании в сфере торговли, таким образом, чтобы при применении рентабельных методов, компания достигла конкретных результатов. В процессе продажи участвуют продавец и покупатель, которые заинтересованы в заключении сделки. Они принимают участие в согласовании условий при обмене товара на денежную единицу.

Продавец – юридическое лицо, которое предлагает и продает свои услуги или товары. Продавец за определенное воздаяние передает услугу или товар покупателю, данный процесс называется сделкой. Продавец при заключении сделки, ищет определенную выгоду – извлечение прибыли.

К основным обязанностям продавца можно отнести:

- Проверка постоянного наличия товара на складах;
- Подготовка товаров для их продажи, сюда входят: сортирование, обеспечение наличия ценников на товаре;
- Осведомленность о характеристиках и об ассортименте товара;
- Знание правил использования товара;
- Передача сведений клиентам о ценах;
- Передача товара за денежную единицу;
- Заказ недостоющих товаров;
- Получение и подтверждение товара;

Покупатель – физическое лицо, либо организация, которая хочет приобрести какой либо товар или услугу взамен на денежный эквивалент. Покупатель приобретает товар за денежную единицу для того чтобы удовлетворить потребности потребителя.

Вентиляция – это такой процесс, который позволяет удалить из желаемого помещения загрязненный воздух и заменить его внешним. В отдельных случаях может потребоваться фильтрация воздуха, а также его кондиционирование, подогрев или охлаждение, ионизация и т.д. Вентиляция необходима в целях соблюдения санитарно-гигиенических норм, сюда входят чистота воздуха, его влажность, температура и т.д. Это положительно влияет на самочувствие и здоровье человека. До начала XIX века под вентиляцией подразумевалось обычное проветривание помещения. Первая теория была создана М.В. Ломоносовым, она была о естественном движении воздуха в трубах. В 1835 году стали появляться первые вентиляторы, которые применялись для проветривания рудника на Алтае. Этот же вентилятор был использован для проветривания трюмов, жилых помещений и для ускорения испарения и сушки.

Основные причины использования вентиляции – это необходимость в борьбе с вредными выделениями:

- Пыль;
- Различные вредные газы;
- Избыточная влага;
- Метеоризмы;
- Пары вредных веществ;
- Избыточное тепло;

Вентиляционные системы – это совокупность устройств, которые обрабатывают, транспортируют, подают и удаляют воздух. К вентиляционным системам можно отнести большое количество разного вида устройств и оборудования, которые можно классифицировать таким образом:

- Тип функционирования;
- Тип конструкции: диаметральное, радиальное, осевое;
- Специфика обслуживания;
- Прямое назначение;

Все конструкции и механизмы, которые направляют свою деятельность на совершенствование потоков воздуха, включены в класс ВС. Данные механизмы необходимы для обеспечения притока внешнего воздуха и для того, чтобы осуществить вытяжку вредного воздуха с вспомогательными функциями обработки, такими как фильтрация, подогрев, ионизация, охлаждение и т.д. Вентиляционное оборудование отличается друг друга: по степени функциональности и, для какого вида помещения предназначено. Например: офис, дом, завод и т.д.

Принудительные или искусственные ВС могут найти свое применение, как для вентиляции жилого помещения – сюда входит кондиционирование,

фильтрация отработанного воздуха и создание, близко к идеальному, микроклимату, так и для крупного промышленного производства. Для сложного вентиляционного процесса используют воздухонагреватели, дымососы, охладители воздуха, модульные системы для кондиционирования, системы для удаления дыма. (см. рисунок 1.1)



**Рисунок 1.1** – Подвиды вентиляционных систем

К отдельному виду ВС относятся каналные системы, которые нужны для того, чтобы выполнять оснащение вентиляционных каналов. В магазинах, в основном, можно встретить прямоугольные и круглые каналы для монтажа. Существенную роль, в вентиляционных системах, играют шумоглушители. Шумоподавляющие механизмы, также можно встретить как для прямоугольных каналов, так и для круглых.

К вспомогательному типу вентиляционного оборудования можно отнести элементы, необходимые для полной комплектации оборудования: решетки, заслонки, фильтры, клапаны, циклоны, колеса и многое другое.

Одним из важных вентиляционных устройств является воздуховод. Это устройство необходимо для того, чтобы правильно переносить воздушные массы. Иначе говоря, воздуховод – это труба. Она состоит из различных фасонных деталей и прямых участков, которые помогают урегулировать поток воздуха. Материал из которого производят воздуховоды: полиэтилен, асбестоцемент, шлако-гипс, кирпич и, конечно, из качественной стали. Также эти устройства делятся по типу установки:

- Приставляются к стенам;
- Выкладываются под балками-перекрытиями;
- Встраиваются в конструкции;

На данный момент представлено более десяти видов данных конструкций.

Вентиляционная система может представлять из себя многоуровневую систему вентилирования на промышленном производстве или может служить для обычного кондиционирования жилого помещения.

## 2 Требования к разработке

### 2.1 Техническое задание

#### А.1 Общие данные, назначение и цели разработки

Цель работы – разработать веб приложение, которое поможет связаться с компаниями, продающими воздухо-вентиляционное оборудование. Данное веб-приложение должно представлять собой интернет площадку, позволяющую объединить продавцов(компаний) и покупателей в одном информационном пространстве. Отличием данного веб-приложения является узконаправленность, так как должно применяться только в сфере по продажам воздухо-вентиляционного оборудования. Основное функциональное назначение – это возможность связи с продавцом (компанией), предоставляющая свой товар, потребитель сам может предложить цену и дальше вести переговоры. Информационная функция приложения позволяет получить сведения о компании, которая предоставляет нужный товар. Данное приложение позволяет значительно экономить рабочее время, позволяет создать честную конкуренцию и объединить компании, которые занимаются продажами вентиляционного оборудования.

Основной задачей является разработка веб-приложения, которое представляет собой интернет площадку, объединяющую компании, которые продают воздухо-вентиляционное оборудование.

Требования:

- Приложение должно включать в себя такие страницы, как главная страница, страница регистрации, страница входа, страница товара, страница добавления товара, страница управления товарами(изменение, удаление), страница обмена сообщениями, страница профиля, страница заказов, страница компании.
- На главной странице должен быть header и footer, в меню три вкладки главная, регистрация, добавить товар. В header поля email и пароль, а также кнопка входа и регистрации. В footer добавить переход на главная страница и добавить товар.
- При нажатии на добавить товар, если пользователь не осуществил вход, то должен быть редирект на окно входа.
- Содержание главной страницы: слова приветствия, список товаров, предоставленный компаниями.
- Приложение должно иметь поля для заполнения данных, которые необходимы, чтобы добавлять товар на главную страницу приложения, а именно: содержимое, количество, цвет шапки, описание, цена, миниатюра изображения, категория;

- Страница регистрация должна состоять из двух частей, первая часть предназначена для регистрации обычного пользователя(покупателя), вторая часть предназначена для регистрации компаний, предоставляющих свои услуги.
- Приложение должно иметь поля для заполнения данных о новом пользователе(регистрация): логин, email, пароль, город либо область, страна;
- Приложение должно иметь поля для заполнения данных о новой компании, предоставляющей свои услуги: логин, email, пароль, город либо область, наличие доставки;
- Поля: категория, цвет шапки должны быть в виде выпадающего списка.
- Поля для загрузки изображений;
- При наличии аккаунта, необходим переход на страницу входа. Она должна иметь поля: email, пароль.
- Список товаров представляет собой таблицу с полями: наименование компании, категория товара, описание, цена и количество.
- При нажатии на товар должен совершаться переход на страницу компании с ее товарами, которые были опубликованы.
- Наличие дополнительного окна, которое несет информационный характер: название компании, email, местоположение, индекс, а также наличие кнопки напишите нам, которая делает переход во вкладку сообщения. В этом окне необходимо реализовать три вкладки: информация, сообщения, отзывы. Элементы вкладки сообщений: место, где отображаются сообщения, поле ввода, кнопка отправления. Вкладка отзывов содержит в себе все отзывы, оставленные для этого магазина.
- На странице компании должны быть отображены все товары, которые были опубликованы компанией. У каждого товара должна быть фотография и поля: категория, описание, цена, сообщение покупателю, а также кнопка добавить в корзину.
- Дополнительное окно, на странице компании, корзина пользователя. Если пользователь не авторизован, то необходимо предложить переход на страницу входа. Возможность очистки корзины, удаления из корзины, увеличения количества продуктов и необходимо кнопка подтверждения.
- В разделе мой профиль – страница о компании должна содержать вкладки: заказы, товары, сообщения, профиль.
- Вкладка заказы содержит информацию о заказах: дата, описание, цена, количество проданного товара, и возможность запросить отзыв у покупателя. При нажатии на кнопку – появляется

окно об успешном отправлении. По нажатию на ссылку в чате, осуществляется переход на страницу с формой для оставления отзыва.

- Вкладка товары – вкладка для управления товарами: удаления и редактирование. А также носит информационный характер: дата, описание, цена, категория.

- Вкладка сообщения – вкладка для переговоров с покупателем, заинтересовавшийся в товаре. Состоит из двух частей: поле ввода и список пользователей с которыми ведутся переговоры.

- Вкладка профиль - логин, email, пароль, город либо область, наличие доставки, а также возможность загрузки логотипа компании и изображения для заголовка магазина.

- Страница добавления товара содержит поля: категория, описание, цена, сообщение покупателю, цвет шапки и возможность загрузки до четырех изображений.

- Страница изменения товара содержит поля: категория, описание, цена, сообщение покупателю, цвет шапки и изображения.

- У покупателя нет возможности добавлять товары. На странице профиль есть три вкладки: сообщения, профиль, заказы.

- Пароли должны шифроваться. Нельзя хранить пароли в открытом виде.

- Приложение должно быть понятным, удобным и простым.

## А.2 Характеристика отчетов и справочников

Для выполнения установленных задач необходимо создать следующие объекты:

Справочник «Пользователь»\* – email, пароль, email сайта, наименование, доставка, город, страна, индекс, логотип, шапка, дата создания, тип.

Справочник «Чат»\* - продавец, покупатель, отправитель, сообщение, дата отправки.

Справочник «Тип»\* - наименование.

Справочник «Отзыв»\* - пользователь, содержимое, дата создания.

Справочник «Заказ»\* - продавец, покупатель, товар, количество, сумма, дата создания.

Справочник «Категория»\* - наименование.

Справочник «Товар»\* - содержимое, количество, цвет шапки, описание, цена, изображение, галерея, категория, статус, дата создания.

Ввод данных осуществляется на двух языках –русский и английский.

## А.3 Состав и сроки работ по созданию системы

а) Разработка объектов по п. 2 настоящего ТЗ –45 рабочих дней;

б) Загрузка начальных данных - 1 день;

в) Обучение персонала заказчика – 3 дня;

#### А.4 Порядок приемки и контроля системы

- а) Тестовая эксплуатация системы;
- б) Нахождения ошибок при разработке и внедрении;
- в) Приемка системы осуществляется в виде работы тестовых данных заказчика.

#### А.5 Требования к содержанию и составу работы по подготовке приложения к вводу системы в действие

Необходимо осуществить загрузку данных в БД:

- Наименования категорий;
- Наименований типов;

#### А.6 Требования к безопасности

- а) Не разглашение персональных данных, хранящихся в бд.
- б) Выдача полного права доступа, только при наличии соответствующей должности.
- в) Выполнение всех законодательно установленных требований к хранению персональных данных.

