

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Данный раздел является описанием работы, подключения, способов взаимодействия с программно-аппаратного комплекса видеоконтроля в системе «Умный дом» для конечного пользователя, что позволит уменьшить время для ознакомления с устройством.

Данный раздел разбит на две части: программную и аппаратную. В программном разделе приводится описание компонент веб-приложения и руководство для его запуска и использования. В аппаратном разделе приведено описание способов подключения камеры, питания, подключения устройства к сети по Wi-Fi.

Данный программно-аппаратный комплекс является очень легким в освоении и имеет минимальный порог вхождения для ознакомления с его работой.

6.1 Аппаратная часть

Перед тем как начать работу с комплексом необходимо ознакомиться с методами подключения устройства к питанию. Устройство имеет micro-USB порт и подключается кабелем USB – micro-USB. Для подачи питания используется блок питания с выходным напряжением 5В и силой тока 2А. После того как устройство подключиться к питанию на микрокомпьютере загорится зеленый светодиод, как показано на рисунке 6.1, который свидетельствует о том, что он включен.



Рисунок 6.1 – Подключение питания к Raspberry Pi Zero W.

В качестве операционной системы микрокомпьютера используется дистрибутив Linux – Raspbian. Эта операционная система является официальной для микрокомпьютеров семейства Raspberry. Ее образ не занимает много места и вместе с графической оболочкой помещается на microSD накопитель объемом в 8 ГБ. После загрузки образа на флэш-накопитель, необходимо вставить его в разъем на плате микрокомпьютера и подключить питание.

После того как микрокомпьютер включен, для дальнейшей настройки необходимо подключить его к монитору через кабель HDMI – mini-HDMI, а так же подключить устройства ввода через переходный кабель micro-USB – USB. Так как все необходимые библиотеки уже включены в дистрибутив, то следует просто загрузить управляющую программу на устройство. Далее необходимо установить соединение с сервером веб-приложения. Так как обмен данными осуществляется через Wi-Fi, нужно чтобы устройство и веб-приложение находились в одной сети.

Чтобы подключить устройство к сети, нужно включить Wi-Fi в меню работы с сетью, как показано на рисунке 6.2, и из списка обнаруженных точек доступа выбрать соответствующую. Далее ввести пароль. Если необходимо подключаться к веб-приложению, которое развернуто пользователем локально, тогда в управляющей программе необходимо сменить значение константы SERVER_URL на ip-адрес устройства, на котором развернуто веб-приложение, с указанием порта, который прослушивает веб-приложение.

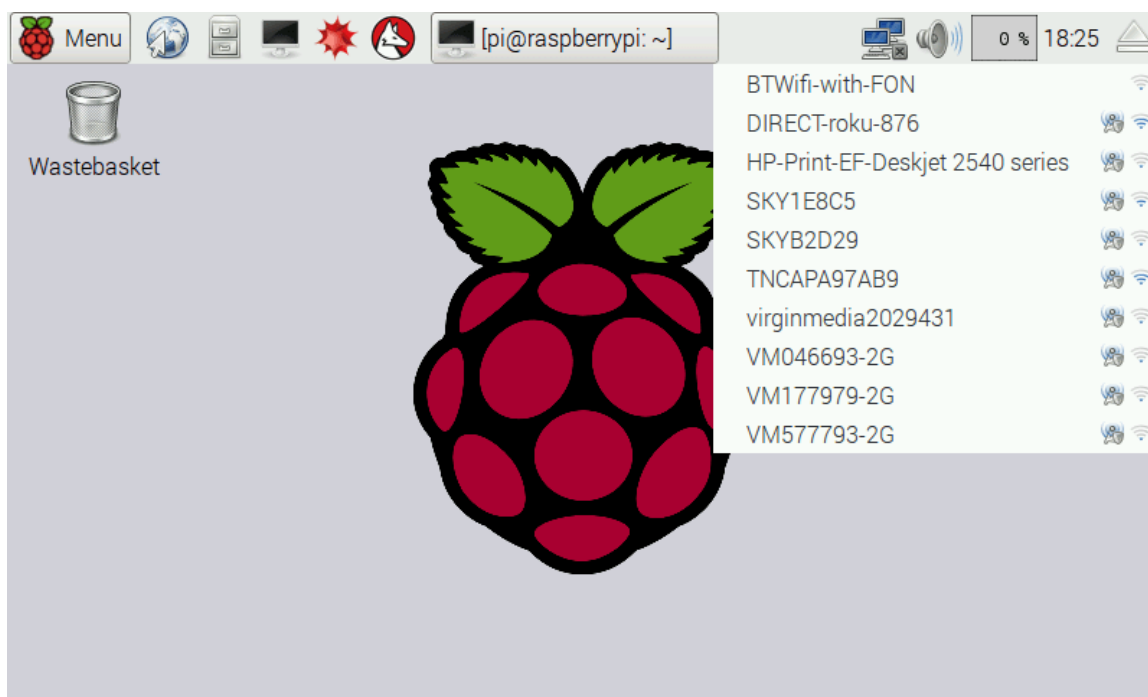


Рисунок 6.2 – Установка соединения с сетью посредством подключения к точке доступа Wi-Fi.

На этом настройка аппаратной части завершена, при настроенной программной части устройство будет успешно посылать сделанные фотографии на сервер веб-приложения, пример сделанной фотографии можно увидеть на рисунке 6.3.



