

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

**ПРЕДОСТАВЛЕНИЕ КАТАЛОГА ЭЛЕКТРОННЫХ
ТОВАРОВ С ИСПОЛЬЗОВАНИЕМ WEB-ТЕХНОЛОГИЙ**

АВТОРЕФЕРАТ ДИПЛОМНОЙ РАБОТЫ

Студента 6 курса 611 группы

Специальности 010501 – Прикладная математика и информатика

факультета компьютерных наук и информационных технологий

Матвеева Дениса Дмитриевича

Научный руководитель
к.ф.-м.н.

А. С. Иванова

Заведующий кафедрой
к.ф.-м.н.

С. В. Миронов

Саратов, 2016

ВВЕДЕНИЕ

AngularJS – это JavaScript фреймворк, предназначенный для создания одностраничных веб-приложений. AngularJS наделен огромным количеством инструментов облегчающих жизнь программисту. В инструменты AngularJS входят: фильтры, директивы, двусторонняя привязка данных и многое другое.

AngularJS изначально спроектирован для создания одностраничных веб-приложений. Таким образом, браузер изначально откроет только одну страницу и дополнительный контент будет загружаться по мере необходимости. При таком подходе экономится трафик и уменьшается нагрузка на сервер.

AngularJS позволяет сократить время разработки приложений. Это достигается за счет встроенных механизмов, которые сильно помогают разработчику. Один из таких механизмов – это встроенный шаблонизатор (template engine). Шаблонизатор позволяет использовать встроенные команды AngularJS для вывода данных.

Другой прекрасный инструмент, который предоставляет AngularJS, – это то как фреймворк управляет DOM(Document Object Model) или, говоря иначе, HTML тегами. Достаточно указать какие элементы к каким объектам данных привязаны, и при каждом изменении состояния на экране будет выводиться обновленная информация.

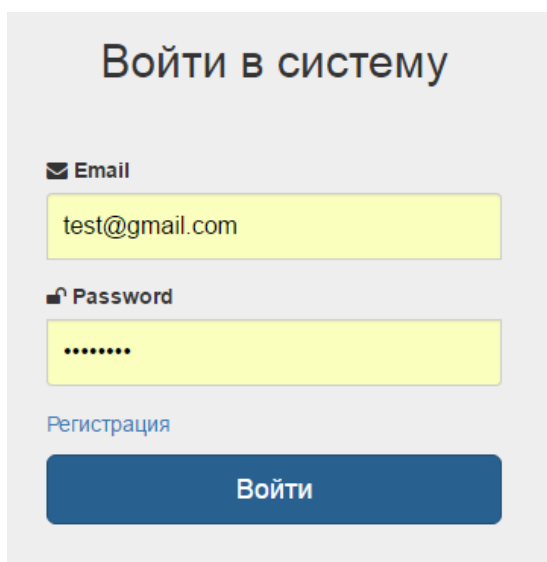
AngularJS использует объектный подход для разработки. Эта архитектура называется MVC (Model View Controller). Отличительной чертой фреймворка является двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом. Таким образом, AngularJS синхронизирует модель и представление.

Цель работы: разработать предоставление справочной информации для телевизоров в виде web-страницы, используя AngularJS, HTML и CSS.

Разработанное приложение представляет из себя web-страницу, предоставляющую справочную информацию о телевизорах.

1. Интерфейс пользователя

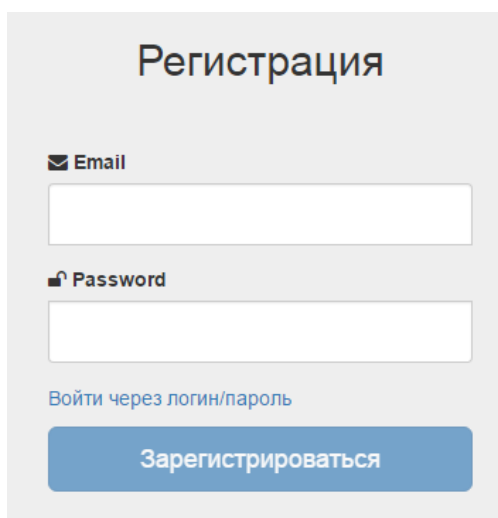
Главная страница представляет страницу авторизации в системе, для этого необходимо ввести адрес электронной почты, пароль и нажать кнопку «Войти» (Рис. 1).