## 2.2 Разработка модели бизнес-прецедентов

Для построения Use Case диаграммы был использован Visual Paradigm Online смотреть рисунок 2.

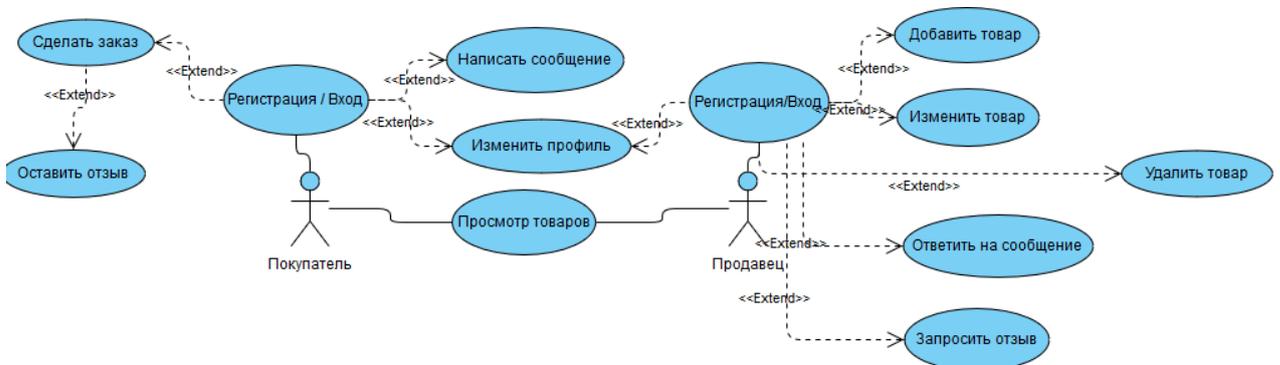


Рисунок 2 – Use Case диаграмма

Любой пользователь, который посещает сайт может просматривать товары, предложенные компаниями. Для того, чтобы добавить товар на сайт пользователю необходимо зарегистрироваться, как компания. Если же пользователь регистрируется, как покупатель, то после регистрации он сможет только сделать заказ, написать сообщение и оставить отзыв.

Функции, предназначенные для авторизованного пользователя:

- Добавление товара (компания)
- Изменение товара (компания)
- Удаление товара (компания)
- Запросить отзыв (компания)
- Написать сообщение
- Оставить отзыв
- Сделать заказ

### 2.3 Логическая и физическая модели

В логическую модель включены семь сущностей: пользователь, заказ, чат, тип, категория, отзыв и товар смотреть рисунок 2.1.

- Сущность «Пользователь» имеет атрибуты: пользователь\_ID – первичный ключ, email, пароль, email\_сайт, наименование, доставка, город, индекс, страна, логотип\_картинка, шапка\_изображение, создано\_в, тип\_ID – внешний ключ. Имеет отношение один ко многим: отзыв и чат.
- Сущность «Чат» имеет атрибуты: чат\_ID – первичный ключ, продавец\_ID – внешний ключ, покупатель\_ID – внешний ключ, отправитель\_ID – внешний ключ, сообщение, создано\_в.
- Сущность «Тип» имеет атрибуты тип\_ID – первичный ключ, наименование. Имеет отношение один ко многим: пользователь.
- Сущность «Категория» имеет атрибуты категория\_ID – первичный ключ, наименование.
- Сущность «Заказ» имеет атрибуты: заказ\_ID – первичный ключ, покупатель\_ID – внешний ключ, продавец\_ID – внешний ключ, создано\_в, товар\_ID – внешний ключ, количество, сумма.
- Сущность «Отзыв» имеет атрибуты: отзыв\_ID – первичный ключ, пользователь\_ID – внешний ключ, содержимое, создано\_в.
- Сущность «Товар» имеет атрибуты: товар\_ID – первичный ключ, содержимое, количество, цвет\_шапки, описание, цена, захват\_изображения, галерея, создано\_в, категория\_ID – внешний ключ, активный\_статус.

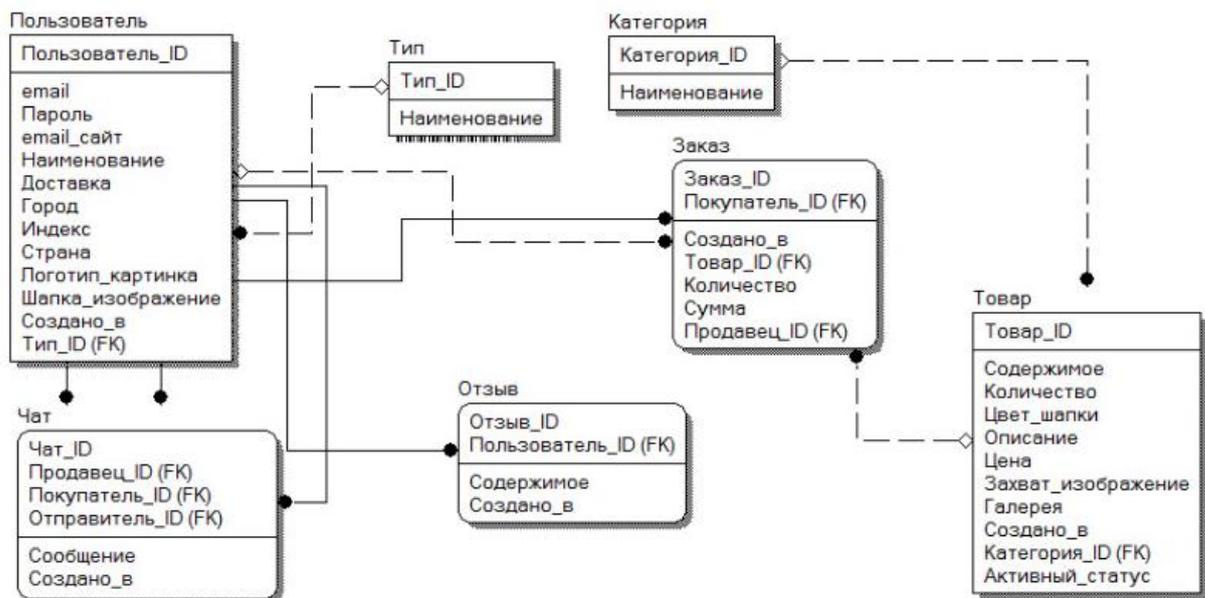


Рисунок 2.1 – Логическая модель

Физическая модель состоит из семи сущностей. В физической модели предоставляются данные о типах данных атрибутов. Физическая модель включает в себя семь основной таблиц смотреть рисунок 2.2.

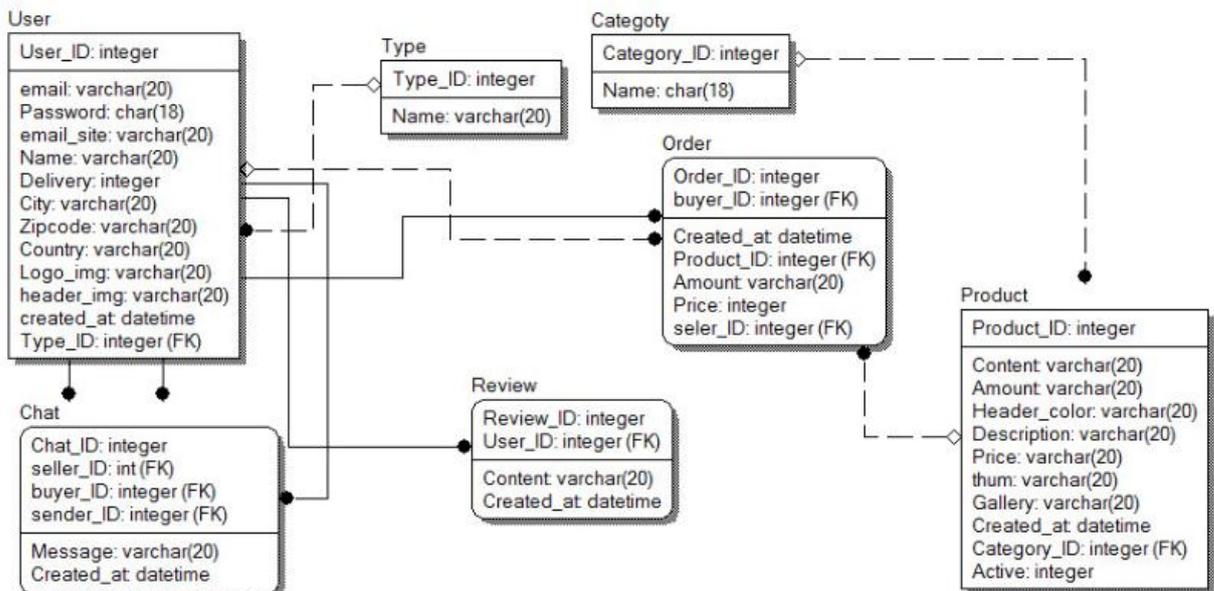


Рисунок 2.2 – Физическая модель

## 3 Технологии

### 3.1 Технологии для создания приложения

Технологии, которые применены:

1. Архитектурный паттерн MVC – шаблон, который дает возможность поделить код приложения. Он состоит из модели (model), представления (view) и контроллера (controller). Такой подход при разделении помогает упростить, большой по объему код. Иначе если записывать весь код в один длинный скрипт, то в нем будет тяжело разобраться, при этом, не сделав ошибку, тяжело будет вносить какие либо изменения. Основная идея MVC – четкое разделение ответственности за разное функционирование в приложении. MVC – не присущ какому то определенному языку программирования и не обязательно требуется использовать объектно-ориентированное программирование. MVC можно применять при проектировании разных видов приложений, например в веб-приложении, программа сначала принимает один запрос от пользователя, после идет обработка полученного запроса, и в конце выводит результат, а потом уже завершается. В случае прихода нового запроса, будет запускаться новая копия программы для данного запроса, при этом копия является независимой.

Приложение состоит из трех взаимодействующих частей смотреть рисунок 3.