Рисунок 6.3 – Пример фотографии, сделанный камерой.

6.2 Программная часть

Для запуска веб-приложения необходима операционная система Ubuntu 16.04.

Ubuntu 16.04 – это дистрибутив Linux, которая распространяется бесплатно. Образ системы можно скачать на официальном сайте, как показано на рисунке 6.4.

Network installer

The network installer lets you install Ubuntu over a network. It includes the minimal set of packages needed to start and the rest of the packages are downloaded over the network. Since only current packages are downloaded, there is no need to upgrade packages immediately after installation.

The network installer is ideal if you have a computer that cannot run the graphical installer, for example, because it does not meet the minimum requirements for the live CD/DVD, or because the computer requires extra configuration before a graphical desktop can be used. The network installer is also useful if you want to install Ubuntu on a large number of computers at once.

- ✓ [Download the network installer for 18.04 LTS](#)
- ✓ [Download the network installer for 16.04 LTS](#)
- ✓ [Download the network installer for 14.04 LTS](#)

BitTorrent

BitTorrent is a peer-to-peer download network that sometimes enables higher download speeds and more reliable downloads of large files. You will need to install a BitTorrent client on your computer in order to enable this download method.

Ubuntu 18.04 LTS

[Ubuntu 18.04 Desktop \(64-bit\)](#)

[Ubuntu 18.04 Server \(64-bit\)](#)

Ubuntu 16.04 LTS

[Ubuntu 16.04 Desktop \(64-bit\)](#)

[Ubuntu 16.04 Desktop \(32-bit\)](#)

[Ubuntu 16.04 Server \(64-bit\)](#)

Ubuntu 14.04.5 LTS

[Ubuntu 14.04.5 Desktop \(64-bit\)](#)

[Ubuntu 14.04.5 Desktop \(32-bit\)](#)

[Ubuntu 14.04.5 Server \(64-bit\)](#)

Рисунок 6.4 – Официальный сайт разработчиков ОС Ubuntu.

После записи образа системы на флэш-накопитель, необходимо произвести установку ОС, как показано на рисунке 6.5. Для этого рекомендуется подключить рабочую станцию к сети проводным или беспроводным способом. Далее необходимо выбрать язык локализации операционной системы, часовой пояс, раздел на диске, куда будет установлена система, а также задать имя пользователя, имя рабочей станции и пароль. Опционально можно активировать возможность скачивания последних обновлений системы непосредственно во время ее установки, чтобы сэкономить время после установки.

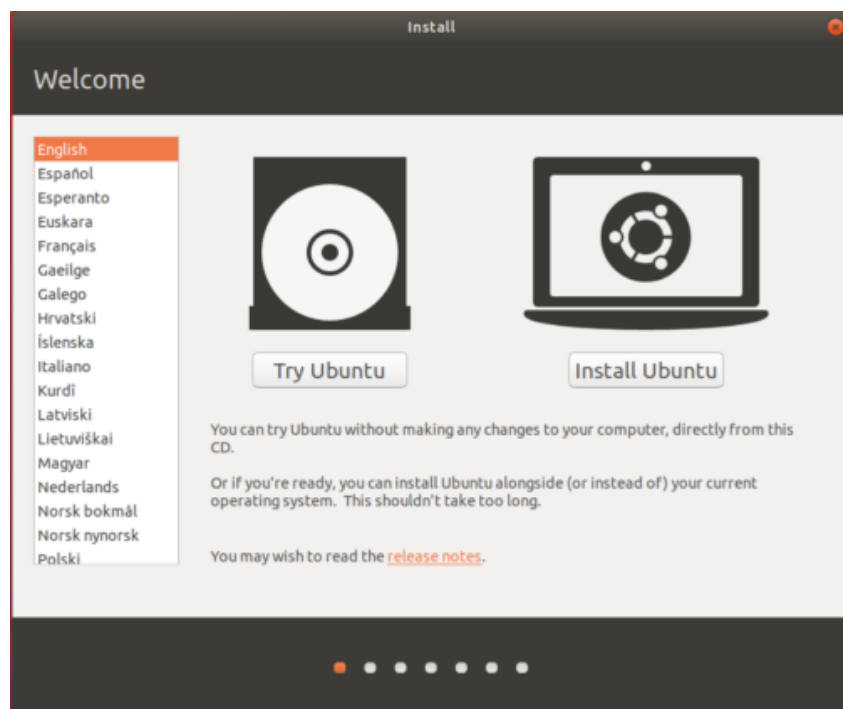


Рисунок 6.5 – Установка операционной системы Ubuntu.

После установки системы необходимо установить утилиту RVM. RVM – это инструмент, который позволяет легко устанавливать, управлять и работать со множеством версий интерпретаторов и наборов библиотек. Это необходимо для разделения окружений каждого из приложений, так как они могут использовать разные версии одинаковых библиотек. При установке их напрямую в систему это может нарушить работу других программ.

