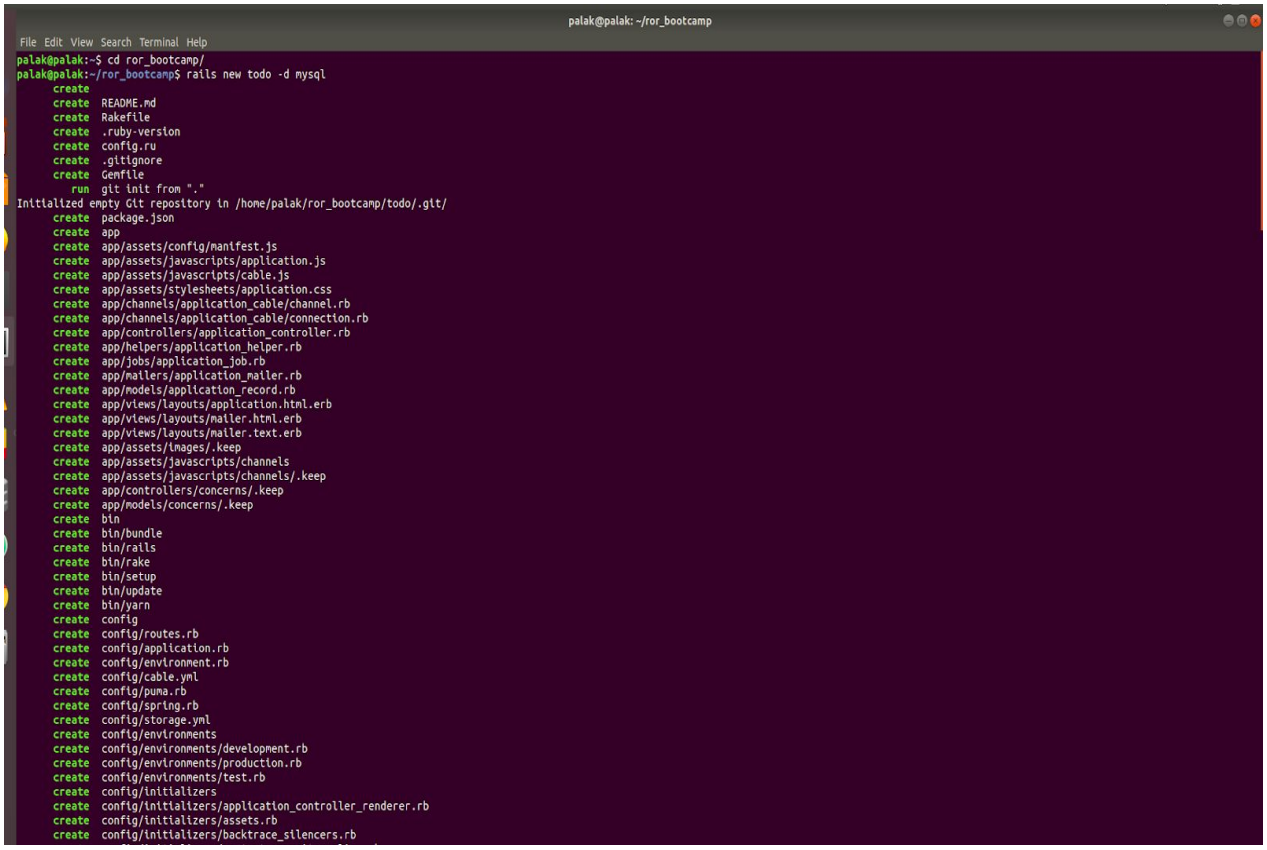
CREATING A SIMPLE TODO APP USING RAILS AND MYSQL

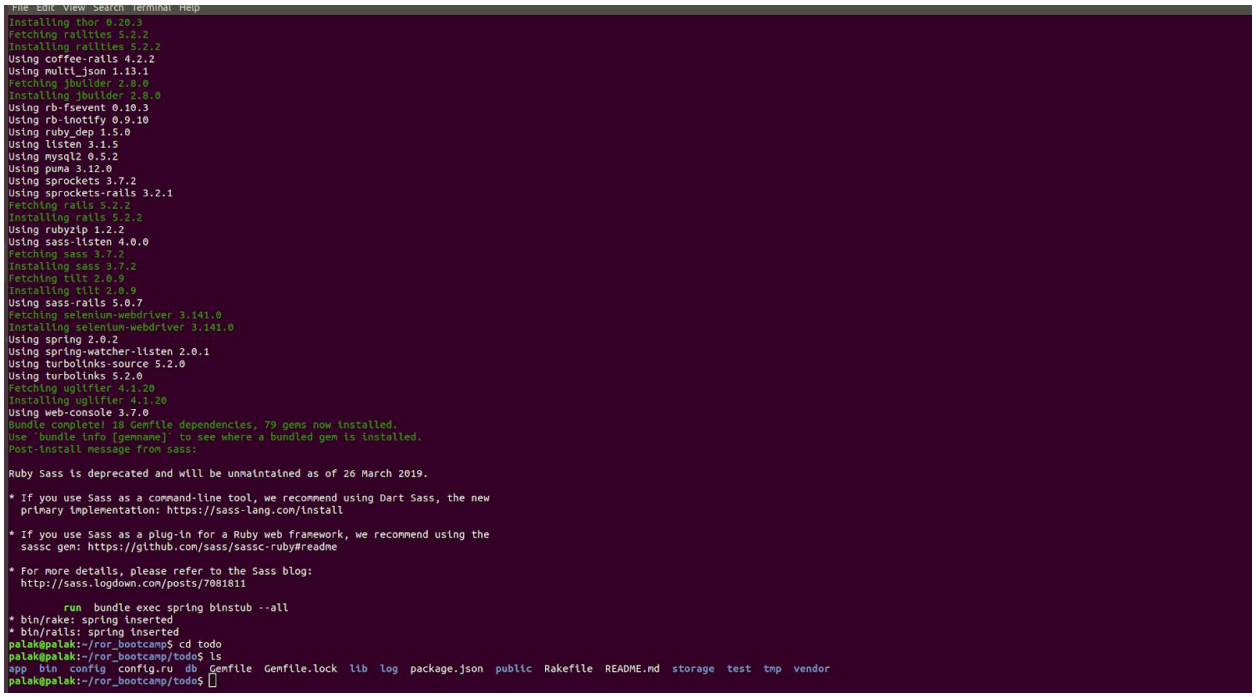
1. Create a rails application with database as mysql . Make sure mysql is already installed.

