



Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт педагогики, психологии и социологии  
Кафедра современных образовательных технологий

УТВЕРЖДАЮ  
Заведующий кафедрой  
И.А. Ковалевич  
« 16 » 06 2017 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03.19 Прикладная информатика в социальных коммуникациях  
РАЗРАБОТКА ВЕБ-САЙТА ДЛЯ ИНТЕРНЕТ-ПРОВАЙДЕРА  
«EXELLENT»

Руководитель  доцент каф. СТО, канд.пед.наук М.М Манушкина

Выпускник  Н.В.Бергер

Красноярск 2017 г

## РЕФЕРАТ

Выпускная квалификационная работа по теме «разработка веб-сайта для интернет-провайдера «Exellent»» содержит 50 страниц текстового документа, 18 рисунков, 21 использованный источник.

Целью дипломной работы является разработка веб-сайта для интернет-провайдера «Exellent»

Дипломная работа состоит из двух основных частей.

В первой части рассматриваются современные технологии, которые применяются при разработке веб-сайтов и используемые методы. Рассматриваются теоретические аспекты разработки веб-сайтов и их использование в информационных целях. Изучаются основные виды веб-сайтов.

Также представлены основные этапы разработки веб-сайта и особенности, которые необходимо учитывать при разработке дизайна для веб-сайта.

Рассматривается программное обеспечение, которое понадобится при разработке веб-сайта.

Во второй части работы описывается практическая часть создания веб-сайта, а именно разработка прототипа, создание макета и html верстка веб-сайта.

Результатом проведенной работы является полноценный веб-сайт компании «Exellent», наполненный необходимым содержанием.

## СОДЕРЖАНИЕ

Введение.....	3
1 Обзор современных технологий и принципов для создания веб-сайтов .....	5
1.1 Современные технологий, используемые для созданий веб-сайтов .....	5
1.2 Этапы создания веб-сайта .....	8
1.3 Дизайн веб-сайта .....	9
1.4 Программное обеспечение, необходимое для создания веб-сайта .....	16
1.4.1 Утилиты .....	16
1.4.2 Текстовые редакторы .....	18
1.4.3 Графические редакторы .....	18
2 Разработка веб-сайта «Exellent» .....	19
2.1 Разработка прототипа веб-сайта.....	19
2.2 Создание дизайна веб-сайта «Exellent» .....	28
2.3 Верстка веб-сайта «Exellent» .....	29
Заключение .....	36
Список использованных источников .....	37
Приложение А .....	39

## ВВЕДЕНИЕ

Интернет, как инструмент коммуникации, который не имеет территориальных границ, позволяет обмениваться различными видами информации. В последние годы интернет занимает огромное влияние на развитие российских компаний, изменяя способы представления компании перед потенциальными клиентами, а также обслуживание существующих клиентов. Количество людей, использующих интернет как важнейшее средство для получения требующихся сведений о предоставляемых услугах, значительно выросло в последнее время. Интернет помогает в росте бизнеса как крупным участникам рынка, так и маленьким предпринимателям.

Чтобы представлять свои услуги в сети интернет компании необходимо обзавестись веб-сайтом. Он будет являться как веб-сайтом визиткой, для получения клиентами контактных данных и представления, в какой сфере услуг работает данная кампания, так и работать непосредственно с действующими клиентами. Наличие собственного веб-сайта положительно сказывается на имидже кампания, клиенты относятся с большим доверием. Также веб-сайт помогает значительно увеличить продажи, являясь главным инструментом для решения различных маркетинговых задач и значительно снижает нагрузку на офисных работников, отвечая клиентам на рутинные вопросы, такие как местоположение, список услуг и т.д. Также, значительно снижается документооборот, за счет наличия на веб-сайте личного кабинета для действующих клиентов и форм обратной связи для онлайн заявок.

Из вышесказанного следует, что веб-сайт является необходимым атрибутом для современной компании и определяет тему дипломной работы – «Разработка веб-сайта для Интернет-провайдера «Exellent»»

Целью работы будет создание полностью рабочего веб-сайта компании «Exellent».

Для достижения цели необходимо решить следующие задачи:

- Сделать обзор деятельности кампания «Exellent».

- Сделать обзор современных методов разработки веб-сайтов
- Выбрать методы разработки веб-сайта.
- Создать дизайн веб-сайта.
- Осуществить верстку веб-сайта.

# **1 Обзор современных технологий и принципов для создания веб-сайтов**

## **1.1 Современные технологий, используемые для созданий веб-сайтов**

Веб-сайт – это набор документов и файлов, которые лежат на каком-то сервере (компьютере) и объединены одним доменным именем.

Доменное имя, это название веб-сайта, которое используется для его поиска в сети. Например, «mail.ru» является доменным именем, которое объединяет странички, лежащие на сервере.

Сервером может является обычный компьютер, на котором установлены определенные программы, позволяющие работать веб-сайту.

Браузер посылает определенный запрос, являющийся доменным именем веб-сайта, сервер его принимает и отправляет обратно некие документы. Это работает как сетевое копирование файлов, только по протоколу http.

Все документы веб-сайта, которые мы видим в браузере, написаны на языке HTML. HTML (HyperText Markup Language) это язык, являющийся стандартом для разработки документов, которые будут отображаться в интернет-браузерах. Ключевой смысл языка HTML в разметке текста, для чего используются специальные команды, называемые теги, графически они отображаются в угловых скобках. HTML-документы хороши тем, что для их просмотра подходит практически любой браузер. Документ, написанный на языке HTML, будет правильно отображаться на различных устройствах, имеющих различное разрешение экрана и операционную систему. Практически все документы содержат в себе стандартные элементы, такие, как титульная надпись, заголовки или списки. С помощью тэгов HTML можно обозначать данные элементы, обеспечивая веб-браузеры минимальной информацией для отображения данных элементов, сохраняя в целом общую структуру и информационную полноту документов. Браузер – это окно, через которое мы просматриваем интернет веб-сайты. То есть все странички должны быть написаны на языке HTML.

Создать документы, написанные на языке HTML, которые станут основой вашего веб-сайта, можно несколькими способами.

Основным способом является текстовый набор HTML-кода в редакторе. Это достаточно трудоемкий процесс, который подразумевает глубокое знание принципов построения документов в HTML, умение работать с каскадными таблицами стилей CSS, а также как минимум с JavaScript.

Вторым способом является использование CMS. Это движок веб-сайта, который позволяет пользователю почти автоматически создавать веб-сайты, подразумевает меньший набор знаний для создания и требуется гораздо меньшее время на разработку. Также при использовании CMS гораздо проще осуществлять дальнейшее обслуживание веб-сайта, добавление или удаление новой информации.

Также есть возможность создавать веб-сайты с помощью специальных редакторов, которые объединяются под общим названием WYSIWYG- «What You See Is What You Get», который можно перевести как «что ты видишь, то ты и получаешь» Создание веб-сайтов в этих программах напоминает работу в графических и текстовых редакторах, но на выходе мы получим не изображение, а документ на языке HTML. Этот метод позволяет достаточно быстро получить документ с достаточно сложным оформлением.

Рассмотрим достоинства и недостатки этих методов. По началу в кажется, что достоинства CMS или редакторов WYSIWYG неоспоримы, но это далеко не так. Веб-сайты, написанные вручную, хоть и требуют большего времени на создание, но зато получаются качественней.

Обозначим преимущества веб-сайтов, написанных вручную:

Созданный таким образом веб-сайт гораздо сложнее взломать, так как нет MySQL базы данных и отсутствует файл конфигурации, а именно с помощью эти файлов в первую очередь хакеры пытаются взломать веб-сайт.

Веб-сайты, написанные на HTML, если сравнивать их с CMS системами отображаются гораздо быстрее. Так получается потому, что сами

файлы получаются намного компактнее и не надо делать запросы в MySQL базу данных.

Скорость отображения веб-сайта положительно влияет на позицию выдачи при поисковых запросах. При одинаковых условиях веб-сайт, написанный на HTML будет выше.

Веб-сайт занимает меньше места, что позволяет экономить на тарифах хостинга.

HTML веб-сайт не будет содержать продублированных страничек, что положительно отображается на результатах поиска

Для описания внешнего вида элементов применяется каскадные таблицы стилей CSS.

CSS довольно элементарный формальный язык, который придумали для описания внешнего вида документов. Это говорит о том, что он довольно прост и состоит из самобытных примитивных конструкций, которые не так сложны для изучения. Самое сложное не синтаксис, не правила написания конструкций, а огромное количество CSS свойств для запоминания, которые выполняют различные задачи. Благо, все правила англоязычные с соответствующей смысловой нагрузкой. Простой перевод на наш язык дает понятие о том, что это правило делает и наоборот - при переводе того, что мы хотим добиться определенным свойством на английский язык, велика вероятность того, что мы получим правильное свойство. Это значительно упрощает запоминание CSS правил на интуитивном уровне. Например, если нужно задать фоновый цвет достаточно сделать перевод на английский, в результате чего получаем background-color (отдельные слова в CSS пишутся через дефис).

CSS довольно просто использовать в HTML документах. Его можно подключить как внешний CSS файл к документу. Для этого достаточно прописать тег `<link rel="stylesheet" href="путь/до/файла.css">` в теге `<head>`. Это наиболее распространенный способ подключения таблиц стилей к



документу, когда внешнее оформление страниц выносится в отдельный внешний CSS файл.

CSS имеет довольно простой синтаксис. Правила объявлений настолько просты, что их можно описать в одном предложении. Сначала пишется селектор, который выбирает конкретный элемент на странице, после фигурных скобок пишутся свойства со значениями после двоеточия, а сами свойства отделяются друг от друга точкой с запятой. Самое сложное в CSS объявлении - селектор.