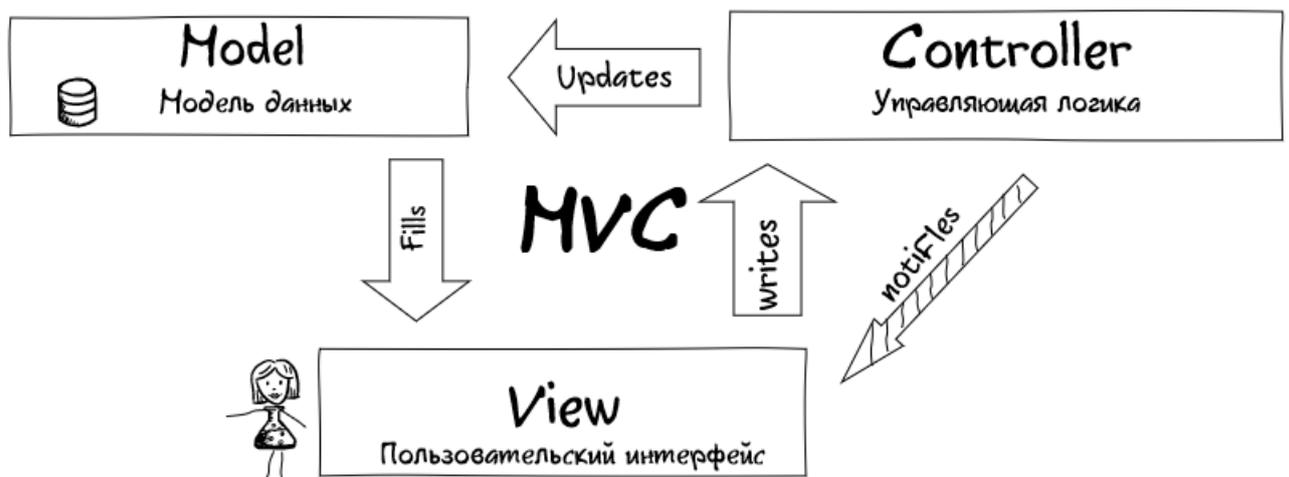


Рисунок 3 – MVC

Контроллер управляет запросами, которые приходят от пользователя, когда он выполняет какие либо действия, в виде HTTP запросов, например GET или POST. Основная функция контроллера – вызвать и управлять действием необходимых объектов и ресурсов, которые нужны для выполнения определенных действий, запрашиваемых пользователем. Контроллер выбирает соответствующую модель для конкретной задачи и выбирает необходимый вид.

Модель является ответственной за данные и правила, которые используются, когда происходит работа с данными. Она сохраняет и использует сущности из базы данных, используемые приложением. А также содержит логику, которая реализуется в приложении. Модель выдает контроллеру представление данных, которые необходимы пользователю.

Представление необходимо для отображения данных, которые выдает контроллер. Понятие представления тесно связано с шаблоном, который позволяет менять внешний вид приложения. Зачастую при создании веб приложений, реализуется непосредственно через HTML страницы.

Наряду с MVC, также используется и MVP. Это архитектурный шаблон проектирования, который необходим для построения интерфейса пользователя. MVP является производным шаблоном от MVC. Для разделения логики и отображения MVP использует: модели, виды и представители. Модель обеспечивает хранение всей бизнес логики и если есть необходимость, то может обращаться к хранилищу данных. Вид отвечает за отображения данных. Представитель обеспечивает взаимодействие между Model и View, виды обращаются за обновлениями к Представителям.

Бесспорно, одним из существенных преимуществ MVC является четкое разделение логики приложения и логики интерфейса пользователя. А также уменьшение сложности больших приложений, так как код становится более структурированным.

MVC лучше использовать, когда необходимо при каждом обращении пользователя (когда возникает событие), пересоздавать весь компонент. В MVP нет необходимости пересоздавать представление. То есть MVP позволяет использовать представление повторно. У каждого вида есть определенный интерфейс и позволяет создать механизм, который будет связывать View и Presenter. Подытоживая, можно отметить, что MVP и MVC аналогичны друг другу, но применяются, в зависимости от условий использования.

2. Laravel – бесплатный веб фреймворк общего направления с открытым исходным кодом, который определен для разработки приложений с использованием архитектуры MVC. Laravel является PHP фреймворком, из чего следует, что он написан на PHP.

Структура приложения, предложенная Laravel по умолчанию (также можно организовать приложение по своему усмотрению):

- `app` – данный каталог смело можно назвать ядром приложения, так как основная часть всего находится именно здесь. Он состоит из дополнительных подкаталогов: `Console`, `Providers` и `Http`. В каталоге `Http` располагаются контроллеры, запросы форм и посредники. Здесь, практически, располагается вся логика разрабатываемого приложения. `Console` содержит все команды `artisan`, которые необходимы для разработки приложения. В `Providers` заключены сервис провайдеры разрабатываемого приложения.
- `bootstrap` – данный каталог содержит файлы, загружающие данный фреймворк и настраивающие автозагрузку.
- `config` – данный каталог содержит в себе все конфигурационные файлы. Например: `database.php` – содержит конфигурации БД.
- `database` – данный каталог содержит в себе файлы для миграции и классы, которые позволяют заполнить данными БД.
- `public` – каталог-входная точка, так как папка содержит файл `index.php`. Этот файл является входной точкой запросов, которые поступают в приложение. Также может содержать `CSS`, `JavaScript` файлы.
- `resources` – содержит необходимые представления.
- `routes` – эта папка содержит в себе файлы, которые отвечают за маршрут.
- `storage` – данный каталог содержит сессии, кэш и шаблоны.
- `test` – папка содержит в себе файлы для автотеста.
- `vendor` – содержит файлы, которые необходимы для `Composer` и его зависимостей.

Для сравнения с Laravel можно взять `CodeInteger`, который также является PHP веб фреймворком, использующим архитектурную модель MVC.

Плюсы `CodeInteger`:

- Легок в изучении, подходит для новичков;
- Необходим, если нужно сделать быстро и реализовать достаточно простой проект;
- Использование архитектурной модели MVC;

Минусы `CodeInteger`:

- Нет шаблонизаторов;
- Использует только паттерн `Singleton`;
- Вымирающий фреймворк;

К особенностям Laravel можно отнести:

- Структура кода в виде MVC;

- Наличие достаточно информативной документации;
- Наличие консоли artisan;
- Blade шаблонизатор;
- Миграции;
- Bootstrap при установке фреймворка;
- jQuery из коробки;
- Механизмы для аутентификации и регистрации;
- Готовые правила – Laravel валидаторы;
- Поддержка ORM;
- Инструменты, которые помогают организовать очередь процессов;
- Планировщик задач – Cron;
- Достаточно простой роутинг;

Из минусов Laravel можно назвать:

- Высокий входной порог, должны быть достаточно углубленные знания;
- Сложность поиска ошибок;

3. MySQL – это популярная реляционная СУБД с открытым исходным кодом. MySQL используется при разработке средних и малых приложений. Часто используется в качестве сервера, который получает обращения от удаленных и локальных клиентов. Гибкость обеспечивается за счет того, что СУБД поддерживает большое количество типов таблиц. Основные процессы, которые происходят в MySQL:

- MySQL создает БД, в которой будет хранить и данными, которой будет управлять. Они, в свою очередь, определяют отношения каждой таблицы.
- Клиент, вводя команды Structure Query Language на MySQL, делает запрос.
- Приложение ответит на запрос и отобразится на стороне клиента.

Причины использования MySQL:

- Простота в использовании и гибкость – можно менять исходный код, как хочется.
- Высокая производительность – поддержка кластерных серверов.
- Безопасность – наличие системы управления учетными записями и системы доступа.

Использование стандарта SQL, позволяет достаточно просто производить миграцию в другие SQL подобные БД. Вдобавок есть поддержка транзакций и поддержка сложных запросов.

Наряду с MySQL, при разработке веб приложений используется MongoDB. MongoDB – это документоориентированная СУБД с открытым исходным кодом, которая не требует описания схем таблиц. Преимуществом

же MongoDB, можно назвать гибкий JSON формат документов, а также наличие достаточно простой масштабируемости.

Выбор MySQL основывается на критериях сравнения: изменение или удаление из таблицы, поиск и скорость добавления. При условии, что в таблице не более 100000 строк, был выполнен поиск, добавление внешнего ключа, соединение двух таблиц, используя LEFT JOIN, внесение изменений по ключу, удаление записи по ключу. В результате сравнения, с индексом, в среднем, MySQL в 12 раз быстрее MongoDB. Но если брать результаты без индекса, то MySQL быстрее MongoDB в 4.24 раза.

Характеристики MySQL для разработки:

- При реляционной структуре, необходимо тщательное планирование и нужен контроль;
- Разные данные можно легко использовать из различных приложений;
- Гибкость;
- Достаточно большое количество приложений 15+ лет;

Характеристики MongoDB для разработки:

- Из-за отсутствия строго формата документа, скорость разработки увеличивается;
- Понятно, как масштабировать;
- Нет необходимости в синхронизации схемы в БД и приложении;
- Простые предписанные решения;

Администрирование у MySQL достаточно гибко, в наличии много разных подходов. У MongoDB же больше ориентированность на то, что работа происходит стандартным образом, что приводит к минимизации администрирования.

4. Apache – это HTTP сервер, основным достоинством которого являются - гибкость конфигураций и надежность. Основные функциональные возможности Apache: использование внешних модулей для предоставления некоторых данных, возможность вносить изменения в сообщениях об ошибках, использование СУБД для аутентификации пользователя и т.д.

В Apache реализованы разные механизмы для разграничения доступа к данным и для обеспечения безопасности:

- Наличие модулей, которые реализуют авторизацию и аутентификацию СУБД;
- Запрет доступа, к некоторым файлам;
- Механизмы, которые обеспечивают авторизацию на основе HTTP;

В тоже время, наиболее распространенным веб-сервером с открытым исходным кодом является Nginx. Одна из его особенностей это возможность легко масштабирования на минимальном железе. На данный момент, все

больше и больше администраторов начинают использовать Nginx, так как он эффективно потребляет ресурсы, а также то, что может быть веб сервером и прокси.

Одно из самых важных отличий Apache от Nginx это то, как происходит обработка соединений и, как происходит ответ на разные виды трафиков.

Apache позволяет использовать несколько модулей, относящихся к мультипроцессингу (MPM – multi processing models). Эти модули отвечают за то, как будет обработан запрос от клиента. Так администраторы смогут устанавливать политику обработки соединений. Модули multiprocessing Apache:

- `mpm_prefork` – модуль, который позволяет на каждый запрос создать один процесс с одним потоком. При этом каждый процесс обрабатывает одно соединение за единицу времени. Если количество запросов небольшое, то `mpm` работает быстро.
- `mpm_worker` – модуль, который создает процессы, каждый процесс управляет несколькими потоками. При этом каждый поток обрабатывает одно соединение. Данный модуль намного лучше масштабируется, чем `mpm_prefork`, так как потоки более эффективны, чем процессы.
- `mpm_event` – модуль, который оптимизирован под работу повторное использование соединений или `keep alive`. Он похож на `mpm_worker`, но данный модуль выделяет самостоятельные потоки для активных соединений и для `keep alive`.

Так как Nginx появился после Apache, разработчики были осведомлены о проблемах масштабирования. Поэтому он был изначально разработан на базе асинхронных не блокирующих `event driven` алгоритмов. Nginx выстраивает процессы, которые обслуживают тысячи соединений. Этот результат был достигнут за счет того, что механизм основан на быстром цикле, который проверяет и обрабатывает события. Разделение обработки соединений и основной работы способствует тому, что каждый `worker` может заниматься своей работой и отрываться только когда происходит новое событие. Соединение, которое обрабатывается `worker`ом, попадает в `event loop` с другими соединениями. События проходят асинхронную обработку, что позволяет обрабатывать задачи – непрерывно. После того, как соединение будет закрыто, оно удаляется из цикла. Данный алгоритм позволяет Nginx масштабироваться, когда есть ограничения в ресурсах.

Также между Apache и Nginx есть различия в обработке запросов к динамическому и статическому контенту.

Apache может раздавать как статический, так и динамический контент. При статическом контенте он использует `file base` методы, производительность который в свою очередь зависит от выбранного модуля. Динамический контент Apache раздает посредством встраивания

интерпретатора необходимого языка в каждый worker. Поэтому веб сервер сам может обрабатывать запросы, обращенные к динамическому контенту.