Для установки RVM необходимо выполнить в командной строке, как показано на рисунке 6.6, следующую команду:

```
curl -sSL https://get.rvm.io | bash -s stable -
ruby
```

После выполнения данной команды необходимо позволить выполнение команд при входе в систему, как показано на рисунке 6.7

```
ubuntu: ~
ubuntu@ubuntu:~$ curl -sSL https://get.rvm.io | bash -s stable
Downloading https://github.com/rvm/rvm/archive/1.28.0.tar.gz
Downloading https://github.com/rvm/rvm/releases/download/1.28.0/1.28.0.tar.gz.asc
gpg: Signature made Mon 19 Dec 2016 06:50:13 PM UTC using RSA key ID BF04FF17
gpg: Good signature from "Michal Papisi (RVM signing) <mpapisi@gmail.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:       There is no indication that the signature belongs to the owner.
Primary key fingerprint: 409B 6B17 96C2 7546 2A17  0311 3804 BB82 D39D C0E3
Subkey fingerprint: 62C0 E5F4 DA30 0D94 AC36  166B E206 C29F BF04 FF17
GPG verified '/home/ubuntu/.rvm/archives/rvm-1.28.0.tgz'

Installing RVM to /home/ubuntu/.rvm/
Adding rvm PATH line to /home/ubuntu/.profile /home/ubuntu/.mkshrc /home/ubuntu/.bashrc /home/ubuntu/.zshrc.
Adding rvm loading line to /home/ubuntu/.profile /home/ubuntu/.bash_profile /home/ubuntu/.zlogin.
Installation of RVM in /home/ubuntu/.rvm/ is almost complete:
```

Рисунок 6.6 – Выполнение команды установки RVM.

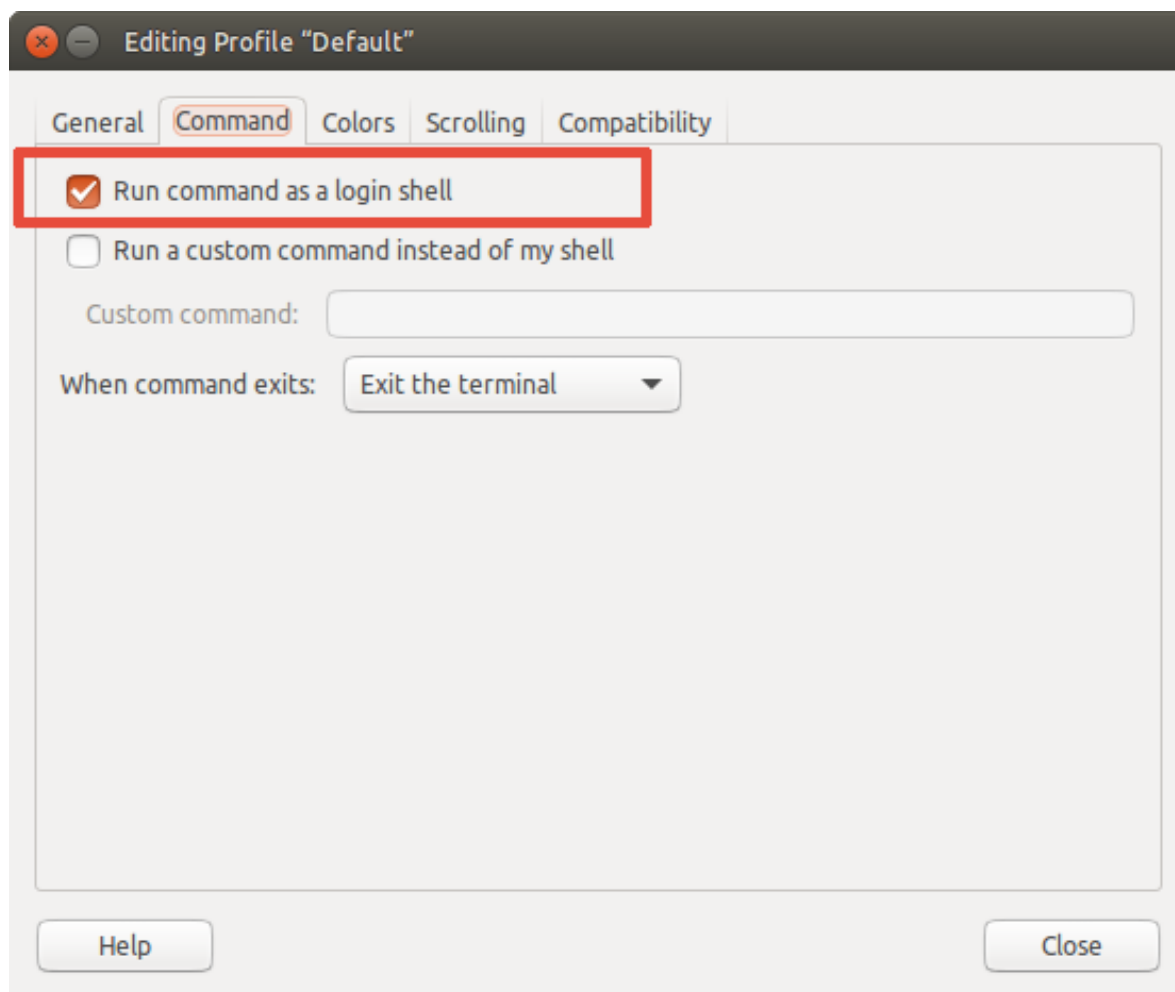


Рисунок 6.7 – Запуск команд при входе в систему.