rails new todo -d mysql



2. Go inside the newly created folder called todo and ls to see the folders and files which got created.

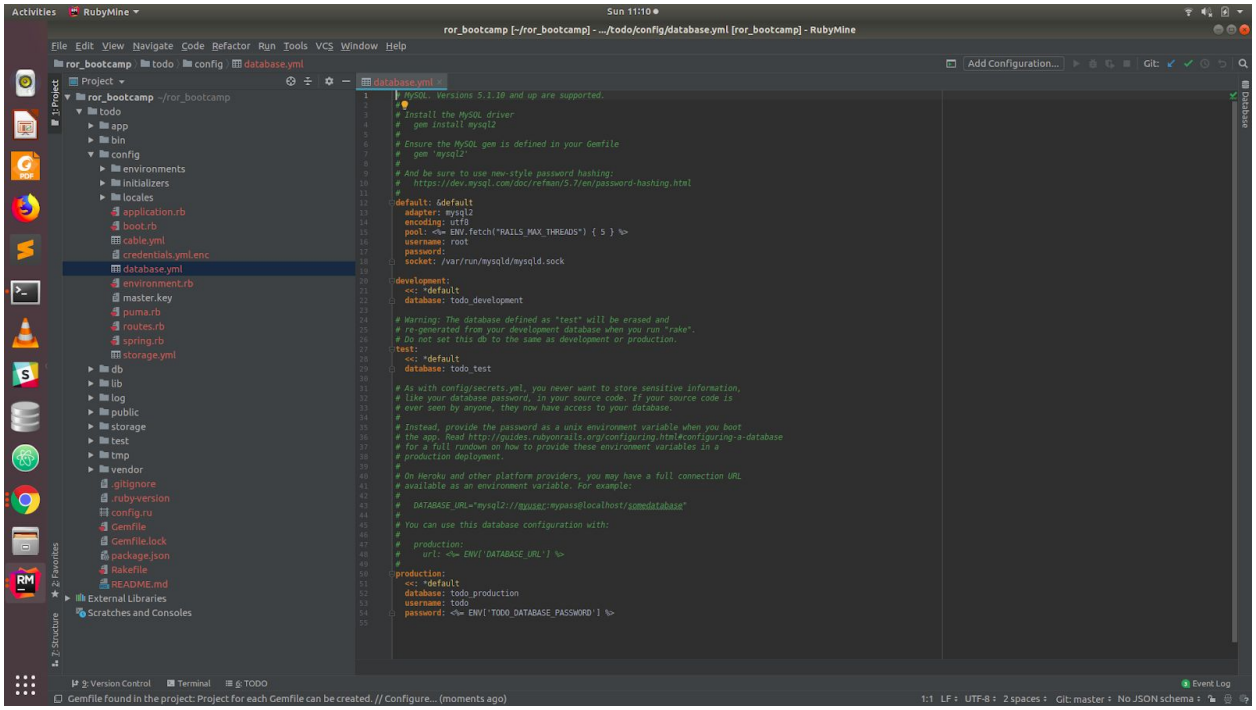
cd todo
ls



3. Now open the folder **todo** in RubyMine or any other text editor you use.

Go to todo/config/database.yml

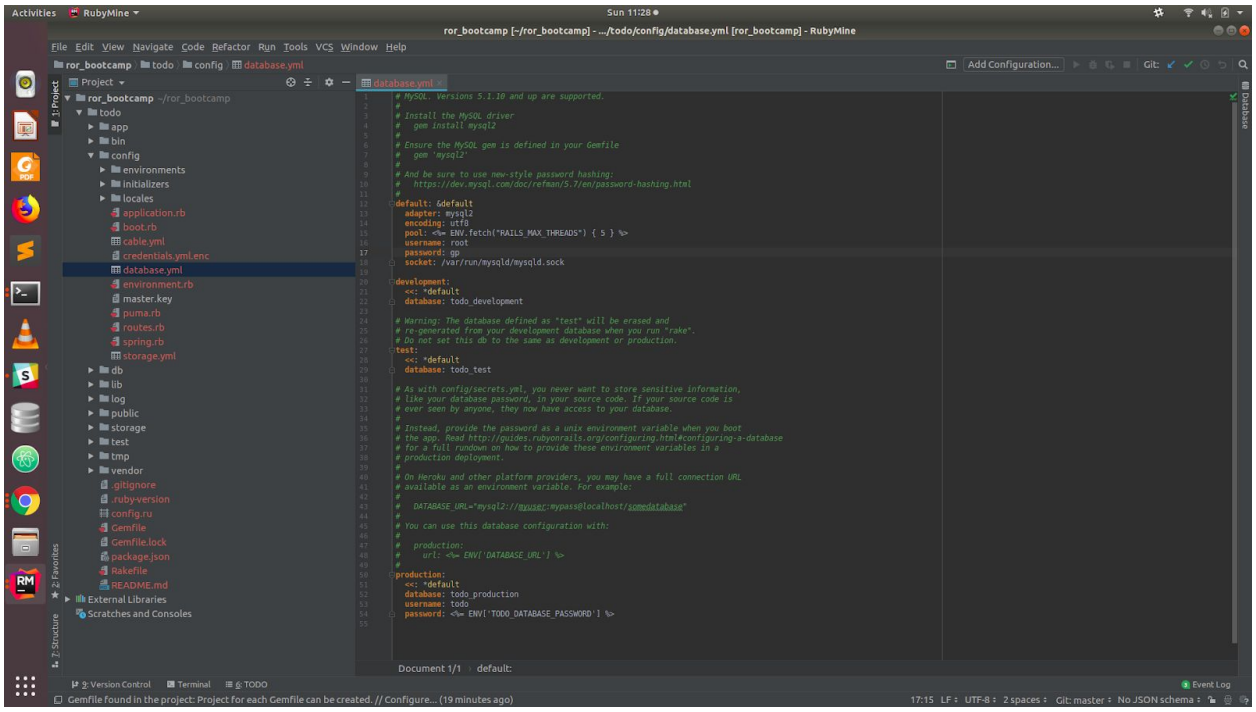
Open the file.