У Nginx нет возможности самостоятельной обработки запросов к динамическому содержимому. То есть для обработки каких либо запросов, ему необходимо передать запрос, для исполнения, внешнему процессору, далее необходимо дождаться генерирования и получения запроса. И только после этого отправить результат клиенту. Это усложняет процесс настройки. Но так как здесь нет интерпретатор в каждый worker, то overhead будет только, если запрос был к динамическому контенту. Статическое содержимое будет возвращено простым способом и запросы, направленные интерпретатору будут выполняться только, когда необходимы.

5. Протокол HTTP – протокол, который расположился на прикладном уровне, для передачи данных в формате HTML. На данный момент используется для передачи данных произвольного типа. В основу HTTP легла технология клиент – сервер:

- Есть клиенты, которые инспирируют соединения и отсылают запрос;
- Есть сервера, которые ждут соединение, чтобы получить запрос. После производят необходимые манипуляции и далее возвращают результат с сообщением.

Особенностью HTTP, можно назвать, то, что есть возможность указать в ответе и запросе способ представления одного ресурса по разным параметрам: язык, формат, кодировка и т.д. Благодаря этому между клиентом и сервером может происходить обмен двоичными данными, хоть протокол и является текстовым.

Метод HTTP – это последовательность из различных символов, которая указывает на основную операцию над ресурсом. Методы HTTP:

- GET – возвращает ресурс;
- POST – создает новый ресурс;
- PUT – обновляет существующий ресурс;
- DELETE – удаляет ресурс;

Аналогичным HTTP является FTP протокол. FTP протокол – протокол для передачи файлов по сети. Оба протокола необходимы при заливании и скачивании файлов из Интернета, работают поверх TCP/IP.

Скорость передачи FTP:

- Поток, который передается, не содержит в себе мета описаний, только бинарные данные. В отдельном соединении содержатся справочные данные.
- По перекодировке данных, которые передаются, нет накладных расходов.

Скорость передачи HTTP:

- Использование постоянных соединений, которые существуют, повышает производительность ТСП, так как не тратит время для создания новых соединений.
- Автоматическое уменьшение объема передачи данных, что позволяет увеличить скорость передачи.
- Конвейерная обработка, которая позволяет быстрее делать запрос нескольких файлов с одного сервера.
- Экономия время обработки, так как в потоке передачи данных нет управляющих команд.

Для одного файла с небольшим размером и медленным соединением FTP окажется лучше, но для получения нескольких файлов – HTTP.

HTTP и FTP имеют методы аутентификации, базовую аутентификацию логин/пароль. Но в отличии от FTP, HTTP не отправляет пароль в виде обычного текста.

Наличие постоянного соединения, то есть HTTP может держать постоянное соединение в количестве равное одному, для безграничного количества передач данных. В свою очередь, FTP необходимо пересоздавать соединения для каждой новой передачи. А это сказывается на производительности.

## 4 Интерфейс приложения

### 4.1 Интерфейс

Интерфейс приложения состоит из таких страниц, как главная страница, страница регистрации, страница входа, страница добавления товара, страница компании, страница профиля, вкладка сообщения, вкладка, заказы, вкладка, товары.

1. Главная страница включает в себя приветственный заголовок, меню из трех пунктов: главная, мой профиль, добавить товар и список товаров предложенных компаниями. А также поля для входа и переход на страницу регистрации смотреть рисунок 4 и рисунок 4.1.



**Рисунок 4** – Главная страница



**Рисунок 4.1** – Поле для входа и переход на страницу регистрации

2. Страница входа включает в себя поля email и пароль смотреть рисунок 4.2.

Войти

Email\*

Пароль\*

Войти

Еще не зарегистрированы?

Регистрация

**Рисунок 4.2** – Страница входа

3. Страница профиля содержит в себе четыре либо три вкладки, в зависимости от того, кто был авторизован покупатель или продавец. Профиль продавца: заказы, товары, сообщения, мой профиль смотреть рисунок 4.3, рисунок 4.4, рисунок 4.5, рисунок 4.6. Профиль покупателя: заказы, сообщения, мой профиль смотреть рисунок 4.7, рисунок 4.8, рисунок 4.9.

ЗАКАЗЫ    ТОВАРЫ    СООБЩЕНИЯ    **МОЙ ПРОФИЛЬ**

МОЙ ПРОФИЛЬ Сохранить

Email\* alex@gmail.com

Пароль

Название компании EES @connecttoair.com

Подтвердить пароль

Город / Область Алматы

Доставка Да

Индекс 040714

Загрузите логотип компании Обзор... Файл не выбран.

Страна Казахстан

Загрузите главное изображение компании Обзор... Файл не выбран.

**Рисунок 4.3** – Вкладка Мой профиль

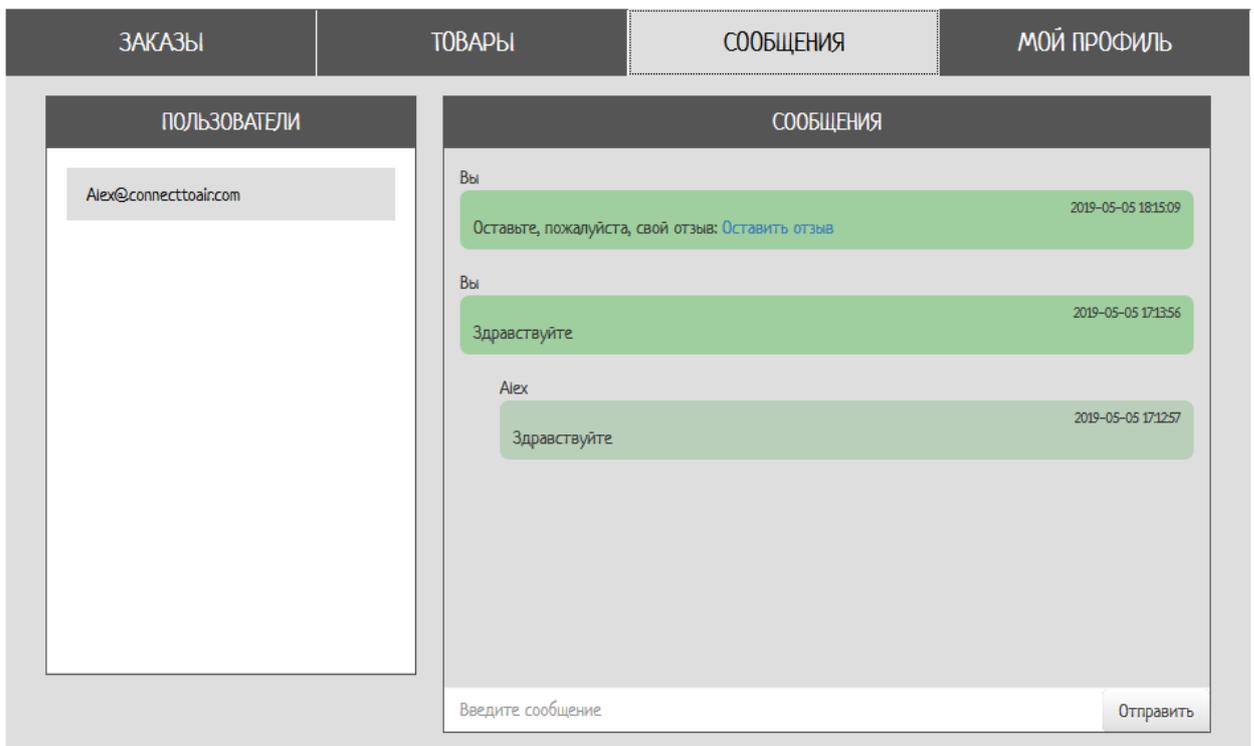


Рисунок 4.4 – Вкладка сообщения

ЗАКАЗЫ		ТОВАРЫ	СООБЩЕНИЯ	МОЙ ПРОФИЛЬ	
#	ДАТА	ОПИСАНИЕ	ЦЕНА / ЕДИНИЦА	КОЛ-ВО ПРОДАНО	ЗАПРОСИТЬ ОТЗЫВ
2	05/06/2019	Воздухораспределительное оборудование / ahjdbahadacs	9000 тг	1	<a href="#">Запрос</a>
1	05/06/2019	Воздухораспределительное оборудование / Заслонки воздушные предназначены для регулирования количества воздуха и невзрывоопасных воздушных смесей, агрессивность которых по отношению к углеродистым сталям обыкновенного качества не выше агрессивности воздуха	500000 тг	1	<a href="#">Запрос</a>

Рисунок 4.5 – Вкладка заказы

ЗАКАЗЫ		ТОВАРЫ	СООБЩЕНИЯ	МОЙ ПРОФИЛЬ			
#	ДАТА	ОПИСАНИЕ	ЦЕНА / ЕДИНИЦА	НАЛИЧИЕ	ТИП ТОВАРА	УДАЛИТЬ	ИЗМЕНИТЬ
2	2019-05-06 01:38:15	Дефлекторы устанавливают на вытяжных шахтах в системах естественной вентиляции для усиления тяги под действием ветра. Дефлекторы имеют номера от 3 до 10, которые соответствуют наружному диаметру шахты. Номер дефлектора принимается в соответствии с расчет	40000 тг	Да	Воздухораспределительное оборудование		

Рисунок 4.6 – Вкладка товары

ЗАКАЗЫ      СООБЩЕНИЯ      МОЙ ПРОФИЛЬ

**МОЙ ПРОФИЛЬ** Сохранить

**Email\*** 
**Пароль**

**Название компании** 
**Подтвердить пароль**

@connecttoair.com

**Город / Область**

**Индекс**

**Страна**

**Рисунок 4.7** – Вкладка мой профиль для покупателя

ЗАКАЗЫ      СООБЩЕНИЯ      МОЙ ПРОФИЛЬ

**ПОЛЬЗОВАТЕЛИ**

EES@connecttoair.com

**СООБЩЕНИЯ**

EES 2019-05-05 18:15:09  
Оставьте, пожалуйста, свой отзыв: [Оставить отзыв](#)

EES 2019-05-05 17:13:56  
Здравствуйте

Вы 2019-05-05 17:12:57  
Здравствуйте

Введите сообщение Отправить

**Рисунок 4.8** – Вкладка сообщения для покупателя

ЗАКАЗЫ			СООБЩЕНИЯ		МОЙ ПРОФИЛЬ	
#	ДАТА	ОПИСАНИЕ	ЦЕНА / ЕДИНИЦА	КОЛ-ВО ПРОДАНО		
1	05/06/2019	Воздухораспределительное оборудование / Заслонки воздушные предназначены для регулирования количества воздуха и невзрывоопасных воздушных смесей, агрессивность которых по отношению к углеродистым сталям обыкновенного качества не выше агрессивности воздуха	500000 тг	1		

**Рисунок 4.9** – Вкладка заказы для покупателя

4. Окно для регистрации пользователя смотреть рисунок 4.10.