После установки RVM, следует установить пакет `libpq-dev`, необходимый для корректной работы базы данных PostgreSQL. Для этого следует выполнить данную команду:

```
sudo apt install libpq-dev
```

Дальше предстоит установка Docker. Это ПО для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Для установки Docker необходимо выполнить в командной строке следующие команды в строгом порядке:

1. `sudo apt-get update`
2. `sudo apt-get install \`
`apt-transport-https \`
`ca-certificates \`
`curl \`
`software-properties-common`

3. `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
4. `sudo add-apt-repository \`
`"deb [arch=amd64] https://download.docker.com/linux/ubuntu \`
`$(lsb_release -cs) \`
`stable"`
5. `sudo apt-get update`
6. `sudo apt-get install docker-ce`

В контейнере Docker мы запустим базу данных PostgreSQL, которая хранит данные веб-приложения, с помощью следующей команды, результат которой изображен на рисунке 6.8:

```
sudo docker run -d -p 5432:5432 --name camera_secure_web_app postgres:9.5
```

Данная команда автоматически скачает и запустит в контейнере образ базы данных PostgreSQL версии 9.5. Контейнер обменивается информацией с системой через 5432 порт, который мы указываем в команде.



```
alexsuslik@alexsuslik-N56VV:~$ sudo docker run -d -p 5432:5432 --name camera_secure_web_app postgres:9.5
Unable to find image 'postgres:9.5' locally
9.5: Pulling from library/postgres
f2aa67a397c4: Already exists
8218dd41bf94: Already exists
e9b7fa2e6bd8: Already exists
7288a45ee17f: Already exists
0d0f8a67376c: Already exists
972b115243de: Already exists
d38528c83dd1: Already exists
9be166d23dee: Already exists
fee4a201e525: Pull complete
722e80813bb5: Pull complete
3fec20710223: Pull complete
e08ccca798a2: Pull complete
46a73d452c3d: Pull complete
76d46f84a1e5: Pull complete
Digest: sha256:0d94b9517961ef1d09f6db201375fc2d68e169256fc29d911ca60a3d11b040c5
Status: Downloaded newer image for postgres:9.5
```

Рисунок 6.8 – Выполнение команды запуска контейнера с базой данных
номера

После этого необходимо перейти в директорию веб-приложения и установить интерпретатор языка Ruby версии, которая указана в файле `ruby-version`, как показано на рисунке 6.9. Для этого необходимо выполнить следующую команду:

```
rvm install <версия интерпретатора>
```

```

ubuntu@ubuntu:~$ rvm install 2.4.0
Searching for binary rubies, this might take some time.
Found remote file https://rvm.io.global.ssl.fastly.net/binaries/ubuntu/16.04/x86_64/ruby-2.4.0.tar.bz2
Checking requirements for ubuntu.
Installing requirements for ubuntu.
Updating system...
Installing required packages: gawk, libsqlite3-dev, sqlite3, libgmp-dev, libtool.....
Requirements installation successful.
ruby-2.4.0 - #configure
ruby-2.4.0 - #download
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 16.4M  100 16.4M    0     0  76536      0  0:03:45  0:03:45 --:--:-- 78136
No checksum for downloaded archive, recording checksum in user configuration.
ruby-2.4.0 - #validate archive
ruby-2.4.0 - #extract
ruby-2.4.0 - #validate binary
ruby-2.4.0 - #setup
ruby-2.4.0 - #gemset created /home/ubuntu/.rvm/gems/ruby-2.4.0@global
ruby-2.4.0 - #importing gemset /home/ubuntu/.rvm/gemsets/global.gems.....
ruby-2.4.0 - #generating global wrappers.....
ruby-2.4.0 - #gemset created /home/ubuntu/.rvm/gems/ruby-2.4.0
ruby-2.4.0 - #importing gemsetfile /home/ubuntu/.rvm/gemsets/default.gems evaluated to empty gem list
ruby-2.4.0 - #generating default wrappers.....

```

Рисунок 6.9 – Выполнение команды установки интерпретатора языка Ruby

Следующим шагом является установка фреймворка Rails, с помощью которого написано веб-приложение. Для этого нужно выполнить данную команду:

```
gem install rails
```