You can see that the adapter says mysql2 which means that the database which we are using is mysql.

4. Fill the username and password field with the username and password you chose while installing mysql in your systems.

Eg. my username and password are
Username: root
Password: gp



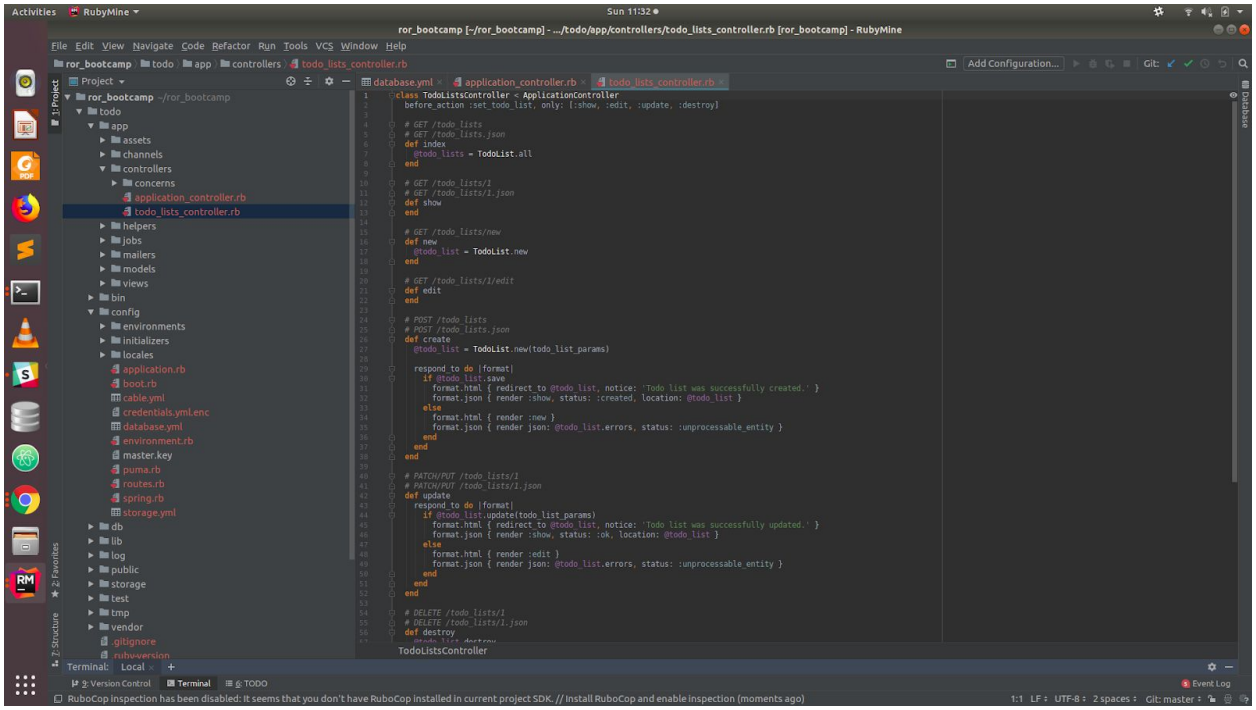
5. After that create a todo_list using scaffold