### Регистрация пользователя

**Логин\***

**Email\***

**Введите пароль\***

**Город / Область**

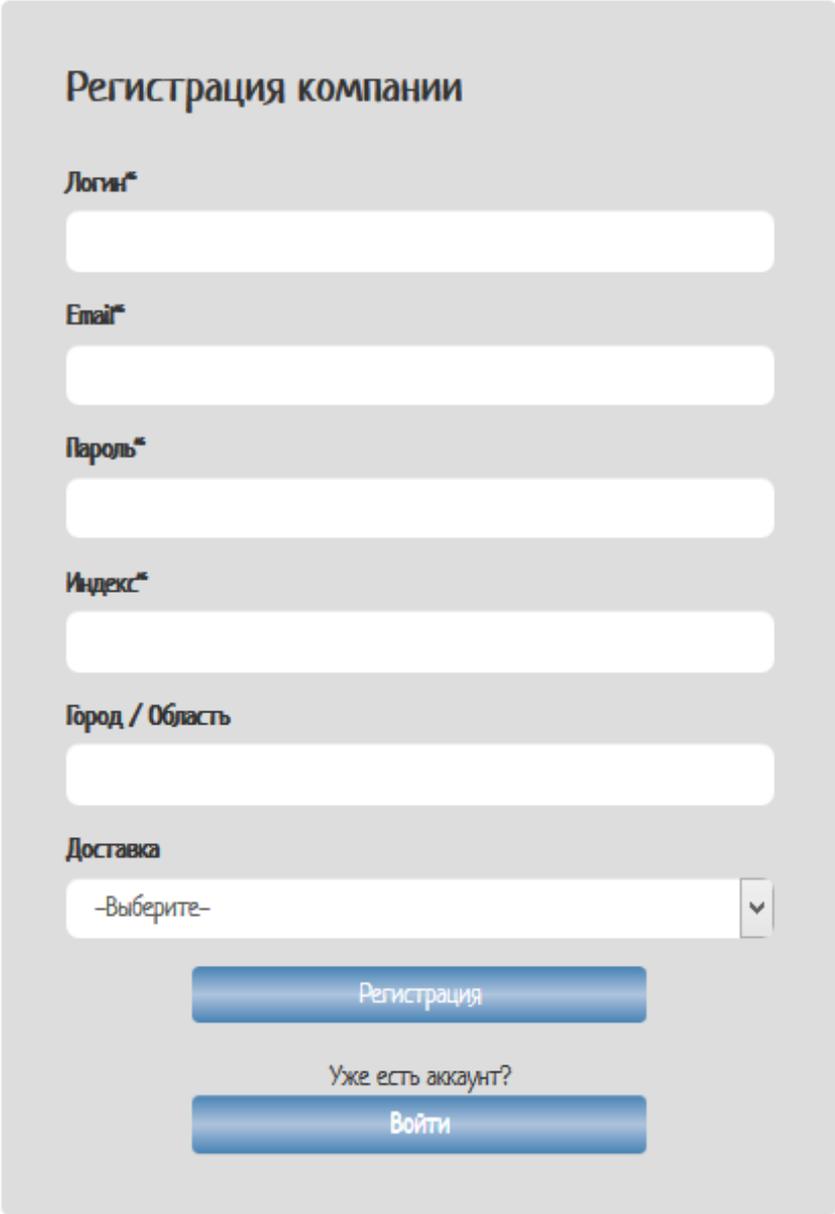
**Страна**

**Индекс\***

Уже есть аккаунт?

**Рисунок 4.10** – Форма регистрации пользователя

5. Форма регистрации для компании смотреть рисунок 4.11.



The image shows a registration form for a company. The form is titled "Регистрация компании" (Company Registration). It contains several input fields and buttons. The fields are: "Логин\*" (Login), "Email\*", "Пароль\*" (Password), "Индекс\*" (Index), "Город / Область" (City / Region), and "Доставка" (Delivery). The "Доставка" field is a dropdown menu with the text "-Выберите-" (Select). Below the fields are two buttons: "Регистрация" (Registration) and "Уже есть аккаунт? Войти" (Already have an account? Log in).

**Рисунок 4.11** – Форма регистрации для компании

6. Список товаров, предложенных компаниями смотреть рисунок 4.12.

## ТОВАРЫ ПРЕДЛОЖЕННЫЕ КОМПАНИЯМИ

### Вентиляционное оборудование

Компания	Категория товара	Описание	В наличии	Цена	Количество
EES (2+)	Воздухораспределительное оборудование	Дефлекторы устанавливаются на вытяжных шахтах в системах естественной вентиляции для усиления тяги под действием ветра. Дефлекторы имеют номера от 3 до 10, которые соответствуют наружному диаметру шахты. Номер дефлектора принимается в соответствии с расчетом	Да	40000 тт	4

Звоните нам, если у вас есть вопросы +7727234-46-67



Рисунок 4.12 – Список товаров

7. Страница компании в состав которой входят: список товаров, предложенных компанией, вкладки отзывы, информация, сообщения. Окно для покупки товара смотреть рисунок 4.13, рисунок 4.14, рисунок 4.15, рисунок 4.16, рисунок 4.17.

o Местоположение : Алматы, KZ  
o Индекс : 040714

[Напишите нам](#)

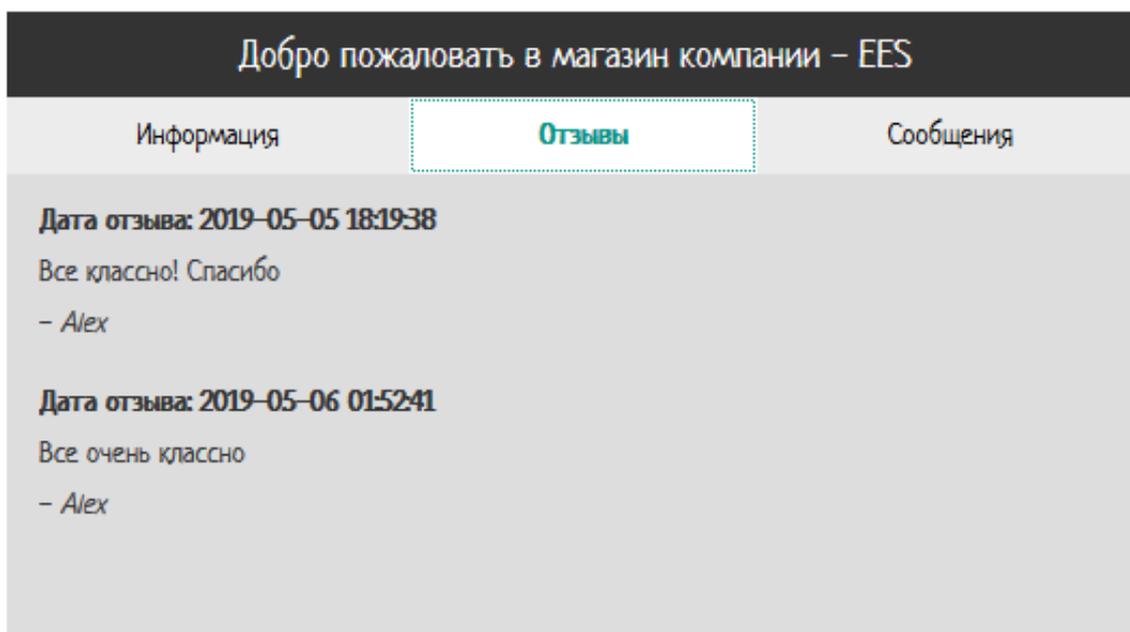
---



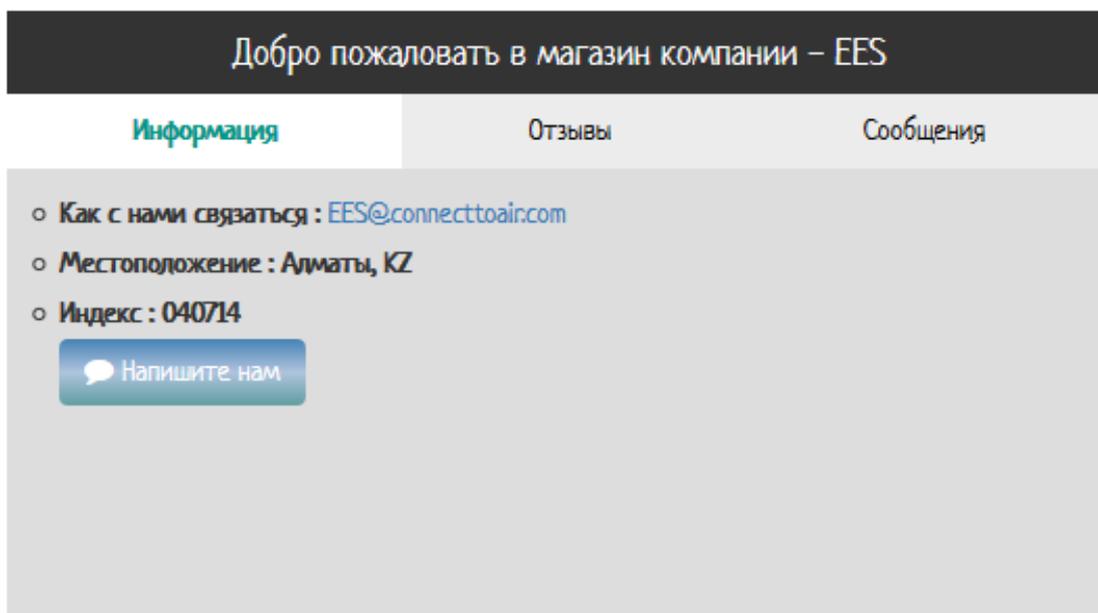
**Тип продукта:** Воздухораспределительное оборудование  
**Описание продукта:** Дефлекторы устанавливаются на вытяжных шахтах в системах естественной вентиляции для усиления тяги под действием ветра. Дефлекторы имеют номера от 3 до 10, которые соответствуют наружному диаметру шахты. Номер дефлектора принимается в соответствии с расчетом  
**Цена:** 40000 тт  
**Наименование:** Звоните нам, если у вас есть вопросы +7727234-46-67

[Добавить в корзину](#)