The screenshot shows a login form with the title "Войти в систему". It contains two input fields: "Email" with the value "test@gmail.com" and "Password" with masked characters ".....". Below the password field is a link for "Регистрация" and a blue button labeled "Войти".

Рис. 1

В случае если пользователь не создан в системе, то есть возможность зарегистрироваться, для этого необходимо нажать на «Регистрация». После система перейдет на страницу регистрации (Рис. 2).

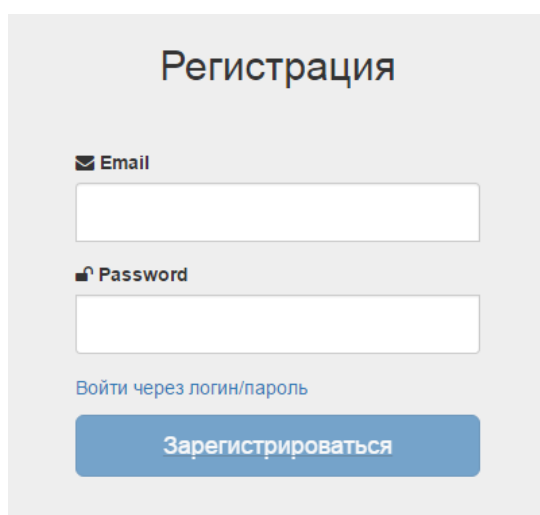


The screenshot shows a registration form with the title "Регистрация". It contains two empty input fields: "Email" and "Password". Below the password field is a link for "Войти через логин/пароль" and a blue button labeled "Зарегистрироваться".

Рис. 2

Указываем адрес электронной почты, пароль и нажимаем на кнопку «Зарегистрироваться». При создании нового пользователя существует ряд проверок, в случае их невыполнение, система не позволит создать нового пользователя.

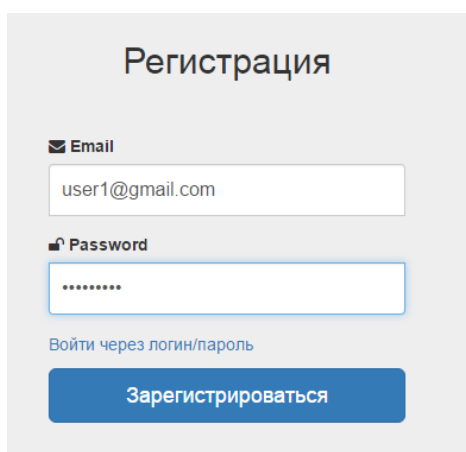
1. Если поля почты и пароля не будут заполнены или некорректны, то кнопка «Зарегистрироваться» будет неактивна (Рис. 3).



The screenshot shows a registration form titled "Регистрация". It contains two input fields: "Email" and "Password". Both fields are empty. Below the fields is a link "Войти через логин/пароль" and a blue button labeled "Зарегистрироваться". The button is inactive, indicated by its greyed-out appearance.

Рис. 3

Кнопка в активном состоянии (Рис. 4).



The screenshot shows the same registration form as in Figure 3, but now the "Email" field is filled with "user1@gmail.com" and the "Password" field is filled with ".....". The "Зарегистрироваться" button is now active and highlighted in blue.

Рис. 4

2. Некорректный формат электронной почты. Необходимо соблюдать формат электронной почты, в противном случае

под полем почты будет выводиться информационное сообщение (Рис. 5).