Результат выполнения данной команды можно увидеть на рисунках 6.10 и 6.11.

```

alexssu@alexssu@ubuntu:~/web_app$ gem install rails
Successfully installed concurrent-ruby-1.0.5.gem (100%)
Fetching: tzinfo-1.0.1.gem (100%)
Successfully installed tzinfo-1.0.1
Fetching: thread_safe-0.3.6.gem (100%)
Successfully installed thread_safe-0.3.6
Fetching: tzinfo-1.2.5.gem (100%)
Successfully installed tzinfo-1.2.5
Fetching: activesupport-5.2.0.gem (100%)
Successfully installed activesupport-5.2.0
Fetching: rack-2.0.5.gem (100%)
Successfully installed rack-2.0.5
Fetching: rack-test-1.0.0.gem (100%)
Successfully installed rack-test-1.0.0
Fetching: mini_portile2-2.3.0.gem (100%)
Successfully installed mini_portile2-2.3.0
Fetching: nokogiri-1.8.2.gem (100%)
Building native extensions. This could take a while...
Successfully installed nokogiri-1.8.2
Fetching: crass-1.0.4.gem (100%)
Successfully installed crass-1.0.4
Fetching: loofah-2.2.2.gem (100%)
Successfully installed loofah-2.2.2
Fetching: rails-html-sanitizer-1.0.4.gem (100%)
Successfully installed rails-html-sanitizer-1.0.4
Fetching: rails-dom-testing-2.0.3.gem (100%)
Successfully installed rails-dom-testing-2.0.3
Fetching: builder-3.2.3.gem (100%)
Successfully installed builder-3.2.3
Fetching: erubi-1.7.1.gem (100%)
Successfully installed erubi-1.7.1
Fetching: actionview-5.2.0.gem (100%)
Successfully installed actionview-5.2.0
Fetching: actionpack-5.2.0.gem (100%)
Successfully installed actionpack-5.2.0
Fetching: activemodel-5.2.0.gem (100%)
Successfully installed activemodel-5.2.0
Fetching: arel-9.0.0.gem (100%)
Successfully installed arel-9.0.0
Fetching: activerecord-5.2.0.gem (100%)
Successfully installed activerecord-5.2.0
Fetching: globalid-0.4.1.gem (100%)
Successfully installed globalid-0.4.1
Fetching: activejob-5.2.0.gem (100%)
Successfully installed activejob-5.2.0
Fetching: mini_mime-1.0.0.gem (100%)
Successfully installed mini_mime-1.0.0
Fetching: mail-2.7.0.gem (100%)
Successfully installed mail-2.7.0
Fetching: actionmailer-5.2.0.gem (100%)
Successfully installed actionmailer-5.2.0
Fetching: nio4r-2.3.1.gem (100%)
Building native extensions. This could take a while...
Successfully installed nio4r-2.3.1
Fetching: websocket-extensions-0.1.3.gem (100%)
Successfully installed websocket-extensions-0.1.3
Fetching: websocket-driver-0.7.0.gem (100%)
Building native extensions. This could take a while...
Successfully installed websocket-driver-0.7.0
Fetching: actioncable-5.2.0.gem (100%)
Successfully installed actioncable-5.2.0

```

Рисунок 6.10 – Начало выполнения команды установки фреймворка Rails

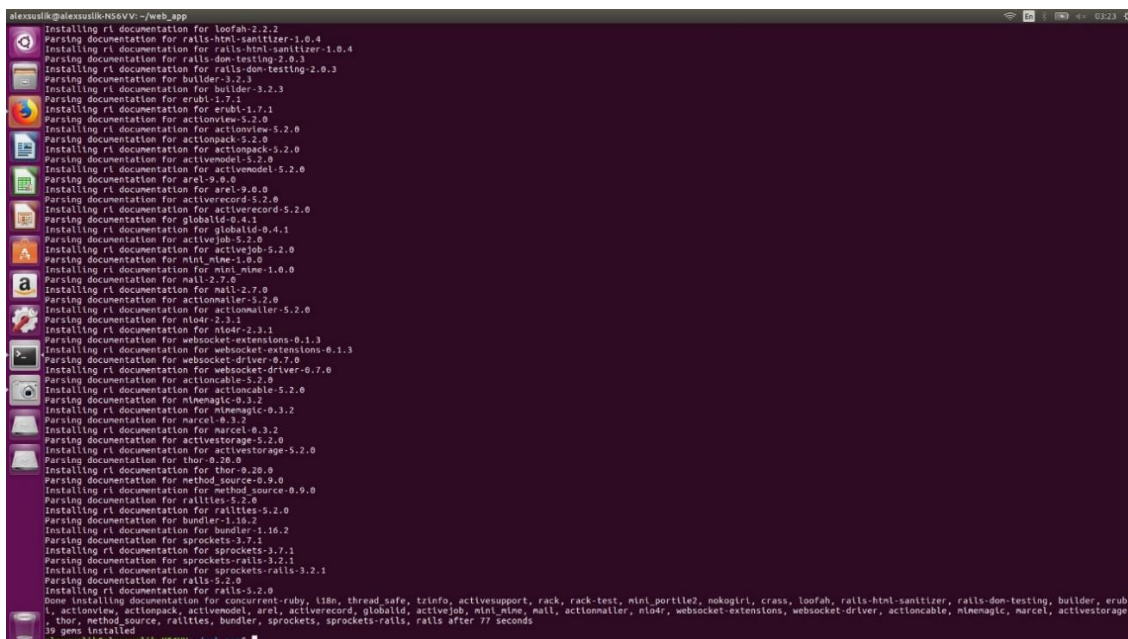


Рисунок 6.11 – Завершение выполнения команды установки фреймворка Rails

Далее необходимо установить утилиту Bundler, которая позволяет автоматически определять версии библиотек указанных в файле Gemfile, управлять их зависимостями, скачивать их и подключать в проект. Для этого необходимо ввести в командной строке следующую команду:

```
gem install bundler
```

Результат выполнения этой команды виден на рисунке 6.12.