CSS селектор - (от слова *select* - выбирать) - это конструкция, с которой начинается каждый блок объявлений, и которая служит для выборки элемента или однотипных элементов на странице для дальнейшей стилизации. Чаще всего в качестве селектора используется определенный класс тега, например:

```
1. //HTML:
2. <div class="my-class"></div>
3. //CSS:
4. .my-class {
5.
6.     background-color: #999;
7.
8. }
9.
```

Здесь селектором выступает класс *my-class* тега *div*, который получает необходимое оформление в CSS файле. В данном случае фоновый цвет - серый. Соответственно, если на странице есть несколько тегов (не только *div*) с классом *my-class*, все эти элементы получат одинаковое оформление - серый фон цвета #999.

## 1.2 Этапы создания веб-сайта

### 1. Создается модель или прототип веб-сайта

На этом этапе примерно набрасывается внешний вид веб-сайта и общая структура веб-сайта, определяется какие элементы и в каком порядке он будет содержать. Для этого могут быть использованы различные программы и онлайн сервисы, но также можно просто отобразить на листе бумаги.

## 2. Создание дизайна веб-сайта

Как правило, макет, отражающий дизайн веб-сайта, рисуется в графических редакторах, таких как Adobe Photoshop или Gimp или Adobe Experience. Это является одним из главных этапов создания веб-сайта. Именно на этом этапе окончательно формируется внешний вид веб-сайта, добавляется вся графика, уточняются и прорисовываются все детали

## 3. Верстка страниц в HTML

Макет веб-сайта, нарисованный в графическом редакторе, разбивается на отдельные графические элементы и описываются на языке HTML, также при верстке могут быть использованы такие технологии, как CSS и javascript, если необходима некая динамическая подгрузка или анимация.

## 4. Разрабатывается программная часть или используется движок

## 5. Веб-сайт размещается на хостинге.

Подготовленные HTML странички копируются на сервер, им присваивается определенное доменное имя и получают публичный доступ

### 1.3 Дизайн веб-сайта

Первое, на что обращает внимание человек, когда заходит на какой-либо веб-сайт, это его дизайн. Именно от того, насколько удачно сделан дизайн веб-сайта зависит захочет ли пользователь просматривать страницу дальше или закроет ее. Поэтому очень важно при проектировании веб-сайта хорошо понимать на какую аудиторию он рассчитан, какие потребности возникают у пользователей. Всеми этими вопросами занимается Ux/Ui. UX или User Experience (дословно: «опыт пользователя») – дисциплина, изучающая опыт взаимодействия пользователя с продуктом, восприятие и реакцию, возникающую в результате использования.

UI — это User Interface (дословно «пользовательский интерфейс») — то, как выглядит интерфейс и то, какие физические характеристики приобретает. Определяет, какого цвета будет ваше «изделие», удобно ли будет человеку попадать пальцем в кнопочки, читабельным ли будет текст и тому подобное...

UX – это понятие, которое применимо не только в веб-дизайне, но и других областях. Опыт взаимодействия – это когда специалист на заводе управляет сложной техникой посредством нажатия пары кнопок, при чем кнопка отключения значительно крупнее. UX – это когда вы срезаете путь до дома тропинкой, хотя есть красивый и чистый тротуар; это когда вы ставите стаканчик с кофе в специальные углубления на панели в салоне вашего авто; это когда педаль тормоза шире педали газа. Все это пользовательский опыт взаимодействия. Примеров может быть масса, но суть одна: UX – это способ достижения цели пользователем максимально комфортным способом. И при создании хорошего дизайна необходимо учитывать это. UX – это область, в которую входит как исследование и проектирование, так и визуализация, и вёрстка. В результате работы над UX мы должны получить рабочий прототип. Так как речь идет о веб-дизайне, таким прототипом является HTML вёрстка.

UX в веб-дизайне – это фундамент, на котором строится дизайн веб-сайта или приложения. Это проектирование интерфейса с учётом потребности людей, для которых вы создаёте дизайн веб-сайта. Кто конечный пользователь, какие цели человек преследует, используя ваш интерфейс, как помочь ему максимально быстро и комфортно получить результат? Вот основные задачи, которые решает UX.

Многие веб-дизайнеры путают веб-дизайн и визуальный дизайн. Кто-то до сих пор считает, что процесс создания дизайна веб-сайта – это, в первую очередь, создание картинок, кнопочек, иконок (UI), совершенно забывая о фундаменте (UX), о исследовании людей, для которых все это рисуется.

Какие же задачи позволяет решить UX?

Главная задача дизайнера – максимально повысить уровень удовлетворенности конечного пользователя от взаимодействия с продуктом. Под продуктом мы будем понимать любой объект взаимодействия с пользователем, будь то реальный продукт, услуга, веб-приложение или веб-сайт.

Задача веб-дизайнера – проложить мост между владельцем бизнеса и потенциальным клиентом. Необходимо понимать, что основная работа по исследованию при проектировании UX – это изучение потенциальной аудитории продукта. Самое сложно здесь – это то, что нельзя пощупать, нельзя измерить – это эмпатия веб-дизайнера или UX специалиста, способность понять желания и чувства других людей.

Right Way – исследование. Исследование – это самый первый шаг дизайнера при работе над любым проектом. Нужно чётко вывести группу пользователей, максимально узнать аудиторию, под которую вы разрабатываете интерфейс, узнать потребности этой аудитории. Конечно, основываясь только на своем личном опыте и своих ощущениях довольно сложно прийти к объективно правильному решению в интерфейсе, ведь вы не беспристрастны по отношению к результату вашего творчества и ваш личный опыт относительно конкретного продукта может быть весьма ограниченным. Тем не менее, если вы хорошо изучили продукт, для которого делаете дизайн веб-сайта, то можете сделать какие-то выводы, принять решения. Самый минимум, который вы просто обязаны сделать это изучить текущий опыт бизнеса, накидать портрет среднего потребителя продукта. Этот пункт говорит о том, что проектирование UX очень тесно связано с бизнесом.

В начале своего исследования нужно пообщаться с владельцем бизнеса, вашим заказчиком. Он является специалистом в своем продукте и у него есть опыт работы с клиентами, он знает свою аудиторию. Необходимо записать все ключевые моменты для дальнейшего анализа. Также нужно изучить опыт аналогичных продуктов.

Wireframing, каркасное моделирование

После того, как вы определили аудиторию, определили цели и ценности персон, можно приступить к моделированию каркаса интерфейса, так называемому Вайрфреймингу. Wireframe служит для распределения информации на будущих страницах по степени важности - сверху вниз. В

процессе моделирования необходимо продумать какая информация будет на странице, определить базовую форму вывода информации, но не вдаваться в детали, не заниматься визуализацией.

Для создания каркаса можно воспользоваться специальным приложением или нарисовать от руки на бумаге в клеточку или в точку, после чего отсканировать и положить в папку с материалами проекта. Отлично подходит онлайн приложение [wireframe.cc](http://wireframe.cc), так как можно в реальном времени вносить правки и согласовывать с клиентом.

Нужно проработать все страницы веб-сайта или приложения и только после того, как каркас всего веб-сайта готов, переходить к следующему шагу.

UI, визуализация после того, основательно проработан UX, создано несколько каркасов и выбрали наиболее привлекательные варианты, можно приступать к визуализации. Визуализация – это прорисовка каркасов, создание единого стиля, оформление контента. Другими словами, мы начинаем работать над UI. Чаще всего для визуализации используется Adobe Photoshop, Sketch.app, Inkscape+Gimp или другие инструменты. Для работы с векторной графикой будем использовать Inkscape – для создания иконок и другой необходимой графикой, для работы с растровой графикой - лучшее решение, это Adobe Photoshop.

В веб-дизайне есть правила хорошего тона. Разрабатывая дизайн веб-сайта, веб-дизайнер придерживается определенных негласных правил при оформлении. Межстрочный интервал, отступы, шрифты, размеры элементов - это все поддается измерению и есть откровенно плохие реализации. Поэтому можно составить свод правил, придерживаясь которых поможет не сделать плохую визуализацию вашего дизайна. Перечислим некоторые правила:

Типографика, текст, ссылки:

- Не использовать слишком большие заголовки;
- Не использовать шрифт меньше 12px;

- Не делать слишком маленький или слишком большой межстрочный интервал;
- Не растягивать буквы инструментом "Transform", шрифт должен быть естественно пропорциональным;
- Не использовать больше 3 шрифтов на странице;
- Не использовать слишком маленький контраст, не печатать светло-серым по белому или тёмно-серым по чёрному;
- Использовать интервал между символами с осторожностью если знаете, что делаете и выбранный шрифт позволяет сделать текст "воздушным" наиболее элегантно;
- Не делать слишком маленьких отступов между абзацами, заголовками и элементами, что называется «дать воздуха дизайну»;
- Не использовать верхний регистр букв без необходимости;
- Не использовать для основных текстовых блоков слишком сложный декоративный шрифт, это должен быть простой и легко читаемый шрифт какого-либо семейства Sans или Serif .
- Все ссылки, за исключением пунктов навигации, должны быть подчеркнуты. Старайтесь также оформлять ссылки, которые уже были посещены, более темным цветом, в отличие от цвета ссылок по умолчанию;
- Если иерархия веб-сайта содержит более 3-х уровней, не забывайте о так называемых хлебных крошках. Это пункты в меню, которые подскажут пользователю на каком уровне он находится.