rails g scaffold todo_list title: string description:text

```
palak@palak:~/ror_bootcamp/todo$ rails g scaffold todo_list title:string description:text
Running via Spring preloader in process 10272
      invoke  active_record
      create    db/migrate/20181209060122_create_todo_lists.rb
      create    app/models/todo_list.rb
      invoke    test_unit
      create      test/models/todo_list_test.rb
      create      test/fixtures/todo_lists.yml
      invoke  resource_route
       route    resources :todo_lists
      invoke  scaffold_controller
      create    app/controllers/todo_lists_controller.rb
      invoke    erb
      create      app/views/todo_lists
      create      app/views/todo_lists/index.html.erb
      create      app/views/todo_lists/edit.html.erb
      create      app/views/todo_lists/show.html.erb
      create      app/views/todo_lists/new.html.erb
      create      app/views/todo_lists/_form.html.erb
      invoke    test_unit
      create      test/controllers/todo_lists_controller_test.rb
      create      test/system/todo_lists_test.rb
      invoke    helper
      create      app/helpers/todo_lists_helper.rb
      invoke      test_unit
      invoke    jbuilder
      create      app/views/todo_lists/index.json.jbuilder
      create      app/views/todo_lists/show.json.jbuilder
      create      app/views/todo_lists/_todo_list.json.jbuilder
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/todo_lists.coffee
      invoke    scss
      create      app/assets/stylesheets/todo_lists.scss
      invoke  scss
      create    app/assets/stylesheets/scaffolds.scss
palak@palak:~/ror_bootcamp/todo$
```

This will create the controller, model, views, migration whatever is necessary. We will have a look at the things it generated for us one by one.

6. **Go to todo/app/controllers/todo_lists_controller.rb**



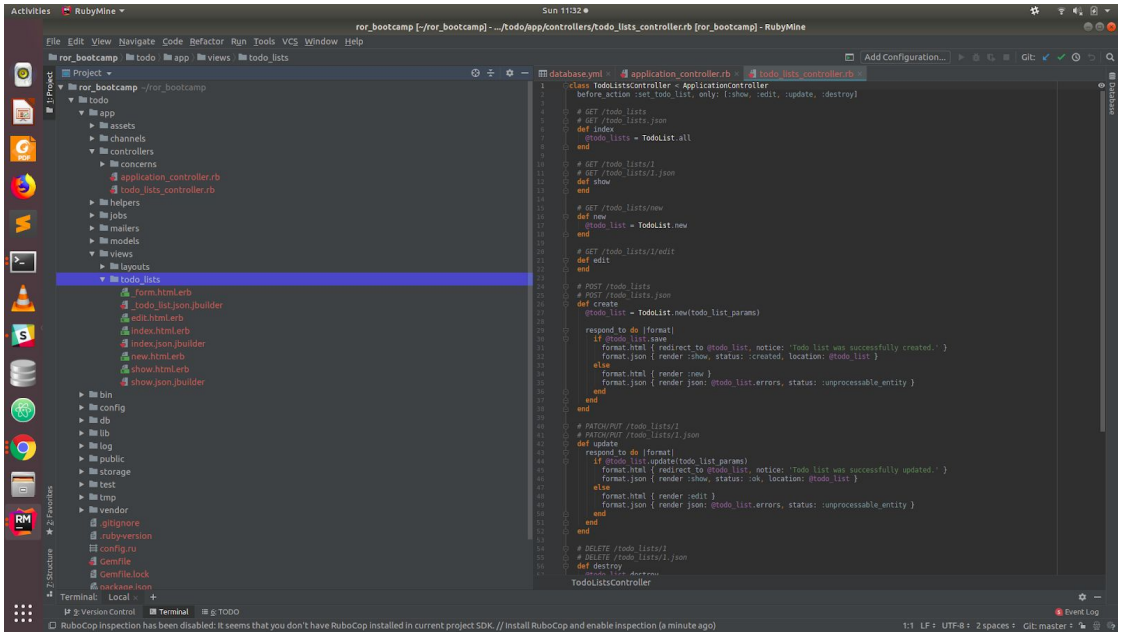
You can see the class TodoListsController which is inheriting from the class ApplicationController in the first line.

The class has all the functions defined which are called actions. They are defined with a keyword def and ended with end. Eg index, show, new etc.