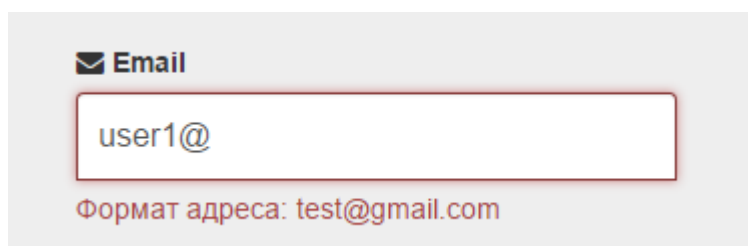


Рис. 5

3. Проверка сложности пароля. Пароль пользователя должен содержать хотя бы одну заглавную, прописную буквы, цифру и знак препинания, а также длина пароля не должна быть меньше 7 символов. Если эти условия не выполнены, то после нажатия на кнопку «Зарегистрироваться» будет выведено сообщение об ошибке, и система не позволит зарегистрировать пользователя с таким паролем (Рис. 6).

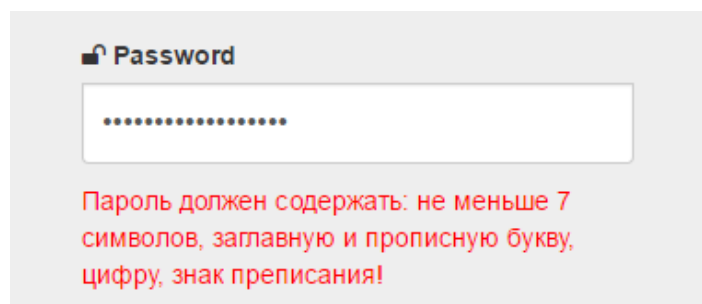


Рис. 6

4. Если указан адрес электронной почты, для которой был уже зарегистрирован, то будет выведено сообщение об ошибке (Рис. 7).

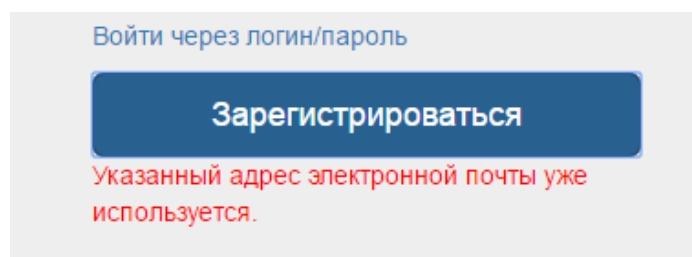


Рис. 7

Если никаких ошибок нет, то будет выведено сообщение, что пользователь успешно создан в системе и система переадресует на страницу авторизации пользователя. После успешной авторизации система переадресует на страницу каталога.

2 Каталог товаров

Страница каталога представляет список телевизоров с краткой информацией о каждом продукте (Рис. 8).