Графика, иконки, фотографии:

- Не использовать в дизайне шаблонные фотографии. Лучше сделать самостоятельно, порекомендовать заказчику обратиться к фотографу или найти наиболее "жизненные" фотографии;
- Не использовать иконки, сделанные из фотографий;
- Все иконки должны быть выполнены в едином стиле;
- Не увеличивать фотографию больше ее оригинального размера;

- Не масштабировать графику непропорционально;
- Не применять режимы наложения слоев, отличные от обычного (Normal);
- Стараться не применять фильтры на изображения, которые должны иметь несколько состояний (обычно и при наведении, например). Все наложения, изменения - только посредством наложения нового слоя. Всё должно быть легко воспроизведено в HTML верстке. К ретуши и подготовке фото это не относится;
- Не масштабировать фотографию до конвертации в смарт объект;
- Обрезать фотографию только посредством обтравочной маски для фигуры, оригинал фотографии, как и в предыдущем пункте, должен быть сохранен в смарт объект;
- Если иконки рисуются в Photoshop, не растривать их в уменьшенном размере. Не нужно забывать, что макету еще предстоит этап верстки и все иконки необходимо будет векторизовать. Оригинальные смарт объекты с иконками должны быть достаточно крупными для качественной трассировки;
- В идеале, у вас должны быть все плоские иконки и графика в формате SVG, в отдельной папке.
- Цвет
- Не использовать чистые цвета, обязательно стараться добиться наиболее приятного оттенка;
- Не использовать более 2-х акцентных цветов на странице и не более двух темно-серых (или черного) цветов для текста. В идеале - только 1 акцентный цвет и 1 темно-серый/черный для текста.
- Стараться закрашивать акцентным цветом только те элементы, которые наиболее важны на странице, акцентировать на них внимание. Это кнопки, стрелочки, галочки важных пунктов, текстовые ссылки, информативные иконки (мелкие иконки типа "логин", "пароль", "почта" и иконки в формах акцентировать не обязательно);

Правила работы в графическом редакторе, организация работы и прочее:

- Называть слои со смыслом;
- Стараться упорядочивать смысловые блоки и составные элементы интерфейса в группы;
- Всегда создавать отдельную папку со всеми используемыми шрифтами в формате TTF или OTF. Иконочные шрифты также должны быть размещены в этой папке;
- Создавать макеты 1 в 1 (72 пикс. на дюйм). При 100% отображении макеты должны быть в таком же масштабе, в котором предполагается результат HTML верстки;
- Обязательно использовать в работе какую-либо систему сеток. Можно разработать самому, но рекомендуется использовать систему сеток Bootstrap. Ширину контента можно настраивать вручную от оригинальной ширины Bootstrap сетки, для этого можно использовать плагины для создания сеток в случае, если пользоваться Photoshop;
- Не создавать декоративные элементы, если в этом нет практического смысла. Если это имиджевый веб-сайт и нужна красивая картинка, данное правило можно опустить;
- Придерживаться смысловой визуальной иерархии.
- Использовать правило "внутреннего и внешнего", которое гласит, что расстояния между внутренними элементами блоков должны быть меньше, чем внешнее расстояние между блоками;
- Не забывать, что люди чаще всего приходят не на главную страницу веб-сайта, а на внутренние, поэтому нужно продумать информативные универсальные блоки - шапку, подвал, сайдбары (если есть). Шапка должна быть максимально информативной, но не перенасыщенной. Обязательные элементы: Логотип, название проекта, навигация. Поиск по веб-сайту и другие элементы размещаются в зависимости от проекта.
- Хорошее решение - размещение в футере развернутой навигации или карты веб-сайта со всеми потаенными местами. Нужно постараться продумать



подвал веб-сайта до мелочей, стараться не делать скудных узких подвалов с логотипом и номером телефона. Лучше делать его высоким и развернутым. Очень часто пользователи находят нужную потаенную информацию именно в подвале веб-сайта.

## **1.4 Программное обеспечение, необходимое для создания веб-сайта**

Рассмотрим программное обеспечение, которое понадобится при разработке веб-сайта.

### **1.4.1 Утилиты**

Для того чтобы автоматизировать процесс веб разработки понадобится такая утилита как gulp.

Gulp - это инструмент, который помогает автоматизировать рутинные задачи веб-разработки. Gulp предназначен для решения таких задач, как:

Создание веб-сервера и автоматическая перезагрузка страницы в браузере при сохранении кода, слежение за изменениями в файлах проекта;

Использование различных JavaScript, CSS и HTML препроцессоров (CoffeeScript, Less, Sass, Stylus, Jade и т.д.);

Минификация CSS и JS кода, а также оптимизация отдельных файлов проекта в один;

Автоматическое создание вендорных префиксов (приставок к названию CSS свойства, которые добавляют производители браузеров для нестандартных свойств) для CSS.

Управление файлами и папками в рамках проекта - создание, удаление, переименование;

Запуск и контроль выполнения внешних команд операционной системы;

Работа с изображениями - сжатие, создание спрайтов, ресайз (png, jpg, svg и др.);

Деплой (отправка на внешний сервер) проекта по FTP, SFTP, Git и т.д.

Подключение и использование в проекте безгранично большого количества Node.js и Gulp утилит, программ и плагинов.

Создание различных карт проекта и автоматизация другого ручного труда.

Можно с уверенностью сказать, что Gulp и множество утилит, написанных для него, подходят для решения практически любой задачи при разработке проекта любой сложности - от небольшого веб-сайта до крупного проекта.

Любой проект, использующий Gulp имеет в корне файл gulpfile.js, который содержит набор инструкций по управлению проектом. Сразу хочется сказать, что написание инструкций для Gulp не является программированием, хотя пишутся на языке JavaScript.

Для установки gulp необходимо установить node.js

Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера.

При создании дизайна веб-сайтов в Adobe Photoshop не обойтись без использования какой-либо системы сеток. Можно создать сетку самостоятельно, используя направляющие, либо воспользоваться уже готовым, проверенным решением - готовой сеткой Bootstrap в формате PSD. Не смотря на то, что размеры Bootstrap сетки несколько стандартизируют дизайн, данный вариант успешно используют в своей работе огромное количество веб-дизайнеров по всему миру. Кроме того, такая стандартизация позволяет максимально быстро выполнять HTML верстку созданного макета, что неминуемо приводит к ускорению работы над проектом в условиях разработки веб-сайтов на потоке или создания шаблонов.

Браузеры, такие как Opera, Google chrome, Mozilla Firefox для тестирования правильного отображения веб-сайта в различных средах.

#### **1.4.2 Текстовые редакторы**

Эти программы будут нужны для написания и редактирования текстовых документов, коими и являются HTML документы. Здесь подойдет даже самый простой редактор «Блокнот», но для удобства лучше воспользоваться специальными программами, которые предназначены для написания HTML кода. Такие программы подсвечивают теги, помогают находить ошибки в коде и лучше структурировать код. Также в них есть множество полезных функций, позволяющих ускорить процесс написания кода. Примерами таких программ являются Notepad++, Aptana Studio, sublime text.

Для наших целей отлично подойдет Sublime Text, так как у него приятный интерфейс, очень гибкая настройка и присутствует функция множественного выделения, которая очень ускоряет процесс верстки.

#### **1.4.3 Графические редакторы.**

При создании прототипа и макета веб-сайта можно воспользоваться графическими редакторами, это программы, позволяющие работать с графикой (создавать и изменять различные изображения). Также, как и текстовые редакторы, они бывают простыми и сложными. К простым в первую очередь относят Windows Paint. К сложным графическим редакторам, которые работают с растровой графикой относится Adobe Photoshop и Gimp. Для векторной графики используется Inscapе, Adobe Illustrator или CorelDraw. Также можно использовать онлайн-сервис wireframe.cc для создания прототипа веб-сайта. В нем можно набросать структуру веб-сайта, обозначить расположение ключевых элементов, их размеры, примерные цвета.

## **2 Разработка веб-сайта «Exellent»**

### **2.1 Разработка прототипа веб-сайта**

После общения с владельцем компании были определены ключевые моменты, которые должны быть отражены на веб-сайте. Так как веб-сайт в первую очередь служит для привлечения новых клиентов, то информация, которая необходима клиенту для принятия решения воспользоваться предоставленными услугами должна находиться вверху страницы. Этой информацией является краткий обзор деятельности компании, ее логотип и тарифы предоставляемых услуг. Также на первом экране страницы должны быть контактные данные кампании и форма обратной связи, где пользователь сможет оставить онлайн заявку на подключение услуг.

Для действующих клиентов компании должна быть форма входа в личный кабинет пользователя, в котором отражается текущий баланс и список подключенных услуг. Также должна быть предоставлена информация о способах оплаты и ссылках на платежные сервисы.

Также на первой странице должна быть ссылка на информацию о самой компании и ее текущих новостях.

Итак, когда мы определились с тем, что должно быть на странице, можно приступать к прототипированию веб-сайта. Для начала определимся с общей структурой веб-сайта. Это нужно, чтобы изначально понять, каким будет объем будущего веб-сайта, какие технологий будут использоваться и какие функции будет включать веб-сайт. Существует несколько основных структур веб-сайтов:

- **Линейная структура** – это когда страницы находятся в строго определенном порядке относительно друг друга и переход с одной страницы на другую строго определен. Сфера применения такой структуры довольно ограничена. Эта структура отлично подходит для различных тестов и обучающих веб-сайтов или веб-книг. Когда страницы располагаются одна за другой, можно

быть уверенным, что пользователь не упустит необходимый материал. Такая структура показана на рис. 1



Рисунок 1 – Линейная структура веб-сайта

Разновидностью выше рассмотренной структуры веб-сайта является линейная структура ресурса с ответвлениями. Она подобна дороге с большим количеством ответвлений от нее. Пользователь перемещается с одной страницы на другую в строго определенном порядке. Однако при необходимости он всегда сможет перейти на другую ветвь и без усилий вернуться назад.

Главным достоинством линейной структуры с ответвлениями является относительно несложная возможность веб-мастерам создать ее на основе обычной линейной структуры. По мере раскрутки веб-сайта в этом довольно часто возникает потребность. Контент сильно разрастается и возникает вопрос улучшения навигации. Поэтому важен такой процесс, как нарисовать структуру веб-сайта.

линейная структура с альтернативными вариантами – еще один подвид линейной структуры веб-сайта. Он отличается от линейной тем, что пользователь имеет большее количество возможностей для поиска информации, а вернее – предоставляется выбор между двумя ответвлениями. Например, разделение на веб-сайте корпоративных и частных клиентов.