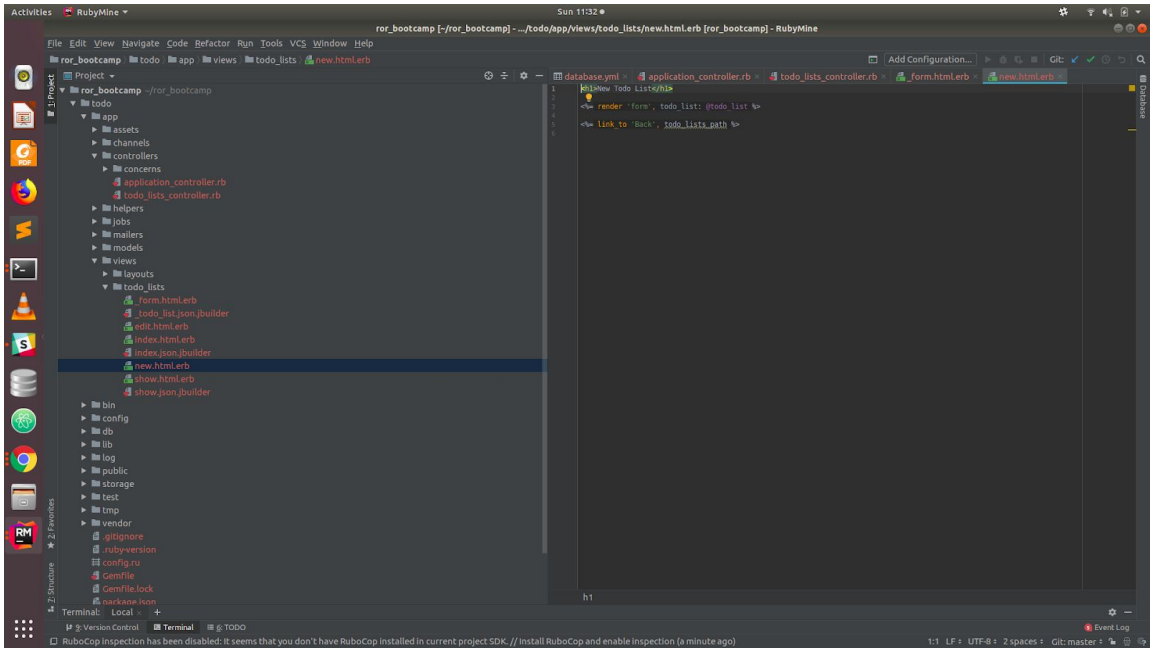
7. **Go to todo/app/views/todo_lists**

Here you can find lot of .html.erb files which take care of how your application appears. .erb refers to embedded ruby which means you can use ruby variables inside the html files but there is a proper syntax for that **<% %> and <%= %>** . The difference between these you can google .

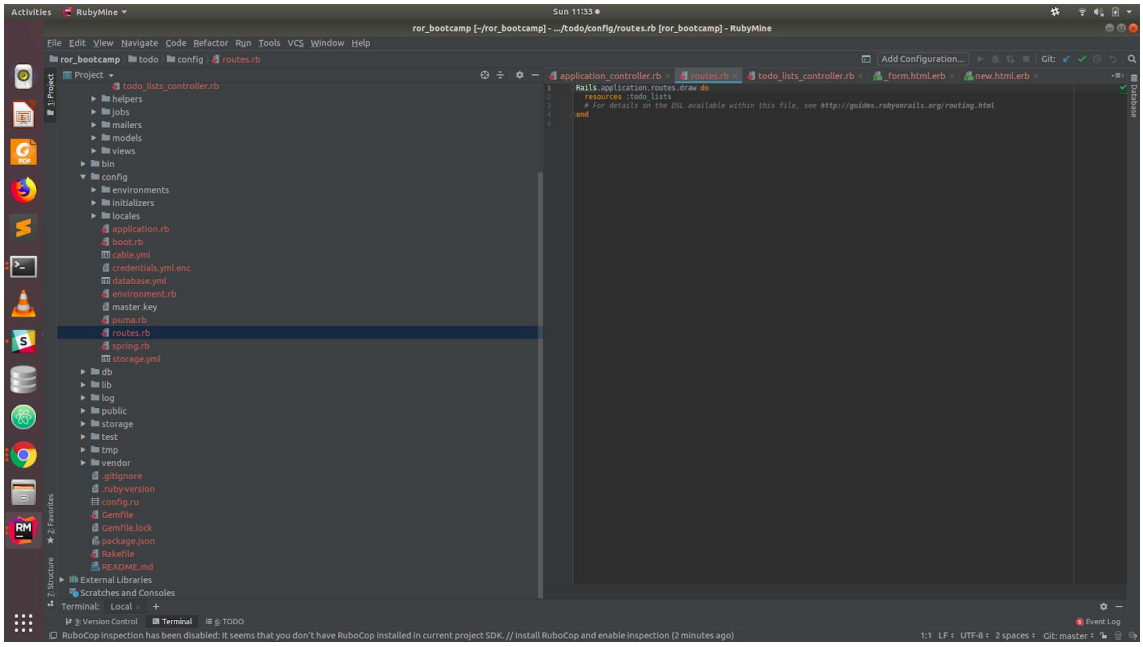
All the html.erb files are files are named same as the action . This is how rails know which html file to render when by finding the corresponding named html.erb file from the views.