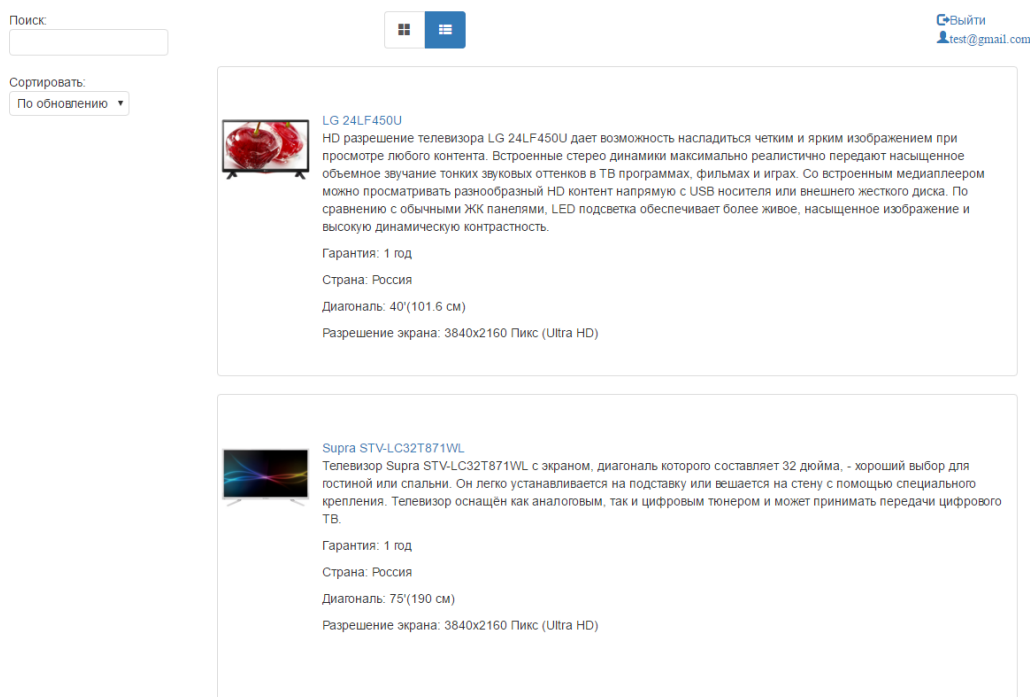


Рис. 8

В левом верхнем углу реализован поиск по всему списку, есть возможность найти определенный вид телевизора по ключевому слову, введенному в поле поиска (Рис.9).

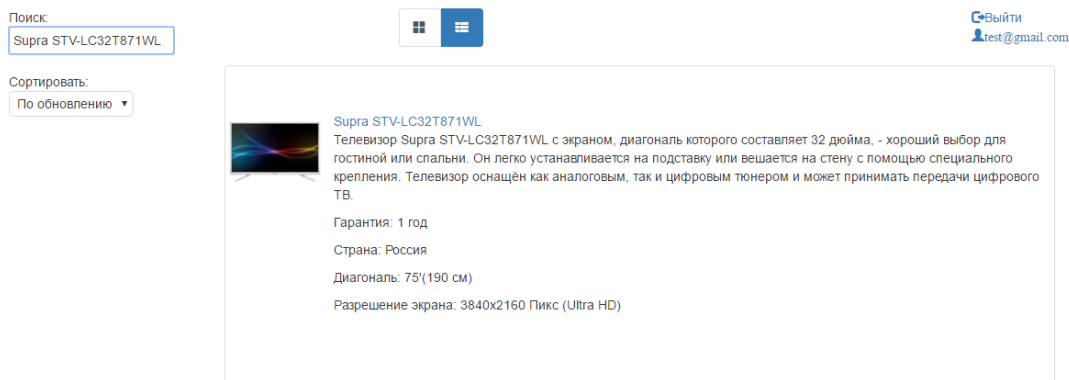


Рис. 9

Ниже представлена возможность сортировки. Отсортировать данные можно по обновлению или по алфавиту (Рис. 10).

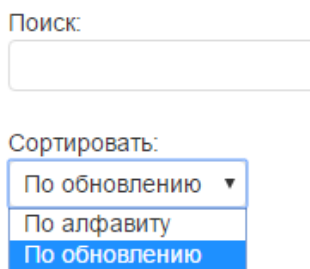


Рис. 10

В правом верхнем углу указан адрес электронной почты, под пользователем которой был выполнен вход в систему. А также кнопка «Выход», при нажатии на которую можно выйти из каталога и вернуться на страницу авторизации (Рис. 11).



Рис. 11

Над каталогом есть две кнопки, которые позволяют менять вид отображения каталога. По умолчанию выбрана правая кнопка и каталог отображается с краткой информацией и фото. Если нажать на левую кнопку,

то каталог будет представлен в сокращенный форме, где будет показано фото и название товара (Рис. 12).