Чаще всего такую структуру применяют для регистрации посетителей веб-сайта. В таком случае все пользователи начинают работу со стартовой страницы. Но потом происходит разделение - частным лицам предлагается одна форма для заполнения информации, а представители коммерческих структур заполняют совсем иную форму.

Древоподобная структура веб-сайта

Главное достоинство такой структуры – универсальность. Она прекрасно подойдет для любого вида ресурса: и для домашней веб-странички, и веб-сайта-визитки, и корпоративного веб-сайта, каталога или портала, структура веб-сайта интернет магазина. Также возможен мгновенный доступ к какому угодно разделу или странице ресурса. Модель древовидной структуры представлена на рисунке 2.

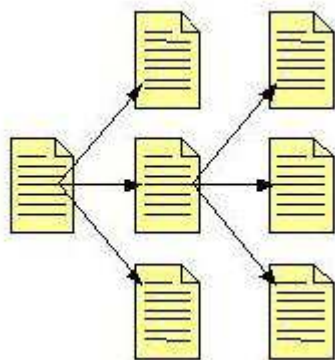


Рисунок 2 – Древовидная структура веб-сайта

#### Решетчатая структура (системы координат)

Это сложная структура ресурса, где все составляющие располагаются в отдельных ветках, по которым пользователь может легко перемещаться (см. рисунок 3). Этот вид структуры чаще всего используется для ссылок или каталогов статей. На первый взгляд, может показаться, что решетчатая структура очень удобна и приемлема для пользователей, но все-таки для обычных веб-сайтов ее лучше не применять в связи с тем, что:

- создание решетчатой структуры очень сложное, так как необходимо копаться в коде или настраивать под нее CMS. Однако использование CMS поможет вам с тем, как узнать структуру веб-сайта;
- использование решетчатой структуры легко может запутать не только пользователя, когда он будет искать информацию, но и самого веб-мастера во время размещения контента;
- решетчатую структуру лучше не применять для больших веб-сайтов, так как на них размещается очень много гиперссылок.

- легче будет воспринимать такой ресурс, если вы представите пример структуры веб-сайта в виде схемы.

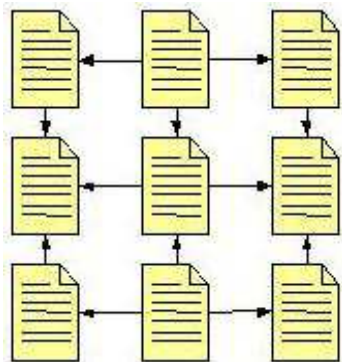


Рисунок 3 – Решетчатая структура веб-сайта

#### Смешанная структура (паутина)

Состоит из двух или более компонентов выше рассмотренных структур в одной, однако используется она редко из-за сложности реализации. Эта структура уже значительно сложнее всех рассмотренных выше. Все страницы в ней также размещаются на различных ответвлениях. Используется, как правило, только в каталогах. При этом перемещаться между ответвлениями на глубинном уровне можно с помощью ссылок на рубрики из других разделов.

Также структуру веб-сайта можно разработать самостоятельно или использовать программу для создания структуры веб-сайта (см. рисунок 4).

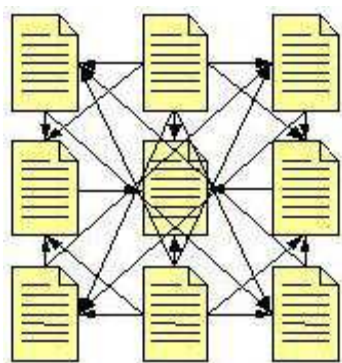


Рисунок 4 – Структура сети

Для нашего веб-сайта будет использоваться древовидная структура, так как она отлично подходит для небольшого веб-сайта компании.

#### Структура навигации по веб-сайту

Виды навигации можно выделять исходя из двух критериев: функционального и визуального.

По своим функциям система навигации делится на следующие виды:

- Языковая – навигация, отвечающая за языковой интерфейс и отображение контента на выбранном пользователем языке.
- Основная – это наиболее важные разделы веб-сайта, как правило меню.
- Глобальная – это те ссылки, которые должны быть видны с любой страницы веб-сайта, например, ссылка на главную.
- Рекламная – ссылки для привлечения посетителей на рекламные страницы веб-сайта с расположением товаров и услуг.
- Тематическая – навигация по страницам веб-сайта одной определенной тематики (рубрики).
- Текстовая – гиперссылки из текста на странице. С точки зрения юзабилити, они нужны для направления пользователя к упомянутому в тексте материалу. С точки зрения оптимизации – это грамотная перелинковка веб-сайта.
- Указательная – по-другому, справочная. Гиперссылка указывает, в какой области веб-сайта сейчас находится посетитель.
- Географическая – используется на веб-сайтах, где имеются разделы, посвященные разным странам.

По визуальному оформлению выделяют следующие виды навигации:

- Текстовая – совпадает с определением текстовой в функциональном плане. Это, пожалуй, самый древний вид навигации.
- Графическая – сейчас наиболее популярный вид навигации с графическим отображением, применяется для всех видов функциональной навигации.
- HTML-формы – помогают в экономии места с помощью выпадающих или открывающихся элементов.
- Java и Flash технологии – с помощью них можно организовать определенную реакцию на действия при наведении курсора, нажатии кнопок мыши или клавиатуры.



Для веб-сайта интернет провайдера хорошим тоном будет использование в дизайне современных технологий. Поэтому для навигации мы будем использовать плагин Mmenu. Он позволяет сделать красивое анимированное меню веб-сайта и имеет гибкую систему настроек как функционала, так и внешнего вида.

Главная страница веб-сайта является важнейших страниц на веб-сайте. Именно от впечатления на главной странице пользователь решит, нравится ему на этом веб-сайте или нет. Любая главная страница должна отвечать нескольким требованиям:

- Ясно отражать тематику веб-сайта.
- Быть понятной целевой аудитории, не содержать ничего лишнего и отвлекающего
- Содержать некое ценное предложение, которое сразу заинтересует потенциального клиента
- Она должна быть удобна и понятна пользователю, чтобы ему не приходилось долго искать элементы управления, также желательно удобное управления на мобильных устройствах
- Иметь общий дизайн

Остальные страницы веб-сайта должны быть объединены общим дизайном, на них должны быть элементы меню, ссылки на главную страницу и название ресурса. В остальных моментах страницы не должны дублироваться, то есть иметь уникальное содержание.

Структура веб-сайта интернет провайдера «Exellent» представлена на рисунке 5.

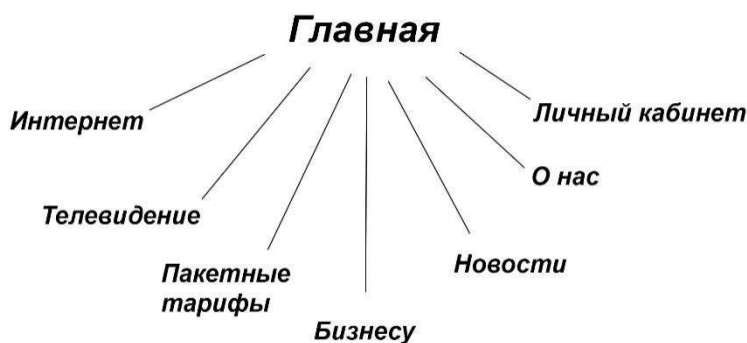


Рисунок 5 – Структура веб-сайта «Exellent»

Веб-сайт должен поддерживаться на всех популярных сегодня операционных системах – Windows, Linux, Android, IOS. Также он должен корректно отображаться во всех популярных браузерах и на всех разрешениях экранов. Для этого применяется адаптивная верстка, которая позволяет веб-сайту менять размеры и расположение своих элементов в зависимости от того, на экране какого разрешения он отображается. К тому же веб-сайты с использованием адаптивной верстки ставятся поисковыми системами на более высокие позиции.

После того как определены основные моменты внешнего вида главной страницы сайта можно приступить к ее прототипу. В онлайн сервисе wireframe.cc. Первый экран прототипа веб-сайта представлен на рисунке 6

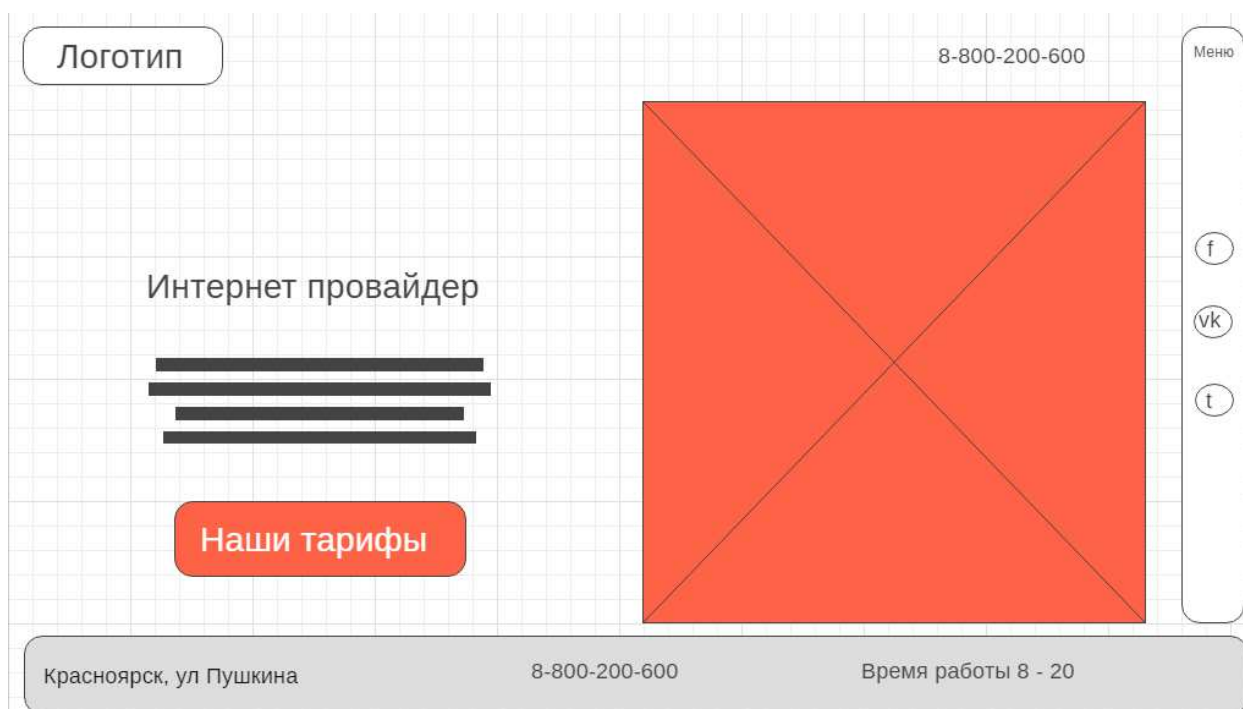


Рисунок 6 – Первый экран прототипа веб-сайта

На нем представлен логотип, контактные данные, обозначен род деятельности компании, имеется логотип и меню навигации. Это то, что пользователь увидит при открытии веб-сайта, далее он может перейти к подробному изучению информации на веб-сайте, которая будет иметь вид, как на рисунке 7.