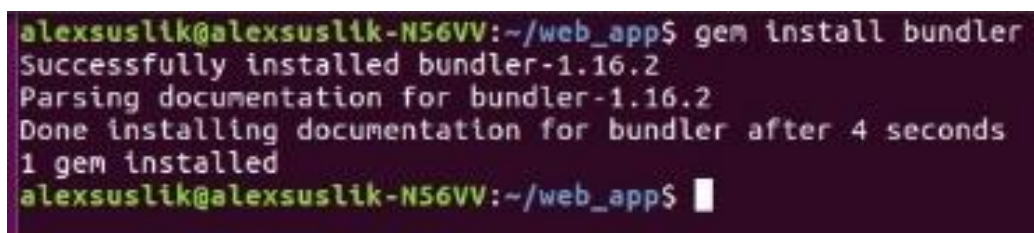
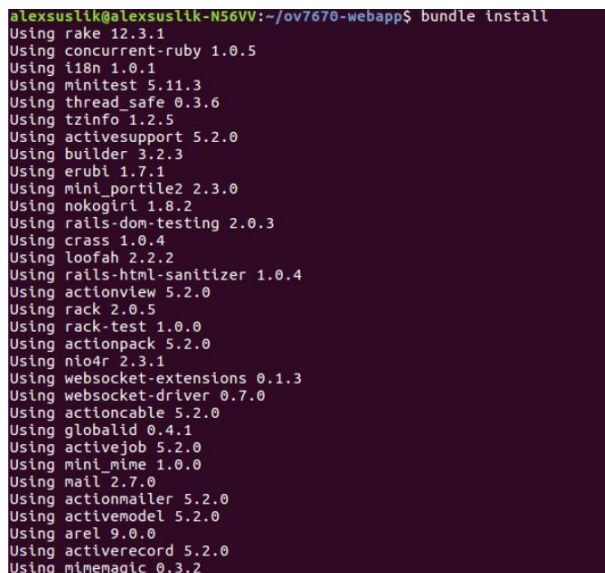


Рисунок 6.12 –Результат выполнения команды установки утилиты Bundler.

После установки утилиты Bundler необходимо воспользоваться ей для скачивания всех необходимых для работы веб-приложения библиотек. Они перечислены в файле с названием Gemfile. Данный файл составляется разработчиком, а утилита Bundler на его основании генерирует файл с названием Gemfile.lock, который хранит не только список библиотек, но также и их зависимости с указанием версий, необходимых для корректной работы. Для скачивания библиотек необходимо выполнить следующую команду в командной строке:

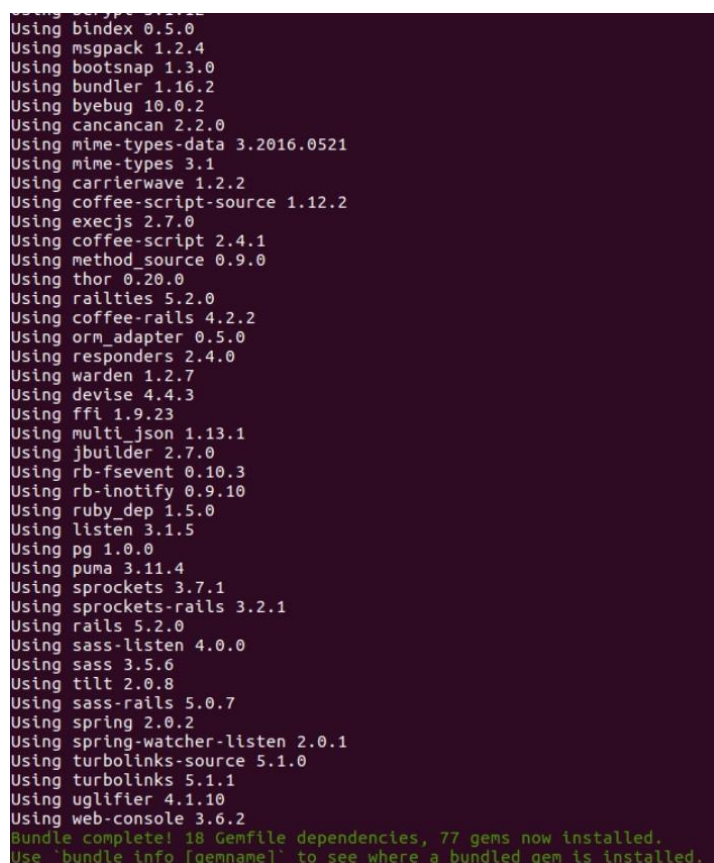
```
bundle install
```

Начало и окончание выполнения данной команды показано на рисунках 6.13 и 6.14, соответственно.



```
alexsuslik@alexsuslik-N56VW:~/ov7670-webapp$ bundle install
Using rake 12.3.1
Using concurrent-ruby 1.0.5
Using i18n 1.0.1
Using minitest 5.11.3
Using thread_safe 0.3.6
Using tzinfo 1.2.5
Using activesupport 5.2.0
Using builder 3.2.3
Using erubi 1.7.1
Using mini_portile2 2.3.0
Using nokogiri 1.8.2
Using rails-dom-testing 2.0.3
Using crass 1.0.4
Using loofah 2.2.2
Using rails-html-sanitizer 1.0.4
Using actionview 5.2.0
Using rack 2.0.5
Using rack-test 1.0.0
Using actionpack 5.2.0
Using nio4r 2.3.1
Using websocket-extensions 0.1.3
Using websocket-driver 0.7.0
Using actioncable 5.2.0
Using globalid 0.4.1
Using activejob 5.2.0
Using mini_mime 1.0.0
Using mail 2.7.0
Using actionmailer 5.2.0
Using activemodel 5.2.0
Using arel 9.0.0
Using activerecord 5.2.0
Using mimemagic 0.3.2
```