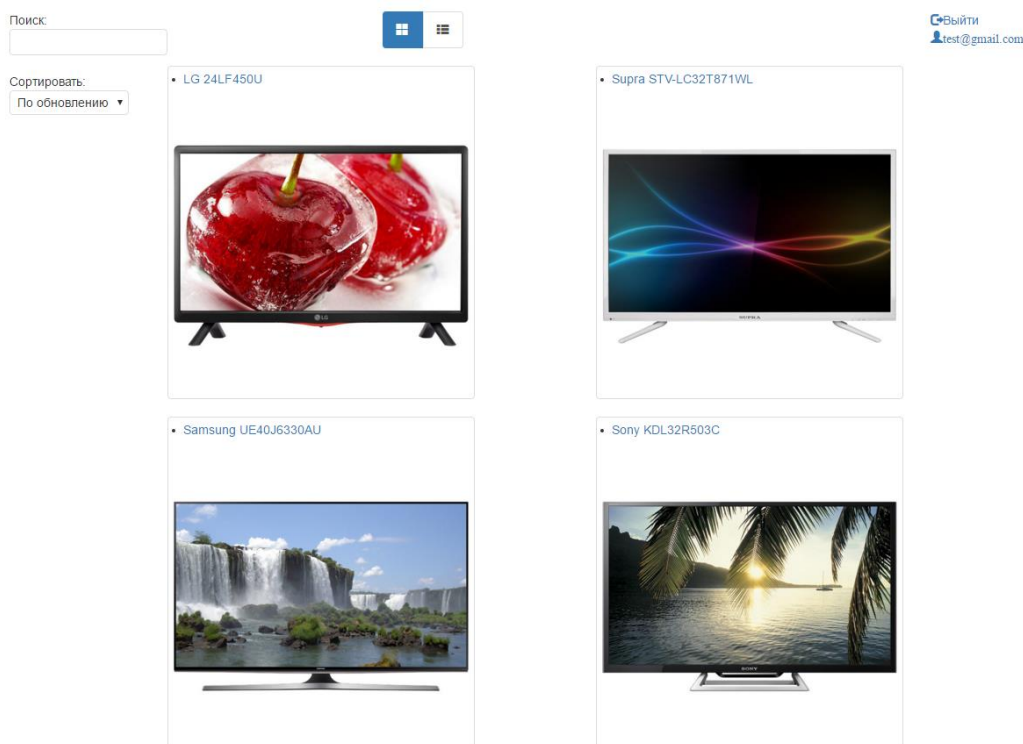


Рис. 12

В любом виде каталога инициировать переход на страницу с подробной информацией о объекте можно нажатием по названию или изображению товара. Окончание url представляет собой уникальный идентификатор объекта (Рис.13).

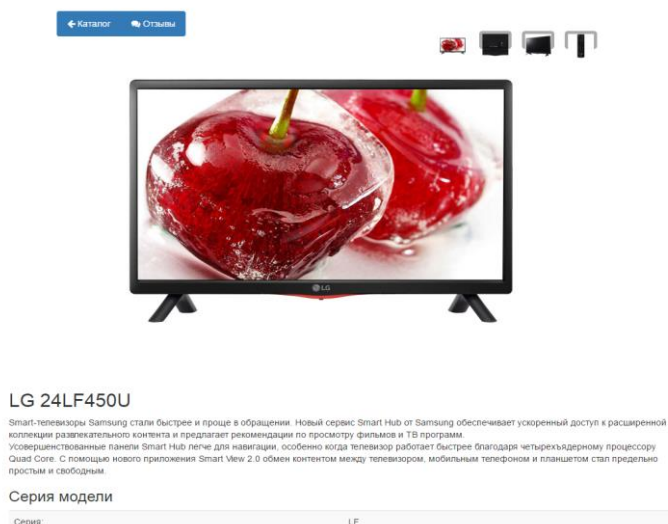


Рис. 13

Подробная информация состоит из названия, фотографий, подробного описания модели и главных характеристик телевизора .