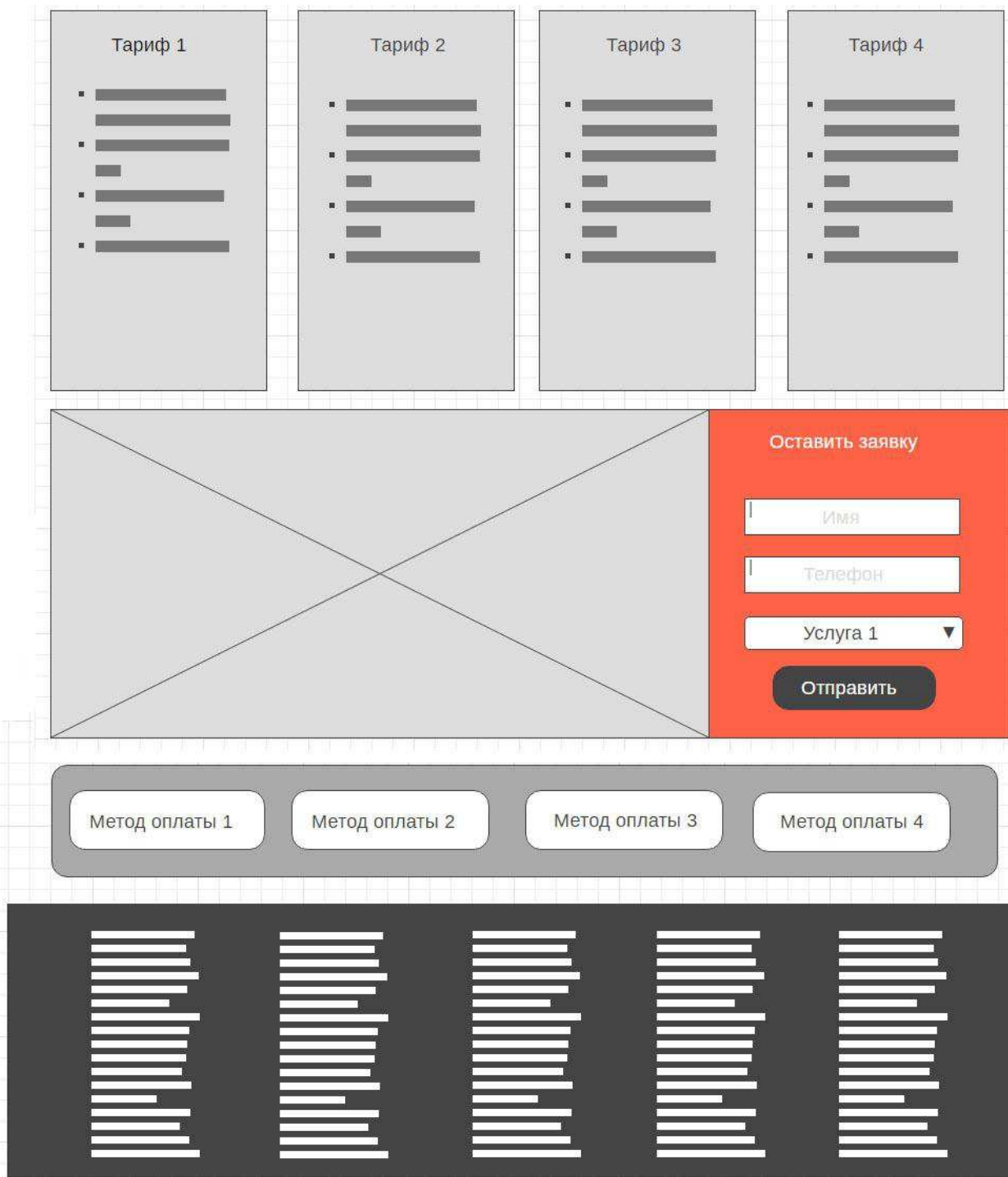


Рисунок 7 – Прототип нижней части главной страницы веб-сайта.

Здесь пользователь может подробнее ознакомиться с действующими тарифами на услуги, предоставляемые компанией, оставить заявку через форму обратной связи и ознакомиться со способами оплаты действующих услуг.

## 2.2 Создание дизайна веб-сайта «Exellent»

Как мы видим дизайн веб-сайта получился достаточно легким, на нем мало текста и представлена только самая необходимая информация. Теперь, когда готов прототип, можно приступать к более тщательной прорисовке и определению окончательного дизайна будущего веб-сайта.

Отрисовка производилась в программе adobe photoshop, так как эта программа позволяет быстро и качественно выполнить её. В дизайне используем мягкие цвета. В качестве акцентного цвета используем цвет, взятый из логотипа компании. Дизайн первого экрана представлен на рисунке 8.

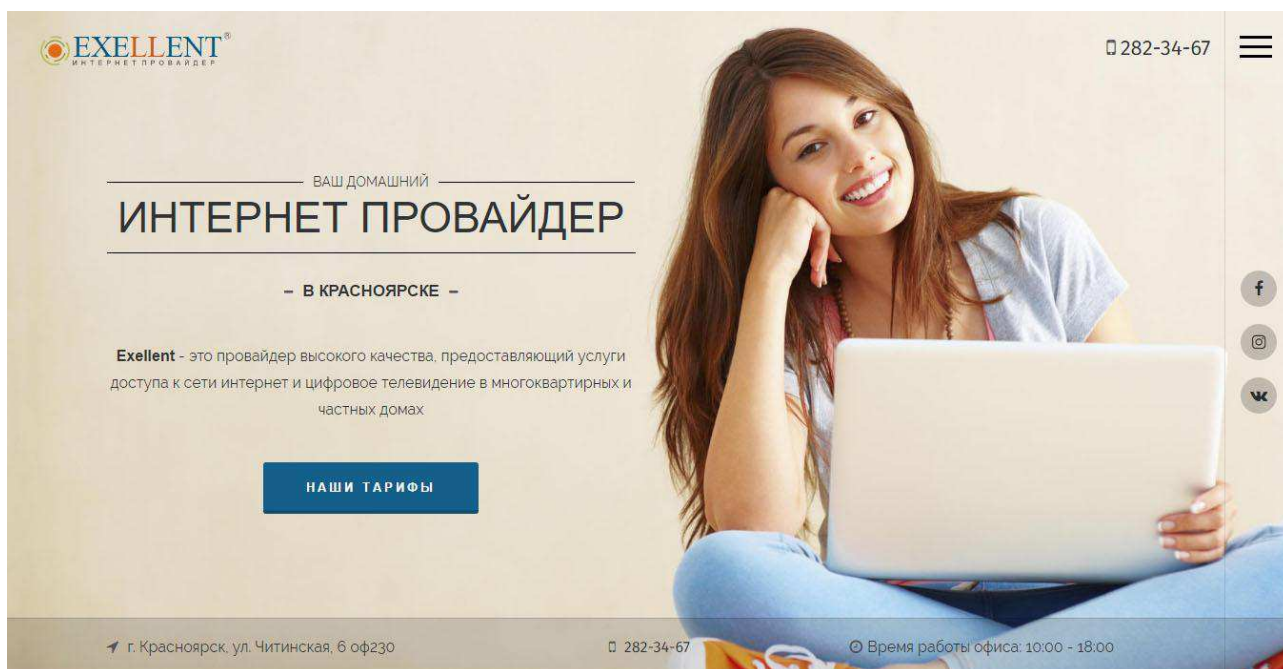


Рисунок 8 – Дизайн первого экрана веб-сайта «Exellent»

После создания макета веб-сайта в Photoshop можно приступать к вёрстке веб-сайта в HTML.

## 2.3 Верстка веб-сайта «Exellent»

Перед непосредственным созданием веб-сайта необходимо подготовить папки, в которых будут содержаться необходимые документы, такие как шрифты, изображения, таблицы стилей, библиотеки и скрипты. Также должна содержаться страничка `index.html`, которая будет являться главной страницей веб-сайта. Структура папок отображена на рисунке 9.



Рисунок 9 – Структура папок

Перед началом верстки необходимо запустить `gulp`, который создаст `web-сервер` на компьютере и позволит в режиме реально времени следить за состоянием верстки.

Для начала мы устанавливаем в файле `index.html` всю необходимую служебную информацию, подключаем стили, указываем титульную надпись `web-страницы`, также указываем путь до каскадной таблицы стилей. Все это делается в теге `<head>`.