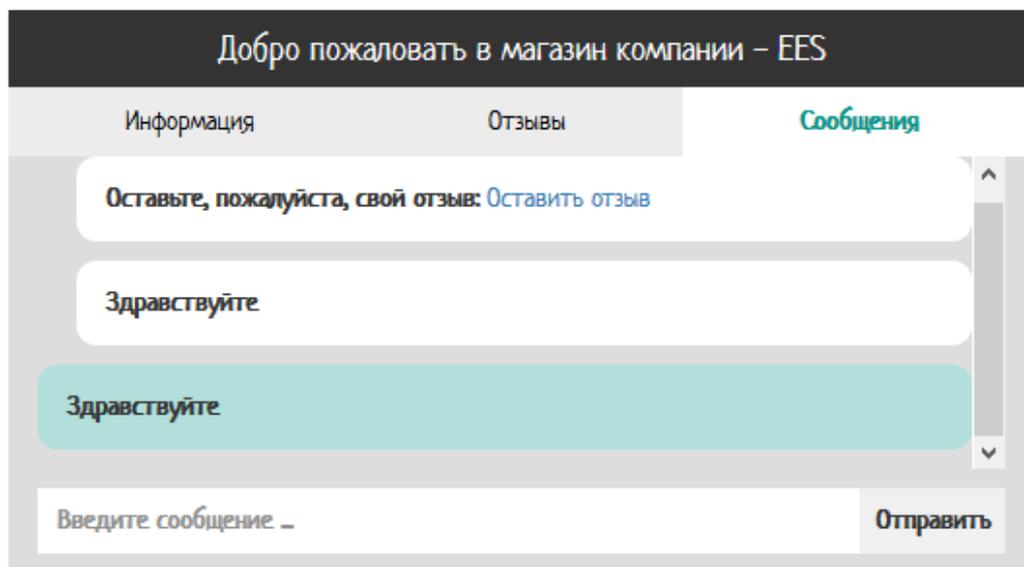
Рисунок 4.13 – Список товаров



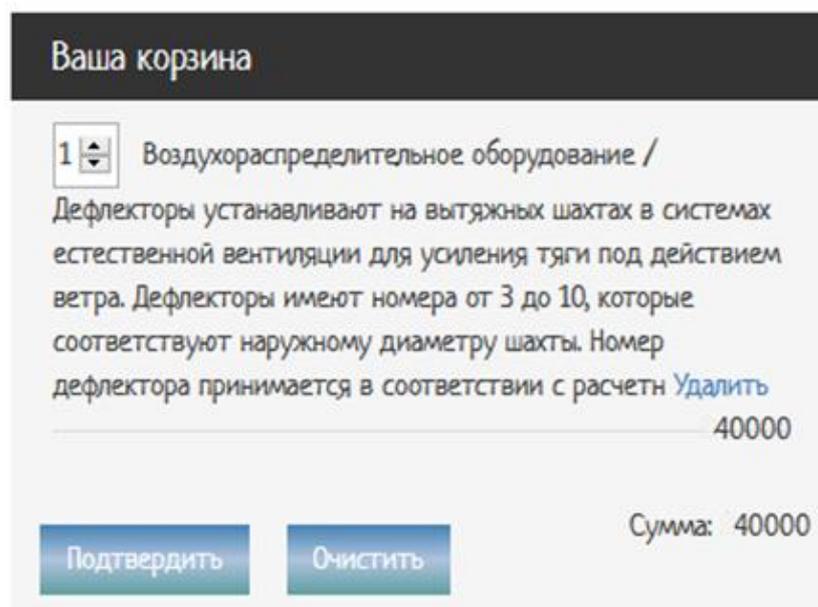
**Рисунок 4.14 – Отзывы**



**Рисунок 4.15 – Информация**



**Рисунок 4.16** – Сообщения



**Рисунок 4.17** – Корзина

8. Страница добавления и изменения товара смотреть рисунок 4.18, рисунок 4.19 и рисунок 4.20.

<b>Тип продукта</b>	Воздухораспределительное оборудование	<b>Описание</b>	Введите полное описание товара(ширина, высота, объем и т.д.)
<b>В наличии</b>	Да	<b>Цена (тг)</b>	
<b>Количество</b>			
<b>Цвет шапки</b>	Зеленый	<b>Загрузить миниатюру товара</b>	Обзор... Файл не выбран.
<b>Сообщение для покупателя</b>			
Напишите, что нибудь для покупателя ...			

Загрузить до 4 изображений

Обзор... Файл не выбран.    Обзор... Файл не выбран.    Обзор... Файл не выбран.    Обзор... Файл не выбран.

ДАЛЕЕ

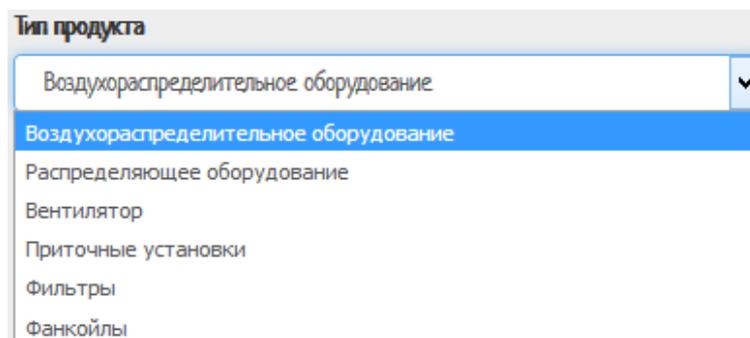
**Рисунок 4.18 – Добавление товара**

Изменить товар #2

<b>Тип продукта</b>	Воздухораспределительное оборудование	<b>Описание</b>	Дефлекторы устанавливают на вытяжных шахтах в системах естественной
<b>В наличии</b>	Да	<b>Цена (тг)</b>	40000
<b>Количество</b>	4		
<b>Цвет шапки</b>	Фиолетовый	<b>Загрузить миниатюру товара</b>	Обзор... Файл не выбран.
			
<b>Сообщение</b>			
Звоните нам, если у вас есть вопросы +7727234-46-67			

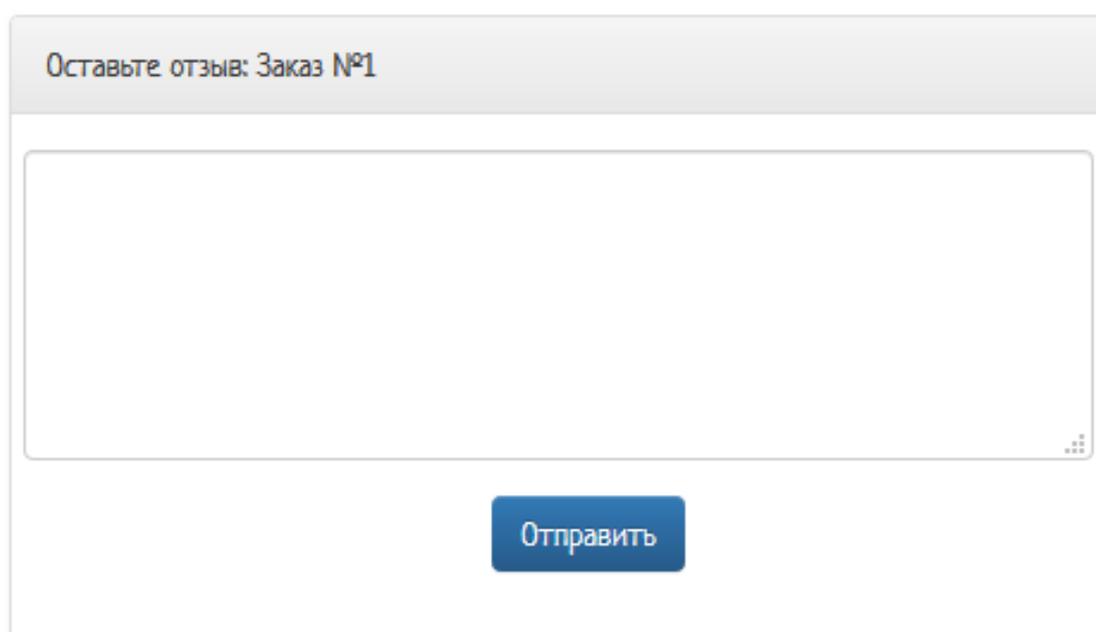
Загрузить до 4 изображений

**Рисунок 4.19 – Изменение товара**



**Рисунок 4.20** – Тип продукта

9. Страница добавления отзыва смотреть рисунок 4.21.



**Рисунок 4.21** – Страница отзыва

## ЗАКЛЮЧЕНИЕ

Опыт полученный в ходе проектирования дипломного проекта, несомненно, пригодится в дальнейшем. Так как был получен опыт работы с таким программным обеспечением, как XAMPP, phpMyAdmin, Visual Studio Code. Изучены новые, для меня, технологии – Laravel Framework, MVC паттерн.

В результате проектирования была создана простая торговая площадка, которая предназначена для продажи воздухо-вентиляционного оборудования. Ее основная функция это торговая функция, то есть связать покупателя и продавца в одном информационном и торговом пространстве. На данный момент, площадка не является полноценным приложением, так как необходимо обязательное вмешательство третьего человека – процесс не до конца автоматизирован.

Использование данной торговой площадки, позволит пользователю сэкономить время, а также деньги. Так как сайт будет содержать достаточно большое количество компаний, что создает конкуренцию. А также возможность узнавать все, что нужно не вставая с места.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Начало работы. Установка Laravel // Электронная версия на сайте <http://laravel.su/docs/5.4/installation>.
- 2 Основы работы с базами данных // Электронная версия на сайте <http://laravel.su/docs/5.4/database>.
- 3 Основы. HTTP – запросы // Электронная версия на сайте <http://laravel.su/docs/5.4/requests>.
- 4 Заметка по Laravel: биндинг моделей в маршрутах // Электронная версия на сайте <https://ruseller.com/lessons.php?rub=37&id=2795>.
- 5 Основы. Шаблоны // Электронная версия на сайте <http://laravel.su/docs/5.4/views>.
- 6 Консоль ARTISAN // Электронная версия на сайте <http://laravel.su/docs/5.4/artisan>.
- 7 Начало работы. Структура директории // Электронная версия на сайте <http://laravel.su/docs/5.4/structure>.
- 8 Безопасность. Шифрование // Электронная версия на сайте <http://laravel.su/docs/5.4/encryption>.
- 9 Концепция MVC для чайников // Электронная версия на сайте <https://ruseller.com/lessons.php?id=666>.
- 10 Проектирование баз данных с ERwin. Базовые концепции моделирования данных. Часть 1 // Электронная версия на сайте <http://www.interface.ru/home.asp?artId=4273>.
- 11 Лабораторная работа № 9. Построение логической модели данных предметной области // Электронная версия на сайте <http://khrpi-iiр.mipk.kharkiv.edu/library/technpgm/labs/lab09.html>.
- 12 Логическая и физическая модели данных // Электронная версия на сайте [https://studwood.ru/1892392/informatika/logicheskaya\\_model\\_dannyh](https://studwood.ru/1892392/informatika/logicheskaya_model_dannyh).
- 13 Основные компоненты диаграммы ERwin - сущности, атрибуты, связи. Часть 1. Понятие сущности // Электронная версия на сайте <http://www.interface.ru/home.asp?artId=4271>.
- 14 Основные компоненты диаграммы ERwin - сущности, атрибуты, связи. Часть 2. Понятие атрибута // Электронная версия на сайте <http://www.interface.ru/home.asp?artId=4272>.
- 15 BPwin и Erwin. CASE-средства для разработки информационных систем // Электронная версия на сайте <https://wm-help.net/lib/b/book/3102032702/29>.
- 16 Laravel аутентификация: установка и использование в проекте // Электронная версия на сайте <http://cccp-blog.com/laravel/laravel-auth#laravel-registratsiya-polzovatelej-i-ih-autentifikatsiya-ustanovka>.