Есть возможность просмотра всех фотографий телевизора, для этого необходимо нажать на фотографию над главной фотографией, и изменится основная (Рис. 14). Для изменения фотографии используется анимация.



Рис. 14

На странице с подробной информации есть возможность вернуться обратно в каталог, для этого необходимо нажать кнопку «Каталог» .

При нажатии на кнопку «Отзывы» можно перейти на страницу со всеми отзывами для данного товара.

3 Страница отзывов товара

Страница отзывов представляет собой список всех отзывов, которые были написаны для данного товара, а также форма создания нового отзыва (Рис. 16).

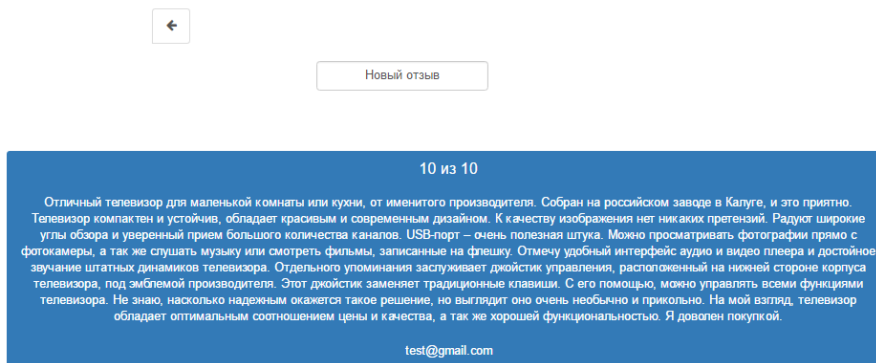


Рис. 16

С помощью кнопки в виде стрелки можно вернуться обратно к товару.

Чтобы оставить новый отзыв необходимо нажать на кнопку «Новый отзыв», появится форма для создания (Рис. 17).

A screenshot of a review form. At the top left is a square button with a left-pointing arrow. To its right is a rectangular button labeled "Новый отзыв". Below the arrow button are two input fields: the first is labeled "Заголовок" and the second is labeled "Комментарий". At the bottom center is a blue rectangular button labeled "Оставить отзыв".

Рис. 21

Заполняем поле Заголовок, Комментарии, нажимаем кнопку «Оставить отзыв», форма закроется и отзыв появится в списке (Рис. 18).

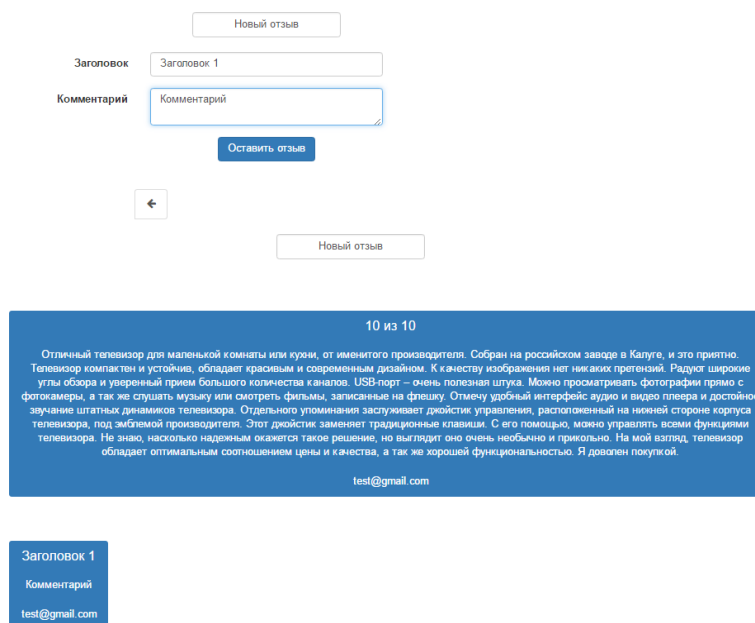


Рис. 18

Заключение

В ходе работы были изучены основы AngularJS, HTML, CSS и JSON. Данные знания были применены для реализации простого web-приложения в виде справочной информации.