На рисунке 10 представлена служебная информация `web-страницы`.

```

4 <head>
5
6 <meta charset="utf-8">
7
8 <title>Exellent</title>
9 <meta name="description" content="">
10
11 <meta http-equiv="X-UA-Compatible" content="IE=edge">
12 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
13
14 <meta property="og:image" content="path/to/image.jpg">
15 <link rel="shortcut icon" href="img/favicon/favicon.ico" type="image/x-icon">
16 <link rel="apple-touch-icon" href="img/favicon/apple-touch-icon.png">
17 <link rel="apple-touch-icon" sizes="72x72" href="img/favicon/apple-touch-icon-72x72.png">
18 <link rel="apple-touch-icon" sizes="114x114" href="img/favicon/apple-touch-icon-114x114.png">
19
20 <!-- Chrome, Firefox OS and Opera -->
21 <meta name="theme-color" content="#1a1a1a">
22 <!-- Windows Phone -->
23 <meta name="msapplication-navbutton-color" content="#1a1a1a">
24 <!-- iOS Safari -->
25 <meta name="apple-mobile-web-app-status-bar-style" content="#1a1a1a">
26
27 <style>body { opacity: 0; overflow-x: hidden; } html { background-color: #fff; }</style>
28
29 </head>

```

Рисунок 10 – Служебная информация web-страницы

Также необходимо подключить шрифты, которые будут использоваться при верстке и обозначить переменные, которые будут использоваться по умолчанию. Так как мы используем препроцессор стилей Sass, то все это делается в соответствующих файлах, с расширением \*. sass.

На рисунке 11 представлено подключение шрифтов. На рисунке 12 – объявление переменных.

```

1 @import "_mixins/font-face"
2
3 +font-face("raleway", "../fonts/RalewayRegular/RalewayRegular")
4 +font-face("raleway", "../fonts/RalewayRegular/Ralewaybold",bold)
5 +font-face("raleway", "../fonts/RalewayRegular/RalewayExtraBold",800)
6 +font-face("raleway", "../fonts/RalewayRegular/RalewayLight",300)
7 +font-face("raleway", "../fonts/RalewayRegular/RalewaySemiBold",600)
8 +font-face("raleway", "../fonts/RalewayRegular/RalewayThin",100)
9 +font-face("firasans", "../fonts/FiraSansRegular/FiraSansRegular")
10

```

Рисунок 11 – Подключение шрифтов

```
1 $default-font: "raleway", sans-serif
2 $accent: #14608b
3 $blue: #66c7cd
4 $red: #f86868
5 $dark: #e9e2d2
```

## Рисунок 12 – Объявление переменных

Далее ориентируясь на макет начинаем создавать блоки с нужным содержанием внутри тега `<body>` и в файле `main.sass` описывать свойства объектов. Полный текст HTML кода содержится в приложении А.

Все отдельные элементы веб-сайта удобно заключать в теги `<div>` и `<span>`.

Тег `<div>` и тег `<span>` не имеют свойств по умолчанию для внешнего отображения, они представляют собой так называемые контейнеры, поэтому к ним можно применять всевозможные CSS-стили, чтобы элементы внутри этих тегов приняли желаемый вид.

Тег `<div>` изначально был предназначен для деления веб-страницы на логические фрагменты, такие как нижний колонтитул (подвал), боковое меню и т.п. Но с появлением новой семантической разметки веб-страниц с помощью HTML5, необходимость в таком повсеместном использовании тега отпала.

В настоящее время тег `<div>` используется для группирования блочных элементов. Тег также позволяет объединять любой набор таких элементов, как заголовок, несколько абзацев, список в единственном блоке, который потом можно позиционировать на веб-странице, создавая сложную схему разметки.

Тег `<div>` следует применять только в случае, если для выделения области не подходит никакой другой элемент страницы.

Тег `<span>` применяется для выделения внутренних (inline) элементов, таких как отдельные слова и фразы, находящихся в пределах абзаца текста или заголовка.



Верстка происходит сверху вниз, поэтому, следуя макету сперва создаем логотип, который также будет являться ссылкой на главную страницу веб-сайта.

Дальнейшим важным моментом будет оформление меню, так как в этом проекте оно анимированное и работает с помощью специального плагина.

У меню навигации есть два положения:

- в положении по умолчанию меню закрыто и пользователь не видит находящихся в нем пунктов и принимает вид, как на рисунке 13.



Рисунок 13 – Закрытое положение меню навигации

Для того чтобы открыть его следует сделать одиночный щелчок мыши по символу «меню», после чего включится анимация открытия и отображаются пункты меню. Также происходит затемнение области страницы, не относящейся к меню.

- в положении открытого меню навигации видны его пункты и пользователь может выбрать нужный. После одиночного щелчка мыши по нужному происходит переход по выбранной ссылке и меню навигации закрывается. В открытом виде меню представляет такой вид, как на рисунке 14.

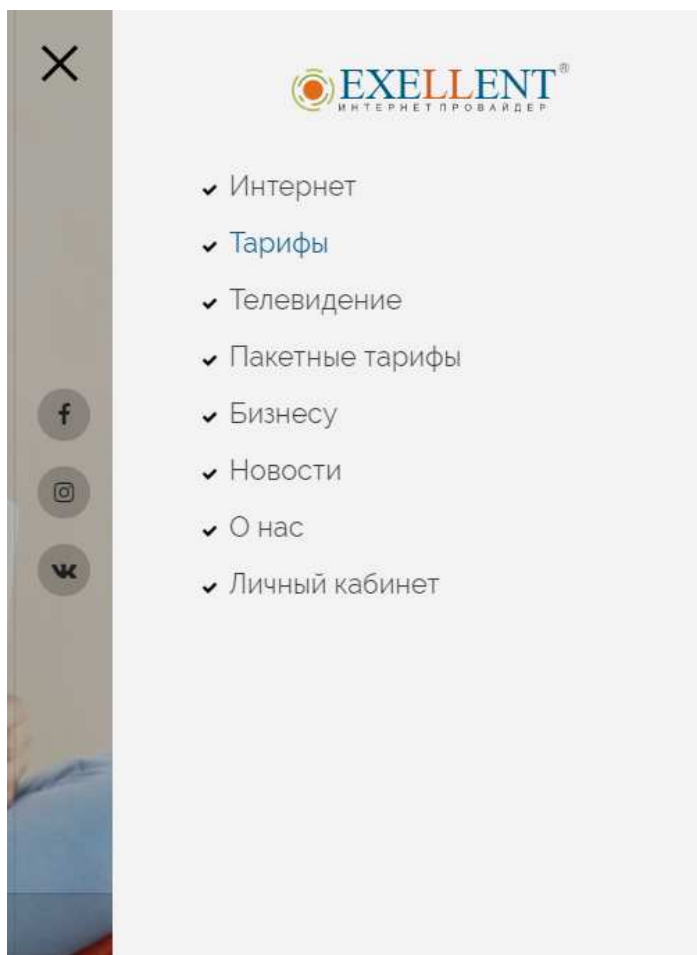


Рисунок 14 – Меню в открытом положении

Такой тип меню очень удобен, так как в закрытом положении не отвлекает пользователя и позволяет направить его внимание на более важную информацию.

Далее следует блок, в котором отображен список текущих тарифов (см. рисунок 15).

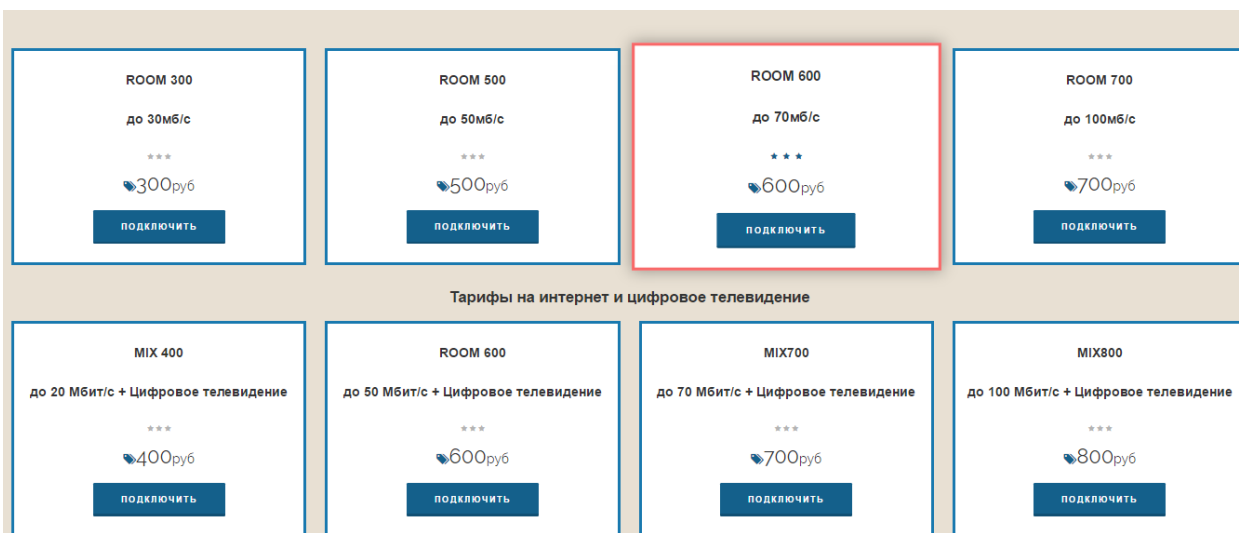


Рисунок 15 – Список тарифов

Также на веб-сайте имеется возможность оставить заявку через форму обратной связи (см. рисунок 16).