8. You can go to each of these .html.erb files and try to understand what's happening there.



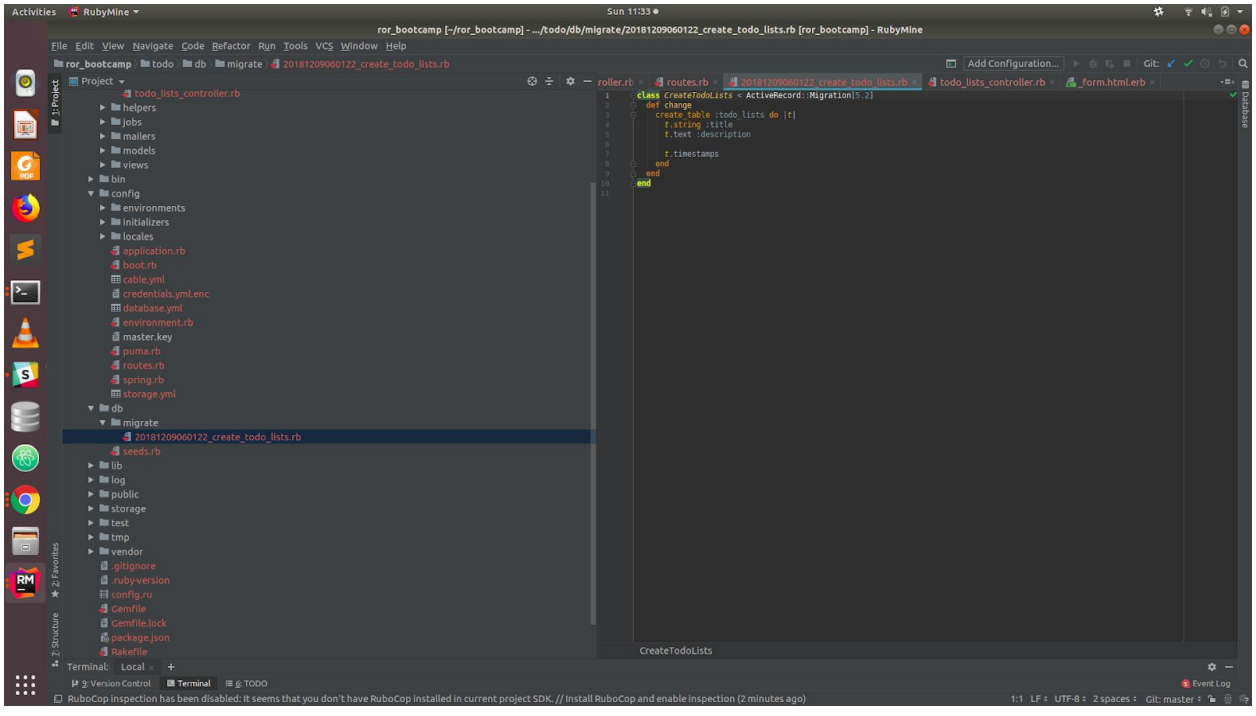
9. **Go to todo/config/routes.rb**



Here all the routes are defined.

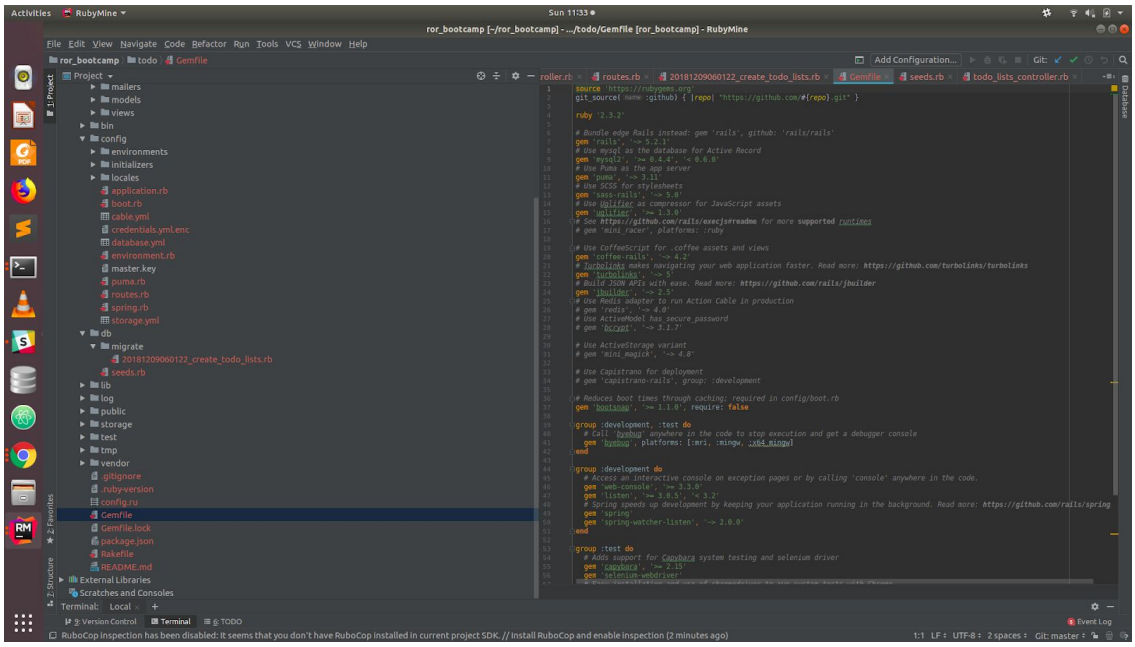
Resource routing allows you to quickly declare all of the common routes for a given resourceful controller. Instead of declaring separate routes for your index, show, new, edit, create, update and `destroy` actions, a resourceful route declares them in a single line of code. A resource route maps a number of related requests to actions in a single controller.

10. **Go to todo/db/migrate** to see the migrations create with timestamp. This got created when we did scaffold.