Рисунок 6.13 –Начало выполнения команды скачивания библиотек.



```
Using bindex 0.5.0
Using msgpack 1.2.4
Using bootsnap 1.3.0
Using bundler 1.16.2
Using byebug 10.0.2
Using cancancan 2.2.0
Using mime-types-data 3.2016.0521
Using mime-types 3.1
Using carrierwave 1.2.2
Using coffee-script-source 1.12.2
Using execjs 2.7.0
Using coffee-script 2.4.1
Using method_source 0.9.0
Using thor 0.20.0
Using railties 5.2.0
Using coffee-rails 4.2.2
Using orm_adapter 0.5.0
Using responders 2.4.0
Using warden 1.2.7
Using devise 4.4.3
Using ffi 1.9.23
Using multi_json 1.13.1
Using jbuilder 2.7.0
Using rb-fsevent 0.10.3
Using rb-inotify 0.9.10
Using ruby_dep 1.5.0
Using listen 3.1.5
Using pg 1.0.0
Using puma 3.11.4
Using sprockets 3.7.1
Using sprockets-rails 3.2.1
Using rails 5.2.0
Using sass-listen 4.0.0
Using sass 3.5.6
Using tilt 2.0.8
Using sass-rails 5.0.7
Using spring 2.0.2
Using spring-watcher-listen 2.0.1
Using turbolinks-source 5.1.0
Using turbolinks 5.1.1
Using uglifier 4.1.10
Using web-console 3.6.2
Bundle complete! 18 Gemfile dependencies, 77 gems now installed.
Use 'bundle info [gemname]' to see where a bundled gem is installed.
```

Рисунок 6.14 –Окончание выполнения команды скачивания библиотек.

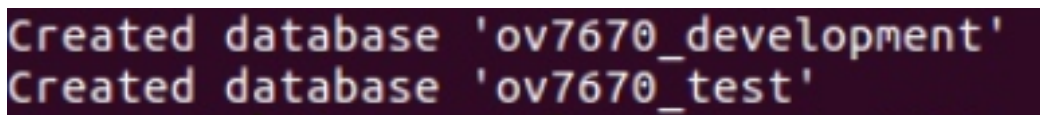
Для корректной работы приложения осталось создать таблицы базы данных и произвести миграции изменений схемы базы данных, для этого необходимо выполнить две команды. Команда для создания таблицы базы данных:

```
rake db:create
```

За ней следует команда для проведения миграций изменений в схеме базы данных:

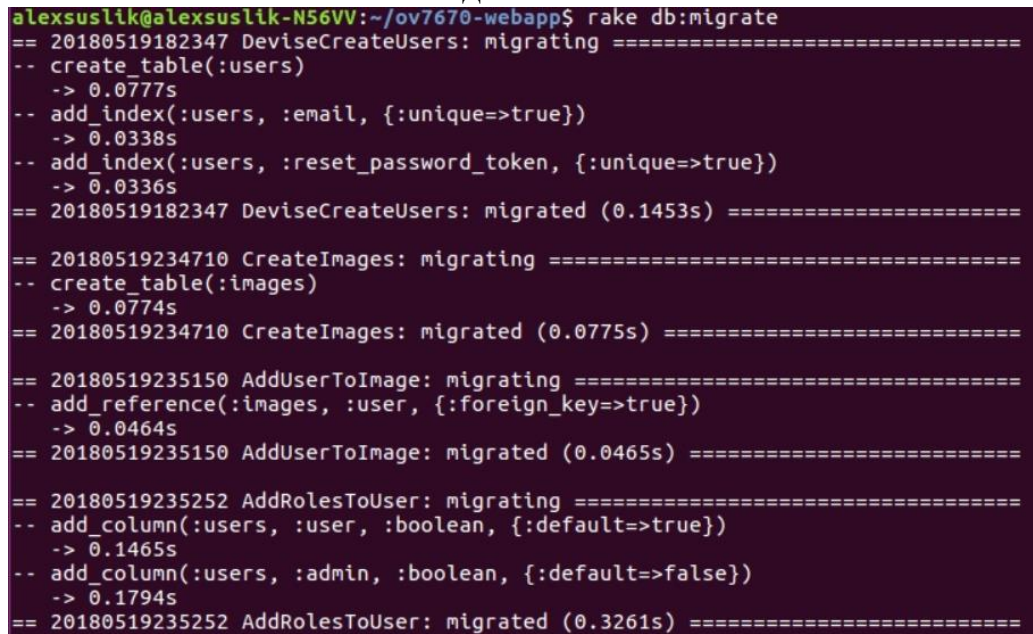
```
rake db:migrate
```

Результаты выполнения команд можно увидеть на рисунках 6.15 и 6.16.



```
Created database 'ov7670_development'  
Created database 'ov7670_test'
```