17 TOO «EES Kazakhstan» // Электронная версия на сайте <https://eeskz.satu.kz/>

18 Зачем необходимо вентиляционное оборудование? // Электронная версия на сайте <http://pravdanews.info/zachem-neobkhodimo-ventilyatsionnoe-oborudovanie.html>

19 Вентиляция // Электронная версия на сайте <https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BD%D1%82%D0%B8%D0%BB%D1%8F%D1%86%D0%B8%D1%8F>

20 Покупатель (Purchaser) – это // Электронная версия на сайте [http://economic-definition.com/Business/Pokupatel\\_\\_Purchaser\\_\\_eto.html](http://economic-definition.com/Business/Pokupatel__Purchaser__eto.html)

21 Покупатель // Электронная версия на сайте <https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BA%D1%83%D0%BF%D0%B0%D1%82%D0%B5%D0%BB%D1%8C>

22 Продажи // Электронная версия на сайте <https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B4%D0%B0%D0%B6%D0%B8>

23 Продажа // Электронная версия на сайте [http://www.marketch.ru/marketing\\_dictionary/marketing\\_terms\\_p/sale/](http://www.marketch.ru/marketing_dictionary/marketing_terms_p/sale/)

24 Продавец // Электронная версия на сайте <https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B4%D0%B0%D0%B2%D0%B5%D1%86>

25 Продавец // Электронная версия на сайте <https://ibrain.kz/professii/prodavec>

26 Описание предметной области // Электронная версия на сайте <https://pandia.ru/text/77/246/75733.php>

27 Паттерн MVC // Электронная версия на сайте <http://artanovy.com/2011/03/arhitekturnyj-pattern-mvc/>

28 Laravel // Электронная версия на сайте <https://ru.wikipedia.org/wiki/Laravel>

29 Архитектура MVC // Электронная версия на сайте <https://github.com/codedokode/pasta/blob/master/arch/mvc.md>

30 Описание MVC - Model View Controller // Электронная версия на сайте <http://design-pattern.ru/patterns/mvc.html>

31 Laravel framework: 18 особенностей движка // Электронная версия на сайте <http://cccp-blog.com/laravel/laravel-frejmwork>

32 Apache HTTP-сервер // Электронная версия на сайте [https://ru.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://ru.wikipedia.org/wiki/Apache_HTTP_Server)

33 Apache vs Nginx: практический взгляд // Электронная версия на сайте <https://habr.com/ru/post/267721/>

34 В чём разница между использованием MVC и MVP // Электронная версия на сайте <https://habr.com/ru/post/171925/>

35 Model View Presenter // Электронная версия на сайте <https://ru.wikipedia.org/wiki/Model-View-Presenter>

- 36 Сравнение производительности MongoDB vs MySQL // Электронная версия на сайте [https://ru.bmstu.wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5\\_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8\\_MongoDB\\_vs\\_MySQL](https://ru.bmstu.wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8_MongoDB_vs_MySQL)
- 37 Самая популярная в мире база данных с открытым кодом // Электронная версия на сайте <https://www.oracle.com/ru/mysql/>
- 38 MySQL // Электронная версия на сайте <https://ru.wikipedia.org/wiki/MySQL>
- 39 Что Такое MySQL: Объяснение MySQL Для Начинающих // Электронная версия на сайте <https://www.hostinger.ru/rukovodstva/shto-takoje-mysql/>
- 40 MongoDB // Электронная версия на сайте <https://ru.wikipedia.org/wiki/MongoDB>
- 41 MySQL и MongoDB — когда и что лучше использовать // Электронная версия на сайте <https://habr.com/ru/post/322532/>
- 42 HTTP // Электронная версия на сайте <https://ru.wikipedia.org/wiki/HTTP>
- 43 HTTP против FTP // Электронная версия на сайте <https://nsoft-s.com/mychatarticles/1164-http-vs-ftp.html>
- 44 Типы HTTP-запросов и философия REST // Электронная версия на сайте <https://habr.com/ru/post/50147/>
- 45 FTP // Электронная версия на сайте <https://ru.wikipedia.org/wiki/FTP>

## Приложение А

### Текст программы

Review.php

```
<?php
```

```
namespace Ventilation;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Review extends Model
```

```
{
```

```
    public function seller()
```

```
    {
```

```
        return $this->belongsTo('Ventilation\User');
```

```
    }
```

```
}
```

User.php

```
<?php
```

```
namespace Ventilation;
```

```
use Illuminate\Foundation\Auth\User as Authenticatable;
```

```
class User extends Authenticatable
```

```
{
```

```
    public function reviews()
```

```
    {
```

```
        return $this->hasMany('Ventilation\Review');
```

```
    }
```

```
}
```

Ad.php

```
<?php
```

```
namespace Ventilation;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Ad extends Model
```

```
{
```

## Продолжение приложения А

```
public function user()
{
    return $this->belongsTo('Ventilation\User');
}
}
Chat.php
<?php

namespace Ventilation;

use Illuminate\Database\Eloquent\Model;

class Chat extends Model
{

}
Order.php
<?php

namespace Ventilation;

use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    //

    public function buyer()
    {
        return $this->belongsTo('Ventilation\User','buyer_id');
    }

    public function seller()
    {
        return $this->belongsTo('Ventilation\User','seller_id');
    }
}
AuthController.php
<?php

namespace Ventilation\Http\Controllers\Auth;
```

## Продолжение приложения А

```
use Illuminate\Http\Request;
use Auth;
use Ventilation\Events\UserRegistered;
use Ventilation\User;
use Validator;
use Ventilation\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ThrottlesLogins;
use Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers;

class AuthController extends Controller
{
    use AuthenticatesAndRegistersUsers, ThrottlesLogins;

    protected $redirectTo = '/account';

    public function __construct(Request $request)
    {
        $this->middleware('guest', ['except' => 'logout','getRegister']);
        $this->request = $request;
    }

    public function showRegistrationForm()
    {
        return view('auth.register');
    }

    public function login()
    {
        $email = $this->request->input('email');
        $password = $this->request->input('password');
        if (Auth::attempt(['email' => $email, 'password' => $password])) {
            return redirect()->intended($this->redirectTo);
        }

        return redirect()->back()
    }
}
```

## Продолжение приложения А

```
->withInput($this->request->only($this->loginUsername(), 'remember'))
->withErrors([
    $this->loginUsername() => "Такой учетной записи не существует!",
]);
}
public function register(Request $request)
{
    $rules = [
        'email' => 'required|email|unique:user,email',
        'password' => 'required|min:6',
        'name' => 'required|unique:user,name|min:3|regex:/^[pL\s\-\]+$/u',
        'zipcode' => 'required',
    ];
    $customErrorMessage = [
        'name.unique' => 'Имя уже существует.',
    ];
    if($request->input('type_ID') == '1') {
        $rules['delivery'] = 'required|boolean';
    }
    $this->validate($request,$rules,$customErrorMessage);

    $user = new User;
    $user->email = $request->input('email');
    $user->type_ID = $request->input('type_ID');
    $user->password = \Hash::make($request->input('password'));
    $user->name = $request->input('name');
    $user->zipcode = $request->input('zipcode');
    $user->city = $request->input('city');
    $user->delivery = $request->input('delivery',false);
    if($request->has('purpose')) {
        $user->purpose = json_encode($request->input('purpose'));
    }
    if($user->type_ID == '1') {
        $email_site = 'store' . $user->id . '@connecttoair.com';
    }else{
```

## Продолжение приложения А

```
$email_site = $this->incrementalHash() . '@connecttoair.com';
}
$user->email_site = $email_site;
$user->save();

\Event::fire(new UserRegistered($user));

return redirect()->back()->with('message','Спасибо за регистрацию.');
```

```
}

private function incrementalHash($len = 6){
    $charset =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvw
xyz";
    $base = strlen($charset);
    $result = "";

    $now = explode(' ', microtime())[1];
    while ($now >= $base){
        $i = $now % $base;
        $result = $charset[$i] . $result;
        $now /= $base;
    }
    return substr($result, -5);
}
}
```

AccountController.php

```
<?php
```

```
namespace Ventilation\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use Ventilation \Ad;
```

```
use Ventilation \Chat;
```

## Продолжение приложения А

```
use Ventilation \Http\Requests;
use Ventilation \Http\Controllers\Controller;
use Ventilation \Order;
use Ventilation \User;

class AccountController extends Controller
{
    public function getIndex()
    {
        $user = User::find(auth()->user()->id);
        if($user->purpose) $user->purpose = json_decode($user->purpose);

        $ads = Ad::where('user_id', $user->id)->where('active',1)-
>orderBy('created_at','desc')->get();
        $allMsg = Chat::where(auth()->user()->type . '_id', auth()->user()->id)-
>orderBy('created_at','desc')->get();
        $contactedUsers = $contactedUserIds = [];
        foreach($allMsg as $msg){
            if(auth()->user()->type_ID == '2') {
                $contactedUserIds[] = $msg->seller_id;
            }else{
                $contactedUserIds[] = $msg->buyer_id;
            }
        }
        $contactedUserIds = array_unique($contactedUserIds);
        foreach($contactedUserIds as $id) {
            $contactedUsers[] = User::find($id);
        }
        foreach($contactedUsers as &$contactedUser){
            $messages = [];
            foreach($allMsg as $msg) {
                if($msg->seller_id == $contactedUser->id || $msg->buyer_id ==
$contactedUser->id) {
                    $messages[] = $msg;
                }
            }
        }
    }
}
```

## Продолжение приложения А

```
    }
    $contactedUser->messages = $messages;
  }

  $orders = Order::where($user->type . '_id', auth()->user()->id)-
  >orderBy('created_at','desc')->get();

  return view('account', compact('user','ads','contactedUsers','allMsg','orders'));
}

public function postRequestReview(Request $request)
{
    $status = 'ok'; $msg = "";
    $orderId = $request->input('order_id');
    $order = Order::find($orderId);

    if(!is_null($order)) {
        $msg = 'Оставьте, пожалуйста, свой отзыв: <a href="'.
route('review.write') .'?order_id='. $orderId .'&token='. \Hash::make($orderId)
.'">ОСТАВИТЬ ОТЗЫВ</a>';

        $chat = new Chat();
        $chat->message = $msg;

        $chat->seller_id = $order->seller_id;
        $chat->buyer_id = $order->buyer_id;

        $chat->sender_id = $order->seller_id;
        $chat->save();
    }
    return response()->json([
        'status' => $status,
        'message' => $msg
    ]);
}
```

## Продолжение приложения А

```
public function putIndex(Request $request)
{
    $this->validate($request,[
        'password' => 'min:6|confirmed',
        'email' => 'required',
        'name' => '|required|unique:user,community_name,.auth()->user()->id
.|regex:/(^[A-Za-z0-9]+$)+/',
        'zipcode' => 'required|numeric',
        'logo_img' => 'image'
    ],[
        'name.unique' => 'Такой логин уже существует.',
        'name.required' => 'Поле логин должно быть заполнено.',
        'name.regex' => 'Формат логина не верный.',
    ]);

    $user = User::find(auth()->user()->id);
    $user->zipcode = $request->input('zipcode');
    $user->city = $request->input('city');
    $user->country = $request->input('country');
    $user->delivery = $request->input('delivery');
    $user->name = $request->input('name');
    $user->email = $request->input('email');
    if($request->has('password')) {
        $user->password = \Hash::make($request->input('password'));
    }
    $user->purpose = json_encode($request->input('purpose'));

    $destinationPath = 'uploads';
    if($request->hasFile('logo')) {
        if($user->logo_img && file_exists($user->logo_img)) {
            unlink($user->logo_img);
        }
        $fileName = $request->file('logo')->getClientOriginalName();
        $fileNameArr = explode('.', $fileName);
        $newFileName = $fileNameArr[0] . strtotime('now') . '!' .
$fileNameArr[1];4
```