11. **Go to todo/Gemfile**

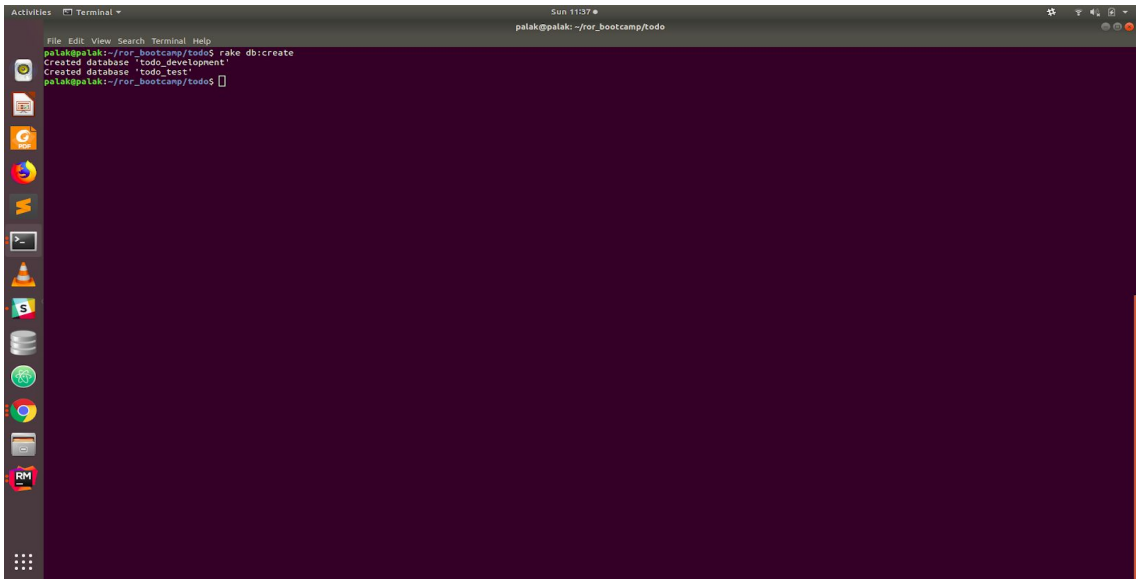
Here you can see all the gems currently installed. In case you want to add some new gems add it here and do bundle install to install them.



12. You need to create the mysql database in your local system. For that go to the terminal

cd todo

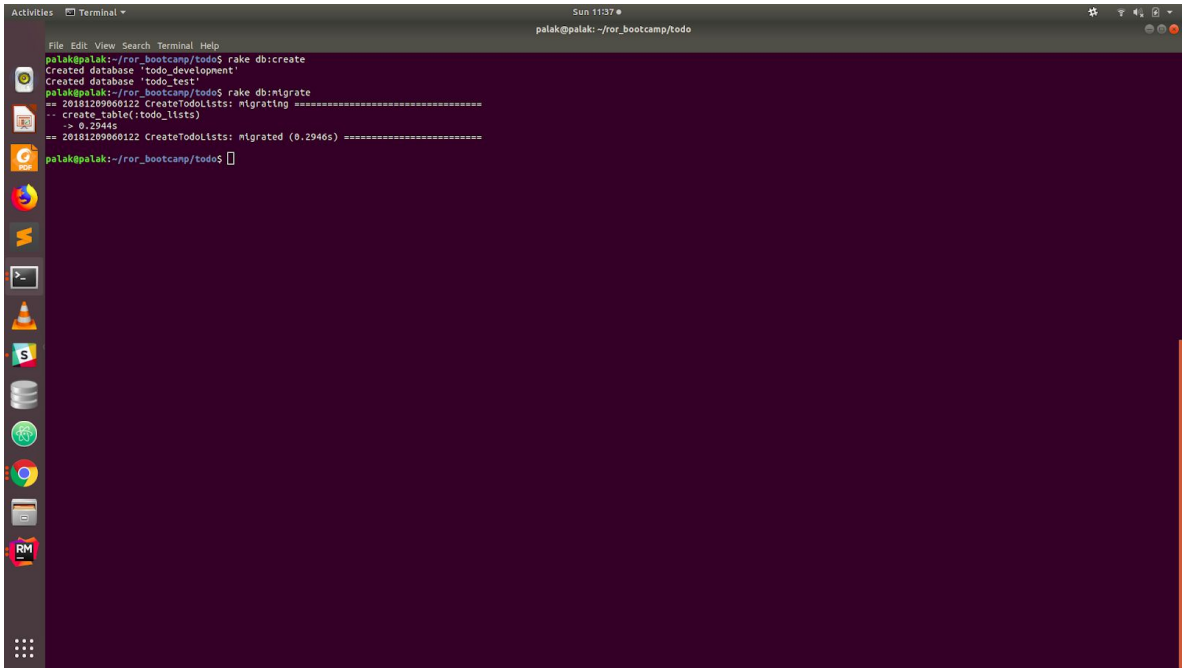
rake db:create



13. You need to make the tables actually appear in your local database. The migration created in the application won't reflect in your database until you run the following command