Рисунок 6.15 –Результат выполнения команды создания таблиц базы данных.



```
alexsuslik@alexsuslik-N56VV:~/ov7670-webapp$ rake db:migrate  
== 20180519182347 DeviseCreateUsers: migrating =====  
-- create_table(:users)  
-> 0.0777s  
-- add_index(:users, :email, {:unique=>true})  
-> 0.0338s  
-- add_index(:users, :reset_password_token, {:unique=>true})  
-> 0.0336s  
== 20180519182347 DeviseCreateUsers: migrated (0.1453s) =====  
  
== 20180519234710 CreateImages: migrating =====  
-- create_table(:images)  
-> 0.0774s  
== 20180519234710 CreateImages: migrated (0.0775s) =====  
  
== 20180519235150 AddUserToImage: migrating =====  
-- add_reference(:images, :user, {:foreign_key=>true})  
-> 0.0464s  
== 20180519235150 AddUserToImage: migrated (0.0465s) =====  
  
== 20180519235252 AddRolesToUser: migrating =====  
-- add_column(:users, :user, :boolean, {:default=>true})  
-> 0.1465s  
-- add_column(:users, :admin, :boolean, {:default=>false})  
-> 0.1794s  
== 20180519235252 AddRolesToUser: migrated (0.3261s) =====
```

Рисунок 6.16 –Результат выполнения команды проведения миграций изменений в схеме базы данных.

Веб-приложение полностью подготовлено к работе. После всех настроек необходимо его запустить. Для этого следует ввести в командной строке следующую команду:

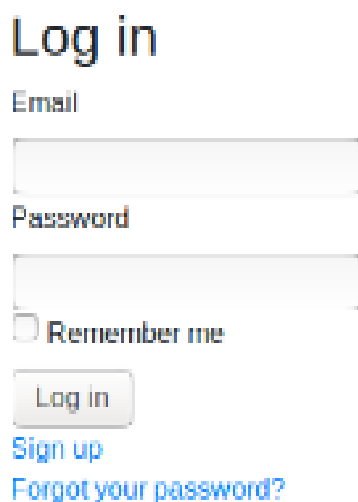
```
rails s
```

Результат выполнения этой команды отображен на рисунке 6.17.

```
alexuslik@alexuslik-N56VV:~/ov7670-webapp$ rails s
=> Booting Puma
=> Rails 5.2.0 application starting in development
=> Run `rails server -h` for more startup options
Puma starting in single mode...
* Version 3.11.4 (ruby 2.4.1-p111), codename: Love Song
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://0.0.0.0:3000
Use Ctrl-C to stop
```

Рисунок 6.17 –Результат выполнения команды запуска сервера веб-приложения.

После запуска приложение доступно для просмотра и взаимодействия с ним в браузере. При переходе по адресу, по которому располагается веб-приложение мы увидим домашнюю страницу, с кнопками «Register» и «Login». После нажатия на кнопку «Login» отображается форма авторизации пользователя, которая продемонстрирована на рисунке 6.18.



Log in

Email

Password

☐ Remember me

Log in

[Sign up](#)

[Forgot your password?](#)

Рисунок 6.18 –Форма авторизации пользователя.

После авторизации веб-приложение перенаправит пользователя на страницу с таблицей, в которой отображаются все фотографии пользователя с дополнительной информацией. Страница, отображающая изображения представлена на рисунке 6.19.

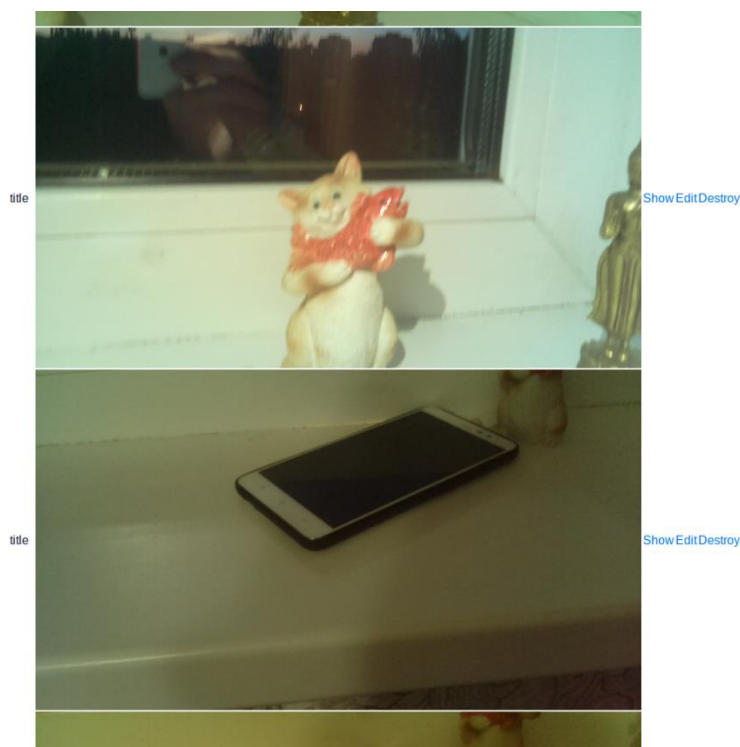


Рисунок 6.19 – Страница, отображающая изображения пользователя.

Если пользователю требуется просмотреть фотографии в реальном размере, необходимо нажать на кнопку «Show». После этого веб-приложение перенаправит пользователя на страницу отображающую определенную фотографию и информацию о ней, как это показано на рисунке 6.20



Рисунок 6.20 – Страница, отображающая отдельную фотографию с информацией о ней.