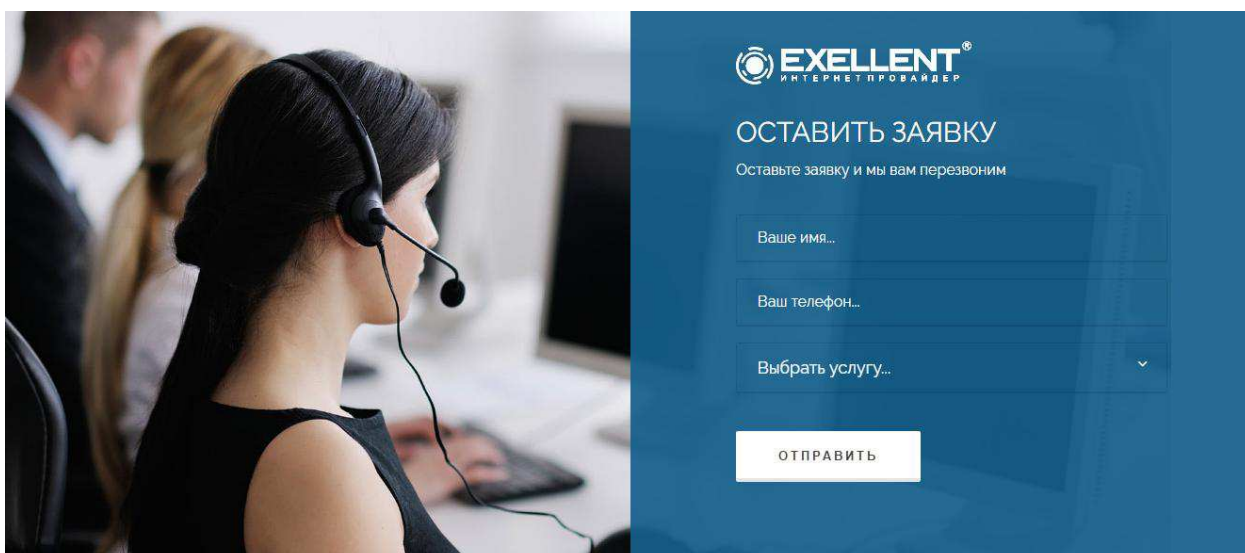


Рисунок 16 – Форма обратной связи

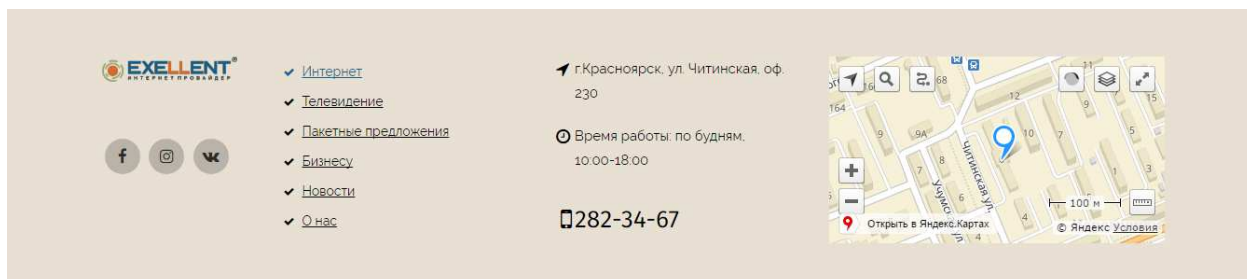
При заполнении формы, на электронную почту придет письмо с введенными данными и сотрудники офиса смогут связаться с клиентом.

Далее идет блок с перечислением платежных систем, через которые возможна онлайн оплата (рисунок 17).



## Рисунок 17 – Платежные системы

Подвал веб-сайта содержит контактную информацию, также в нем содержатся основные пункты меню и контактная информация. Также здесь располагается встроенная карта, на которой подробно можно посмотреть, где находится офис компании (рисунок 18).



## Рисунок 18 – Подвал веб-сайта

Весь дизайн веб-сайта выполнен в общем стиле и выглядит приятно и целостно. Это обеспечивается единством цветовой гаммы и небольшим разнообразием шрифтов.

## ЗАКЛЮЧЕНИЕ

В настоящее время веб-сайт – это неотъемлемый атрибут каждой уважающей себя организации. Веб-сайт как представительство компании в глобальной сети Интернет – это один из важнейших источников сведений о самой компании. А ведь от того, насколько эффективно та или иная фирма доводит свою информацию до партнеров и клиентов, зависит успех любого бизнеса. В рамках данной работы мы рассмотрели основные теоретические вопросы создания веб-сайтов – языковые и программные средства, используемые при разработке веб-ресурсов. Нами были изучены методы создания веб-сайтов, а именно метод создания веб-сайта вручную и с использованием систем управления содержимым (CMS). В конечном счете, мы разработали логическую структуру и дизайн веб-сайта для компании «Exellent». С учетом данной структуры и дизайна мы создали веб-сайт компании «Exellent» используя язык текстовой разметки HTML и каскадных таблиц стилей CSS. Нами были использованы различные плагины, с помощью которых получилось создать современный дизайн веб-сайта. Впоследствии мы опубликовали веб-сайт в сети Интернет и произвели тестирование производительности веб-сайта и его отображения в различных браузерах, исправили возникшие неполадки. Результатом проведенной работы является полноценный веб-сайт компании «Exellent», наполненный необходимым содержимым.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Первый сайт в истории интернета [Электронный ресурс]. – Режим доступа: <http://preal.ru/webhist/pervyy-sayt-v-istorii-interneta/>. – Загл. с экрана.
- 2 Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Заглавная\\_страница](http://ru.wikipedia.org/wiki/Заглавная_страница). – Загл. с экрана.
- 3 March 2013 Web Server Survey [Электронный ресурс]. – Режим доступа: <http://news.netcraft.com/archives/2013/03/01/march-2013-web-serversurvey.html>. – Загл. с экрана.
- 4 Виды, типы, разновидности сайтов [Электронный ресурс]. – Режим доступа: [http://fearit.net/fearit/index.php?option=com\\_content&view=article&id=142&Itemid=156](http://fearit.net/fearit/index.php?option=com_content&view=article&id=142&Itemid=156). – Загл. с экрана.
- 5 Классификация сайтов в современном web-e [Электронный ресурс]. – Режим доступа: <http://www.blog.astramg.ru/articles/1627/>. – Загл. с экрана.
- 6 Понятие веб-сайта. Классификация сайтов [Электронный ресурс]. – Режим доступа: <http://ikt-history.ucoz.net/publ/17-1-0-13>. – Загл. с экрана.
- 7 Классификация веб-сайтов: типы и виды интернет сайтов. Какие бывают сайты. [Электронный ресурс]. – Режим доступа: <http://www.yanajy.com/sdelay-sam/vidy-sajtov.html>. – Загл. с экрана.
- 8 Краткая история HTML и CSS [Электронный ресурс]. – Режим доступа: <http://www.greenlight5.com/index61.htm>. – Загл. с экрана.
- 9 Средства создания сайтов [Электронный ресурс]. – Режим доступа: <http://www.in-internet.narod.ru/teor/sreda.html>. – Загл. с экрана.
- 10 XHTML [Электронный ресурс]. – Режим доступа: <http://htmlweb.ru/html/xhtml.php>. – Загл. с экрана.42
- 11 Введение. DOM в примерах. [Электронный ресурс]. – Режим доступа: <http://javascript.ru/tutorial/dom/intro>. – Загл. с экрана.
- 12 Введение в JavaScript [Электронный ресурс]. – Режим доступа: <http://www.myfirstsite.ru/articles/javascript>. – Загл. с экрана.
- 13 ТЮВЕ Programming Community Index for March 2013 [Электронный ресурс]. – Режим доступа:

- <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. – Загл. с экрана.
- 14 Технология Flash В Создании Сайтов [Электронный ресурс]. – Режим доступа: <http://www.rusarticles.com/veb-dizajn-statya/tehnologiya-flashv-sozdanii-sajtov-3408023.html>. – Загл. с экрана.
- 15 PHP против ASP.Net - что лучше? [Электронный ресурс]. – Режим доступа: <http://nevlabs.ru/articles/web/php-vs-aspnet/>. – Загл. с экрана.
- 16 Методы создания сайтов [Электронный ресурс]. – Режим доступа: [http://www.web-ms.ru/publ/metody\\_sozdaniya\\_sajtov/1-1-0-98](http://www.web-ms.ru/publ/metody_sozdaniya_sajtov/1-1-0-98). – Загл. с экрана.
- 17 Что такое CMS? [Электронный ресурс]. – Режим доступа: [http://www.twl.ru/article\\_01](http://www.twl.ru/article_01). – Загл. с экрана.
- 18 Выбор системы управления сайтами [Электронный ресурс]. – Режим доступа: [http://www.cmsmagazine.ru/library/items/cms/choice\\_cms/](http://www.cmsmagazine.ru/library/items/cms/choice_cms/). – Загл. с экрана.
- 19 Общая классификация CMS [Электронный ресурс]. – Режим доступа: [http://www.solus.ru/articles\\_9.html](http://www.solus.ru/articles_9.html). – Загл. с экрана.
- 20 Что такое SEO (СЕО)? Виды оптимизации сайтов. Внешняя и внутренняя оптимизация сайта [Электронный ресурс]. – Режим доступа: <http://asbseo.ru/optimizaciya-i-prodvizhenie-bloga/chto-takoe-seo-vidyoptimizacii-sajtov.html>. – Загл. с экрана.
- 21 Внутренняя оптимизация сайта [Электронный ресурс]. – Режим доступа: [http://seokleo.ru/inside\\_optimization/](http://seokleo.ru/inside_optimization/). – Загл. с экрана