## Продолжение приложения А

```
$request->file('logo')->move($destinationPath, $newFileName);

$user->logo_img = implode('/', [$destinationPath, $newFileName]);
}

if($request->hasFile('header')) {
    if($user->header_img && file_exists($user->header_img)) {
        unlink($user->header_img);
    }
    $fileName = $request->file('header')->getClientOriginalName();
    $fileNameArr = explode('.', $fileName);
    $newFileName = $fileNameArr[0] . strtotime('now') . '.' . $fileNameArr[1];
    $request->file('header')->move($destinationPath, $newFileName);

    $user->header_img = implode('/', [$destinationPath, $newFileName]);
}

if($user->save())
    return redirect()->back()->with('message','Сохранено');

return redirect()->back()->with('message','Please try again');
}
}
```

AddController.php

```
<?php
namespace Ventilation\Http\Controllers;

use Illuminate\Http\Request;

use Ventilation \Ad;
use Ventilation\Http\Requests;
use Ventilation\Http\Controllers\Controller;
use Ventilation\Repo\AdRepo;

use Ventilation\Repo\UserRepo;
```

## Продолжение приложения А

```
use Ventilation\User;

class AdController extends Controller
{
    public function __construct()
    {
        $this->middleware('seller');
    }

    public function getCreateBanner()
    {

    }

    public function getCreate(AdRepo $adRepo)
    {
        return view('post_free_ad', compact('adsLeft'));
    }

    public function getEdit(Request $request)

    {
        $this->validate($request,[
            'id' => 'required'
        ]);

        $ad = Ad::find($request->input('id'));
        $ad->gallery = !empty($ad->gallery)?json_decode($ad->gallery):[];

        return view('edit_ad', compact('ad'));
    }

    public function getDelete(Request $request)
    {
        $this->validate($request,[
```

## Продолжение приложения А

```
'id' => 'required'
    ]);

    $ad = Ad::find($request->input('id'));
    $ad->delete();

    return redirect()->back();
}

public function postUpdate(Request $request)
{
    $this->validate($request, [
        "amount" => "required",
        "header_color" => "required",
        "description" => "required",
        "category" => "required",
        "price" => "",
        "Content" => "required",
        "thumb" => 'image'
    ]);
    $ad = Ad::find($request->input('id'));

    $ad->amount = $request->input('amount');
    $ad->header_color = $request->input('header_color');
    $ad->category = $request->input('category');
    $ad->description = $request->input('description');
    $ad->price = $request->input('price');

    $destinationPath = 'uploads';
    if($request->hasFile('thumb')) {
        $fileName = $request->file('thumb')->getClientOriginalName();
        $fileNameArr = explode('.', $fileName);
        $newFileName = $fileNameArr[0] . strtotime('now') . '.' . $fileNameArr[1];
        $request->file('thumb')->move($destinationPath, $newFileName);

        $ad->thumb = implode('/', [$destinationPath, $newFileName]);
    }
}
```

## Продолжение приложения А

```
}

if($request->hasFile('gallery')) {
    $galleries = [];
    $files = $request->file('gallery');
    foreach($files as $file) {
        if(!$file) continue;
        $fileName = $file->getClientOriginalName();
        $fileNameArr = explode('.', $fileName);
        $newFileName = $fileNameArr[0] . strtotime('now') . '.' .
$fileNameArr[1];
        $file->move($destinationPath, $newFileName);
        $galleries[] = implode('/', [$destinationPath, $newFileName]);
    }
    $ad->gallery = json_encode($galleries);
}
if($ad->ad_type == 'free') {
    $ad->content = strip_tags($request->input('adContent'));
}else{
    $ad->content = $request->input('adContent');

}

if($ad->save()) {
    $msg = 'Товар изменен';
}else{
    $msg = 'Невозможно обновить';
}

return redirect()->back()->with('message',$msg);
}

private function store($request)
{
```

## Продолжение приложения А

```
$this->validate($request, [
    "amount" => "required",
    "header_color" => "required",
    "category" => "required",
    "description" => "required",
    "price" => "numeric",
    "Content" => "required",
    "thumb" => 'image'
]);
$ad = new Ad;
$ad->user_id = auth()->user()->id;
$ad->amount = $request->input('amount');
$ad->header_color = $request->input('header_color');
$ad->category = $request->input('category');
$ad->description = $request->input('description');
$ad->price = $request->input('price');
$ad->active = 1;

$destinationPath = 'uploads';
if($request->hasFile('thumb')) {
    $fileName = $request->file('thumb')->getClientOriginalName();
    $fileNameArr = explode('.', $fileName);
    $newFileName = $fileNameArr[0] . strtotime('now') . '.' .
$fileNameArr[1];
    $request->file('thumb')->move($destinationPath, $newFileName);

    $ad->thumb = implode('/', [$destinationPath, $newFileName]);
}

if($request->hasFile('gallery')) {
    $galleries = [];
    $files = $request->file('gallery');
    foreach($files as $file) {
        if(!$file) continue;
        $fileName = $file->getClientOriginalName();
        $fileNameArr = explode('.', $fileName);
```

## Продолжение приложения А

```
$newFileName = $fileNameArr[0] . strtotime('now') . '.' . $fileNameArr[1];
    $file->move($destinationPath, $newFileName);

    $galleries[] = implode('/', [$destinationPath, $newFileName]);
}
$ad->gallery = json_encode($galleries);
}

$ad->content =strip_tags($request>input('Content'));

if($ad->save()) {
    $msg = 'Товар добавлен';
}else{
    $msg = 'Невозможно создать объявление.';
}

return $msg;
}
}
```

ChatController.php  
<?php

```
namespace Ventilation\Http\Controllers;

use Illuminate\Http\Request;

use Ventilation\Chat;
use Ventilation\Http\Requests;
use Ventilation\Http\Controllers\Controller;
use Activity;
use Ventilation\User;

class ChatController extends Controller
```

## Продолжение приложения А

```
{
public function __construct(Request $request)
{
    $this->request = $request;
}
public function postStore()
{
    $this->validate($this->request,[
        'message' => 'required',
    ]);
    $chat = new Chat();
    $chat->message = $this->request->input('message');
    if($this->request->has('seller_id') && $this->request->has('buyer_id')) {
        $chat->seller_id = $this->request->input('seller_id');
        $chat->buyer_id = $this->request->input('buyer_id');
    }else{
        if($this->request->has('seller_id')) {
            $chat->seller_id = $this->request->input('seller_id');

            $chat->buyer_id = auth()->user()->id;
        }
        if($this->request->has('buyer_id')) {
            $chat->buyer_id = $this->request->input('buyer_id');
            $chat->seller_id = auth()->user()->id;
        }
    }

    $chat->sender_id = auth()->user()->id;
    $chat->save();
    $status = 'ok';

    return response()->json(compact('status'));
}
}
ProductController.php
<?php
```

## Продолжение приложения А

```
namespace Ventilation\Http\Controllers;

use Illuminate\Http\Request;
use Ventilation\Http\Requests;
use Cart;

class ProductController extends Controller
{
    public function postAddToCart(Request $request)
    {
        Cart::instance('products')->add($request->input('id'), $request->input('name'),
1,0);

        return view('partials.product_cart');
    }
    public function postRemoveFromCart(Request $request)
    {
        Cart::instance('products')->remove($request->input('rowid'));

        return view('partials.product_cart');
    }

    public function postCheckout(Request $request)
    {
        // $buyer = auth()->user();
        // Cart::instance('products')->destroy();
        // Cart::destroy();
        // return "";
    }
}
```

ReviewController.php  
<?php

Namespace Ventilation\Http\Controllers;

## Продолжение приложения А

```
use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;
use Ventrilation\Http\Requests;
use Ventrilation\Http\Controllers\Controller;
use Ventrilation\Order;
use Ventrilation\Review;

class ReviewController extends Controller
{
    public function __construct(Request $request)
    {
        $this->request = $request;
    }
    public function getWrite()
    {

        $orderId = $this->request->input('order_id');

        if(Hash::check($orderId, $token)) {
            return view('leave_review', compact('orderId'));
        }

        abort(404);
    }

    public function postStore()
    {
        $this->validate($this->request,[
            'content' => 'required',
            'order_id' => 'required|numeric'
        ]);

        $order = Order::find($this->request->input('order_id'));

        $review = new Review;
```

## Продолжение приложения А

```
$review->content = $this->request->input('content');
    $review->reviewer = $order->buyer_id->name;
    $review->user_id = $order->seller_id;

    $review->save();

    return redirect()->back()->with('message','Спасибо за отзыв');
}
}
```

SellerMiddleware.php

<?php

```
namespace Ventilation\Http\Middleware;
```

```
class SellerMiddleware
```

```
{
```

```
    public function handle($request, Closure $next)
```

```
    {
```

```
        if($request->user()->type_ID != '1') {
```

```
            return redirect('/')->with('message','Пожалуйста, Войдите или  
Зарегистрируйтесь, как продавец');
```

```
        }
```

```
        return $next($request);
```

```
    }
```

```
}
```

Routes.php

<?php

```
/*
```

```
|-----
```

```
| Routes File
```

## Продолжение приложения А

```
|-----  
|  
| Here is where you will register all of the routes in an application.  
| It's a breeze. Simply tell Laravel the URIs it should respond to  
| and give it the controller to call when that URI is requested.  
|
```

```
*/  
/*
```

```
|-----  
| Application Routes  
|-----  
|
```

```
| This route group applies the "web" middleware group to every route  
| it contains. The "web" middleware group is defined in your HTTP  
| kernel and includes session state, CSRF protection, and more.  
|
```

```
*/
```

```
Route::controller('products', 'ProductController');
```

```
Route::get('password/email', 'Auth\PasswordController@getEmail');  
Route::post('password/email', 'Auth\PasswordController@postEmail');
```

```
Route::auth();  
Route::get('empty',function(){  
    return view('errors.empty');  
});  
Route::controller('review', 'ReviewController', [  
    'postStore' => 'review.store',  
    'getWrite' => 'review.write',  
]);  
Route::controller('cart','CartController',[  
    'getIndex' => 'cart.index',
```

## Продолжение приложения А

```
'postAddToCart' => 'cart.add',  
'postClearCart' => 'cart.clear',  
'postDelete' => 'cart.delete',  
'postSendOrder' => 'cart.send.order',  
'postUpdateQty' => 'cart.update.qty',  
'postSelectAsMyGrower' => 'cart.select.as.grower',  
]);
```

```
Route::group(['middleware' => ['auth']], function () {
```

```
    Route::controller('account','AccountController',[  
        'postRequestReview' => 'account.request.review'  
    ]);
```

```
    Route::controller('chat','ChatController',[  
        'postStore' => 'chat.store'  
    ]);
```

```
    Route::controller('ad', 'AdController', [  
        'getEdit' => 'ad.edit',  
        'getDelete' => 'ad.delete',  
        'postUpdate' => 'ad.update',  
        'postDestroy' => 'ad.destroy'  
    ]);
```

```
});
```

```
});
```

```
.env
```

```
APP_ENV=local
```

```
APP_DEBUG=true
```

```
APP_KEY=base64:jg9yTTcjTk331AVY5yTxuyLE7NcKNty+nxl6QrP9kDM=
```

```
DB_HOST=localhost
```

```
DB_DATABASE=ventilation
```

```
DB_USERNAME=root
```

```
DB_PASSWORD=
```