В конце работы хотелось бы выделить несколько преимуществ разработки на AngularJS:

1. Декларативный стиль кода

При создании шаблонов в AngularJS применяется декларативная парадигма программирования. Это делает код более легковесным, облегчает его чтение и поддержку, так как описывается необходимый конечный результат.

2. Использование директив

В качестве языка шаблонов в AngularJS используется HTML. Он расширяется с помощью директив, которые добавляют в код

сведения о требуемом поведении (например, о необходимости загрузить определенный модуль сразу после загрузки страницы). Директивы позволяют сконцентрироваться на проработке логики и работать более продуктивно. Их можно использовать повторно, что также повышает читабельность кода.

3. MVC

В AngularJS используется схема MVC, разделяющая логику, представление и данные приложения. Это позволяет создавать одностраничные веб-приложения.

4. Двустороннее связывание данных

В AngularJS применяется двустороннее связывание: любые изменения в пользовательском интерфейсе сразу отражаются на объектах приложения и наоборот. Фреймворк сам следит за событиями браузера, изменениями модели и действиями пользователя на странице, чтобы сразу обновлять нужные шаблоны. При этом в коде JavaScript не требуется хранить ссылки на DOM-элементы и явно ими манипулировать.

5. Наличие готовых решений

Для AngularJS существует огромное количество готовых решений, которые позволяют решать довольно разнообразные задачи, используя уже готовые модули. Например, существует несколько модулей для роутинга, самый популярный из которых `ui-router`, так же есть различные модули для работы с таблицами `ui-grid`, `ng-table` и много других.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. AngularJS: введение и основные концепции [Электронный ресурс] – URL: <https://makeomatic.ru/blog/2013/08/14/AngularJSIntro/> (дата обращения 15.01.2016). Загл. с экрана. Яз.рус.

2. AngularJS [Электронный ресурс] URL: <https://docs.angularjs.org/api> (дата обращения 15.01.2016). Загл. с экрана. Яз.рус.
3. StackOverflow [Электронный ресурс] URL: <http://stackoverflow.com> (дата обращения 17.01.2016). Загл. с экрана. Яз.рус.
4. Изучаем AngularJS на пяти практических примерах [Электронный ресурс] URL: http://www.internet-technologies.ru/articles/article_1637.html (дата обращения 20.01.2016). Загл. с экрана. Яз.рус.
5. AngularJS. Быстрое знакомство с популярным фрейворком от Google [Электронный ресурс] URL: <http://iantonov.me/page/angularjs-osnovy> (дата обращения 20.01.2016). Загл. с экрана. Яз.рус.
6. stfalcon.com [Электронный ресурс] URL: <http://stfalcon.com/blog/post/why-use-angularjs-for-webapps> (дата обращения 1.02.2016). Загл. с экрана. Яз.рус.
7. CSS [Электронный ресурс] URL: <http://www.ruled.ru/sintacs-css.html> (дата обращения 7.02.2016). Загл. с экрана. Яз.рус.
8. Руководство по CSS для начинающих [Электронный ресурс] URL: <http://technologyweb.org/> (дата обращения 5.02.2016). Загл. с экрана. Яз.рус.
9. JSON: основы использования [Электронный ресурс] URL: <http://ruseller.com/lessons.php?rub=28&id=1212> (дата обращения 1.02.2016). Загл. с экрана. Яз.рус.
10. Создание AngularJS приложения с использованием Firebase [Электронный ресурс] URL: <https://habrahabr.ru/post/244209/> (дата обращения 25.04.2016). Загл. с экрана. Яз.рус.
11. Эпоха динамических веб-страниц – Firebase [Электронный ресурс] URL: <http://webix.com/ru/blog/epoxa-dinamicheskix-veb-stranic-firebase/> (дата обращения 25.04.2016). Загл. с экрана. Яз.рус.