## ПРИЛОЖЕНИЕ А

### HTML код главной страницы веб-сайта «Exellent»

```
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8"
<title>Exellent</title>
<meta name="description" content="">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
<meta property="og:image" content="path/to/image.jpg">
<link rel="shortcut icon" href="img/favicon/favicon.ico" type="image/x-
icon">
<link rel="apple-touch-icon" href="img/favicon/apple-touch-icon.png">
<link rel="apple-touch-icon" sizes="72x72" href="img/favicon/apple-touch-
icon-72x72.png">
<link rel="apple-touch-icon" sizes="114x114" href="img/favicon/apple-
touch-icon-114x114.png">
<!-- Chrome, Firefox OS and Opera -->
<meta name="theme-color" content="#1a1a1a">
<!-- Windows Phone -->
<meta name="msapplication-navbutton-color" content="#1a1a1a">
<!-- iOS Safari -->
<meta name="apple-mobile-web-app-status-bar-style" content="#1a1a1a">
<style>body { opacity: 0; overflow-x: hidden; } html { background-color:
#fff; }</style>
</head>
<body class="ishome">
<div id="my-page">
```



```

    <div id="my-header">
      <header class="site-header" style="background-image: url(img/top-
bg.png);">
        <div class="top-line">
          <a href="#" class="logo"></a>
          <div class="phone hidden-xs"><i class="fa fa-mobile"></i>282-34-
67</div>
          <a href="#my-menu" class="hamburger hamburger--emphatic">
            <span class="hamburger-box"><span class="hamburger-
inner"></span></span>
            </a>
          <nav id="my-menu">
            <ul>
              <li><a href="#">Интернет</a></li>
              <li class="active"><a href="#">Тарифы</a></li>
              <li><a href="#">Телевидение</a></li>
              <li><a href="#">Пакетные тарифы</a></li>
              <li><a href="#">Бизнесу</a></li>
              <li><a href="#">Новости</a></li>
              <li><a href="#">О нас</a></li>
              <li><a href="#">Личный кабинет</a></li>
            </ul>
          </nav>
        </div>
        <div class="header-flex">
          <div class="flex-center">
            <div class="container">
              <div class="row">
                <div class="col-sm-6 col-xs-8">
                  <div class="header-composition">

```

```
<div class="welcome">
<p><span>Ваш домашний</span></p>
h1 class="h1">Интернет провайдер</h1>
<strong><i class="fa fa-minus"></i>в Красноярске<i class="fa fa-
minus"></i></strong>
</div>
<p><strong>Exellent</strong> - это провайдер высокого качества,
предоставляющий услуги доступа
к сети интернет и цифровое телевидение в многоквартирных и частных
домах</p>
<a href="#" class="button">Наши тарифы</a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="header-social">
<nav class="soc">
<ul>
<li><a href="#"><i class="fa fa-facebook"></i></a></li>
<li><a href="#"><i class="fa fa-instagram"></i></a></li>
<li><a href="#"><i class="fa fa-vk"></i></a></li>
</ul>
</nav>
</div>
<div class="header-contacts">
<div class="container">
<div class="row">
<div class="col-md-4 col-sm-6">
```

<i class="fa fa-location-arrow"></i> г. Красноярск, ул. Читинская, б  
оф230

</div>

<div class="col-md-4 col-sm-6">

<div class="phone-h"><div class="i fa fa-mobile"></div> 282-34-67</div>

</div>

<div class="col-md-4 hidden-sm hidden-xs">

<div class="i fa fa-clock-o"></div>Время работы офиса: 10:00 - 18:00

</div>

</div>

</div>

</div>

</header>

</div>

<div id="my-content"></div>

<section class="sect-tarif">

<h1>Тарифы на интернет</h1>

<container class="row">

<div class="col-md-3 col-sm-6">

<div class="tarif-item">

<h4>ROOM 300</h4>

<!-- <div class="tarif-item-image" style="background-image:  
url(img/speedometer30.png);"></div> -->

<h4>до 30мб/с</h4>

<div class="tarif-item-stars">

<i class="fa fa-star"></i>

<i class="fa fa-star"></i>

<i class="fa fa-star"></i>

</div>

<div class="tarif-item-price">

```

<i class="fa fa-tags"></i><span>300</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>ROOM 500</h4>
<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer50.png);"></div> -->
<h4>до 50мб/с</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">
<i class="fa fa-tags"></i><span>500</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>ROOM 600</h4>
<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer70.png);"></div> -->
<h4>до 70мб/с</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>

```

```

<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">
<i class="fa fa-tags"></i><span>600</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>ROOM 700</h4>
<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer100.png);"></div> -->
<h4>до 100мб/с</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">
<i class="fa fa-tags"></i><span>700</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
<h1>Тарифы на интернет и цифровое телевидение</h1>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>MIX 400</h4>

```

```

<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer100.png);"></div> -->
<h4>до 20 Мбит/с + Цифровое телевидение</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">
<i class="fa fa-tags"></i><span>400</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>ROOM 600</h4>
<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer100.png);"></div> -->
<h4>до 50 Мбит/с + Цифровое телевидение</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">
<i class="fa fa-tags"></i><span>600</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>

```

```

</div>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>MIX700</h4>
<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer100.png);"></div> -->
<h4>до 70 Мбит/с + Цифровое телевидение</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">
<i class="fa fa-tags"></i><span>700</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
<div class="col-md-3 col-sm-6">
<div class="tarif-item">
<h4>MIX800</h4>
<!-- <div class="tarif-item-image" style="background-image:
url(img/speedometer100.png);"></div> -->
<h4>до 100 Мбит/с + Цифровое телевидение</h4>
<div class="tarif-item-stars">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<div class="tarif-item-price">

```

```

<i class="fa fa-tags"></i><span>800</span>руб
</div>
<a href="#" class="button">Подключить</a>
</div>
</div>
</container>
</section>
<section class="s-call inverse" style="background-image:
url(img/call.jpg);">
<div class="container">
<div class="row">
<div class="col-md-5 col-md-offset-7 col-sm-6 col-sm-offset-6">
<form class="callback">
<div class="success"><span>Спасибо за заявку!</span></div>
<!-- Hidden Required Fields -->
<input type="hidden" name="project_name" value="Exellent">
<input type="hidden" name="admin_email" value="hocm13@yandex.ru">
<input type="hidden" name="form_subject" value="Заявка с сайта
Exellent">
<!-- END Hidden Required Fields -->
<input type="hidden" name="Страница" value="">
<div class="callback-logo"></div>
<div class="formhead h3">Оставить заявку</div>
<p>Оставьте заявку и мы вам перезвоним</p>
<input type="text" name="Имя" placeholder="Ваше имя..." required>
<input type="text" name="Телефон" placeholder="Ваш телефон..."
required>
<select name="Услуга">
<option value disabled selected>Выбрать услугу...</option>

```



```

<option>Интернет</option>
<option>Телевидение</option>
<option>Интернет + Телевидение</option>
<option>Интернет в частный дом</option>
<option>Телевидение в частный дом</option>
<option>Интернет + Телевидение в частный дом</option>
</select>
<button class="button button-white">Отправить</button>
</form>
</div>
</div>
</div>
</section>
<section class="s-partners section-overlay" style="background-image:
url(img/bg-3.jpg);">
<div class="container">
<div class="row">
<div class="col-sm-12">
<h2 class="h2 text-center">Платежные системы</h2>
</div>
<div class="col-sm-10 col-sm-offset-1">
<div class="owl-carousel partners">
<div class="partner">
<a href="https://www.citypay.su/?sid=443" target="_blank"></a>
</div>
<div class="partner">
<a href="http://www.krasplat.ru/index.php"></a>
</div>

```

```

<div class="partner">
<a href="#"></a>
</div>
<div class="partner">
<a href="#"></a>
</div>
<div class="partner">
<a href="#"></a>
</div>
<div class="partner">
<a href="#"></a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</section>
  <div id="my-footer">
    <div class="main-footer">
      <div class="container">
        <div class="row">
          <div class="col-lg-2 col-sm-4 col-xs-12"><div class="footer-column">
            
            <div class="social-footer">
              <nav class="soc">
                <ul>
                  <li><a href="#"><i class="fa fa-facebook"></i></a></li>
                  <li><a href="#"><i class="fa fa-instagram"></i></a></li>
                  <li><a href="#"><i class="fa fa-vk"></i></a></li>
                </ul>

```

```

</nav>
</div>
</div></div>
<div class="col-lg-3 col-sm-4 col-xs-12"><div class="footer-column">
<nav class="footer-menu">
<ul>
<li class="active"><a href="#">Интернет</a></li>
<li><a href="#">Телевидение</a></li>
<li><a href="#">Пакетные предложения</a></li>
<li><a href="#">Бизнесу</a></li>
<li><a href="#">Новости</a></li>
<li><a href="#">О нас</a></li>
</ul>
</nav>
</div></div>
<div class="col-lg-3 col-sm-4 col-xs-12"><div class="footer-column">
<p class="footer-addr"><i class="fa fa-location-arrow"></i>г.Красноярск,
ул. Читинская, оф. 230</p>
<p class="footer-addr"><i class="fa fa-clock-o"></i>Время работы: по
будням,<br>10:00-18:00</p>
<p class="footer-addr"><span class="footer-phone"><i class="fa fa-
mobile"></i>282-34-67</span></p>
</div></div>
<div class="col-lg-4 col-sm-12 col-xs-12"><div class="footer-column">
<div class="map">
<script type="text/javascript" charset="utf-8" async src="https://api-
maps.yandex.ru/services/constructor/1.0/js/?um=constructor%3A15b115f06c770b
337fe77927f784822db2d7b33b6042f6d25e30f602afe9fb71&amp;width=100%&a
mp;height=200&amp;lang=ru_RU&amp;scroll=true"></script>
</div>

```


```
</div></div>  
</div>  
</div>  
</div>  
</div>  
</div>  
<link rel="stylesheet" href="css/main.min.css">  
<script src="js/scripts.min.js"></script>  
</body>  
</html>
```

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт педагогики, психологии и социологии  
Кафедра современных образовательных технологий

УТВЕРЖДАЮ  
Заведующий кафедрой  
И.А. Ковалевич  
« 16 » 06 2017 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03.19 Прикладная информатика в социальных коммуникациях  
РАЗРАБОТКА ВЕБ-САЙТА ДЛЯ ИНТЕРНЕТ-ПРОВАЙДЕРА  
«EXELLENT»

Руководитель  доцент каф. СТО, канд.пед.наук М.М Манушкина

Выпускник  Н.В.Бергер

Красноярск 2017 г