cd todo

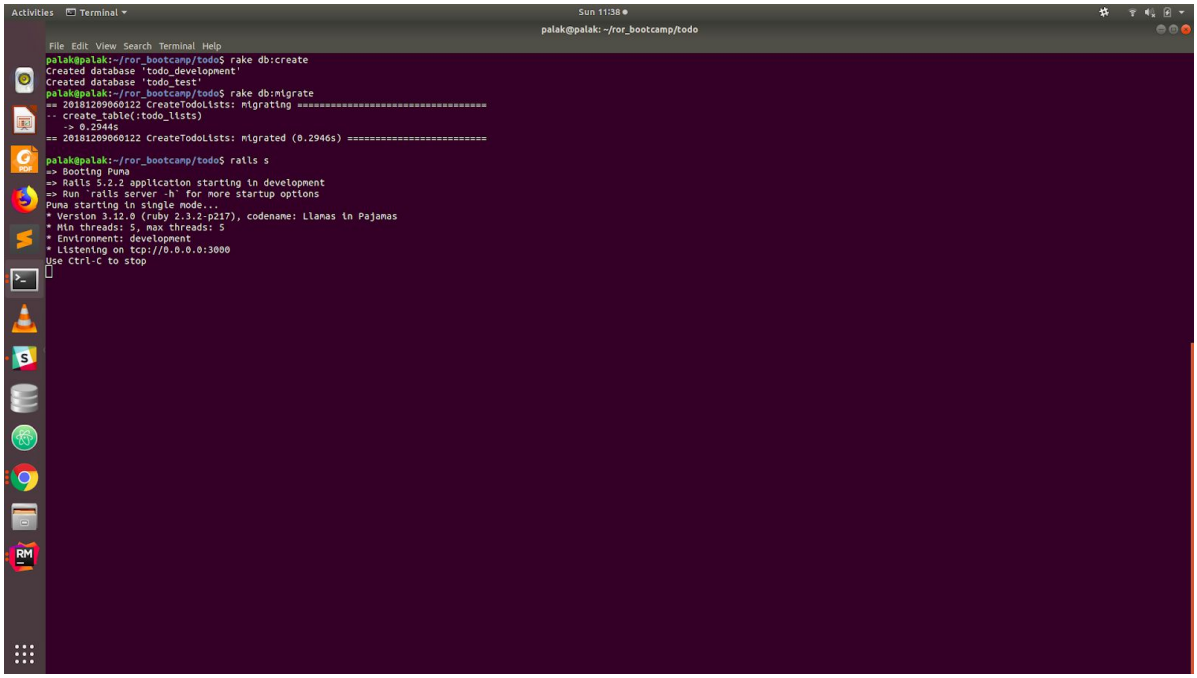
rake db:migrate



14. To see your application running

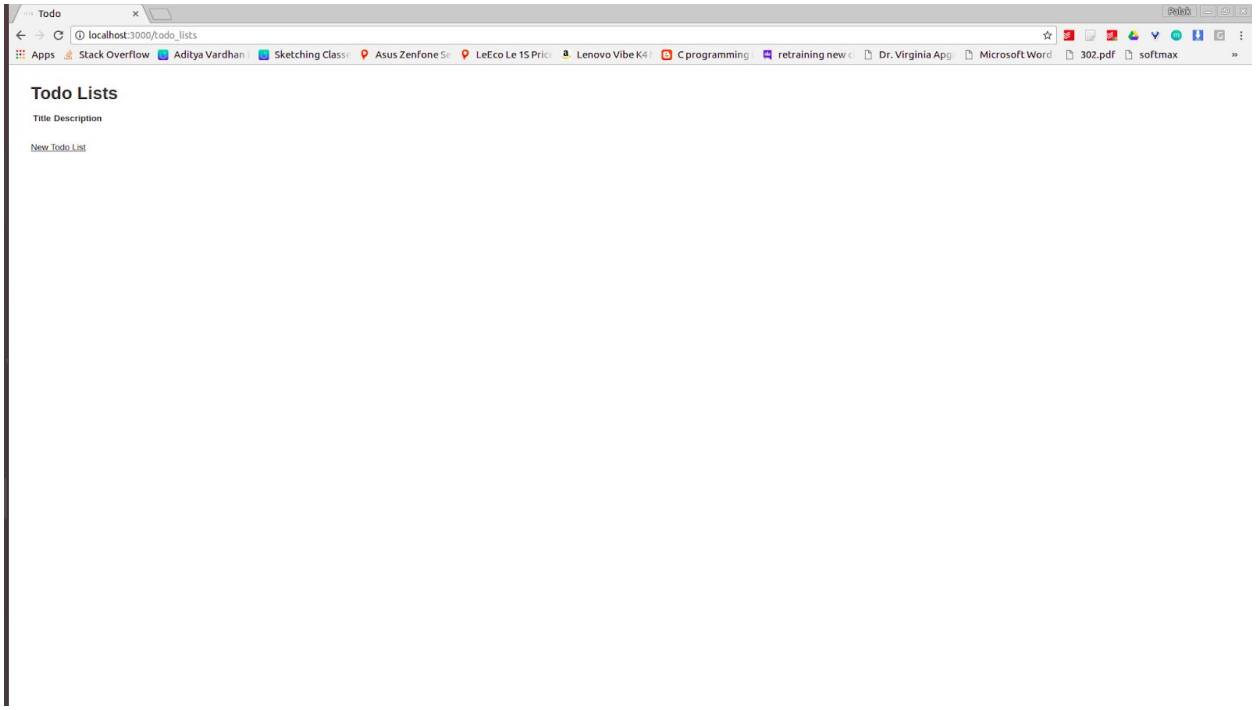
cd todo

Rails server or rails s



15. Open browser and go to

localhost:3000/todo_lists



14. You can also visit
http://localhost:3000/rails/info/routes

This will give you information about all the routes defined under the resources section which we saw in routes.rb file and the corresponding url and controller as well as the action of that controller associated with a particular url.