Bachelor's thesis

Degree programme in Information Technology

2018

Bhim Chhetri

WEB PAGE DEVELOPMENT USING THE RAILS FRAMEWORK

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Information Technology

2018 | 40

BHIM CHHETRI

WEB PAGE DEVELOPMENT USING THE RAILS FRAMEWORK

The thesis is based on Ruby on Rails web framework. The primary objective of the thesis was to become acquainted and understand the theory of the Ruby on Rails framework. The secondary objective of the thesis was to create a website as an practical approach using the Rails framework. There is a short overview of the website that was made as a part of the thesis project at the end of thesis. The thesis explains the beneficial features and tools of the Rails framework. The manner in which these tools and features enable to make better web application is discussed concisely. The thesis also highlights the principle of Ruby on Rails and its surface optimization dependency. The thesis gives emphasis on the vital principles of Rails which makes it attractive and distinctive. The Convention over Configuration, the Do not Repeat Yourself and the MVC principles are explained. The above mentioned principles were taken into account while making the thesis website project. This thesis discusses the range of web threats in web application and their respective countermeasures. Apart from this, various Rails tools and methods that are used to test the performance of rails application are discussed. The scalability issue of the web application in the growing business world is also discussed. The corresponding Rails technique is explained to cope with existing scalability issue of the web application.

KEYWORDS:

SQL injection, Caching, Debugging, Scalability, Optimization, Benchmarking, Profiling

CONTENTS

FIGURES

LIST OF ABBREVIATIONS (OR) SYMBOLS

API	Application programming interface (https://en.wikipedia.org/wiki/Application_programming_interface)
MVC	Model-view-controller (https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller)
DRY	Do not repeat yourself (https://en.wikipedia.org/wiki/Don%27t_repeat_yourself)
CSRF	Cross site Request forgery (https://en.wikipedia.org/wiki/Cross-site_request_forgery)

1 INTRODUCTION

Websites are the modern means of information exchange in business industry, university, health care and many more sectors. Web page development was introduced in 1990 (Emery 2016). Recently, there are web frameworks that facilitates the speedy development of web application and websites. The central idea behind every framework is to give the programmer a specific skeleton or raw sketch based on which web page are developed, according to requirements. In other words, a web framework provides programmers with support programs, tool sets, libraries and API that are required to develop web pages without working from scratch. In web page development, a framework helps to save time and uplift efficiency of any project. Sinatra, Angular, Spring, Bootstrap, Laravel, Rails are some of the web frameworks that have gained the popularity among the web developers (Hotframeworks.com 2018). Every framework that has been in existence in web world is created on the specific set of principle. A principle is an adaptation of idea or concept that determines the behavior of the framework. The software created from that framework does eventually follow the same principle. Clear understanding of principle means making the best website product from the related framework.

The objective of the thesis is to learn about the Ruby on Rails framework and create a website using the Rails framework. In the beginning, Different aspects of Rails framework such as Ruby programming language and its useful feature, Rails debugging method, caching in Rails is discussed. Rails tools and its useful implementation during website development is also explained. The principles adopted by Rails framework such as MVC, the Do not Repeat Yourself and Convention over Configuration is thoroughly discussed in chapter two of the thesis. The final chapter of the thesis examines the various factor affecting the Rails optimization. The Rails security system, performance issue and scalability issue has been discussed under the Rails optimization chapter. In the end, there is a short overview of the website that was made as a part of the thesis project.

2 WHY RUBY ON RAILS ?

The selection of web framework from a list of many is a difficult job. A new web programming learner has to go through dilemmas in picking the most appropriate web framework option, especially when one has no prior programming language. Initial phase research will always be helpful in picking the right web framework. Web API, programming language involved, tools, libraries and community size are a deciding factor in opting for a web framework.

Ruby on Rails is a popular web framework in present day web community. This framework uses ruby as the programming language. Commonly known as rails, among the rails community, this framework was first introduced in July, 2004 (Grimmer 2018). Rails has Libraries, tools, API, web resources and various other feature needed to make required web application and website. The deciding factors to pick rails framework are discussed below.

1.1 Ruby as core programming language

Ruby is the native programming language for rails. Ruby was created by Yukihiro Matsumoto (Stewart 2017). Ruby was publicly released for the first time in 1995. The development of ruby has been greatly influenced by other pre-existing programming language such as Perl, Small talk, Ada and Lisp. According to Yukihiro, the simplicity of ruby is adopted from lisp, the object oriented idea is conceived from Small talk and the blocks needed for higher function is influenced from Perl. (Ruby-lang.org 2018.)

In 1995, ruby made a first public release. Ruby 0.95 was the early version whereas, ruby 2.4 is the latest version. The features of ruby language are listed below.

- The simple appearing nature of ruby makes it easy for coders to read and write.

- Ruby has exception handling. In a code, try and catch statements can be used to find whereabouts of existing error.
- Ruby is a truly object oriented language. Everything in ruby is considered as object. Every bit of code can have their properties and methods. It has meta classes, inheritance and mixin properties to inject objected oriented concept in rails.
- Garbage collection system is another feature ruby that can be applied to all objects.
- Web developers does not need to worry about the operating system platform. Ruby is cross platform and can be developed in Linux, Mac or Windows.

1.2 Rails built-in database and components

Rails comes with an embedded SQLite database. SQLite is the default database in rails However, MYSQL or PostgreSQL can also be implemented as an alternative database . The SQLite database is fairly easy to use and needs almost zero configuration during the development phase of database in rails. Rails can use the database in three different environment. The development environment is used while developing application, the Test environment is used for testing purpose and the Production environment for deploying application in real world. (Guides.rubyonrails.org 2018a.)

SQLite is an open source library with the right to use and modify the code. The local disk file is used to read and write data as it is a server less, self-contained database. Files can be easily transferred from 32 bit system to 64 bit systems and vice versa. This feature of SQLite enables the convenient data transfer wherever it is implemented. (Sqlite.org 2018.)

Rails possesses a range of components for assisting in web development. Action controller, Action view, Active record, Action mailer are some of the

components in rails (Guides.rubyonrails.org 2018b). The respective responsibility of these components are listed below.

- Action Controller: Rails is designed under MVC concept. The Action Controller component handles the 'C' or Controller of web application in rails. The web pages-related action such as incoming requests, extracting parameters and sending them to destined action is supervised by Action Controller. The session management process for the client is another job of Action Controller.

- Action View: The Action View component in rails can create XML and HTML file by default. The successful end users request for the web page end up with page rendered in the client browser. This view can be as a result of HTML file or XML file generated by the action view component. The Action View component of rails is responsible for rendering the page in the desktop

- .Active Record: The Active Record is related with model part of model-view-controller structure. Model is referred as database in the surface level understanding. Active Record wires the link between two or more models in an application. Database related to create, update, read and delete functionality is another management part of this component. This functionality is also known as CRUD. It also plays vital role in maintaining the database of web application independently.

- Action Mailer: Modern day web application usually have email service and support. The Action Mailer in rails is used to build email services for the end users of the rails application.

- Active Support: Rails websites or applications need much support in the code and during the production phase. The Active Support is the set of ruby library extension and group of utility classes to aid websites or application in rails.

1.3 Rails command line utility

Rails has all the command line tools required to develop web application. The Rails command line utility comes with the installation of the rails framework. Command line tools are handy when a developer has to interact with the web application. In case the needed command line tool is not available in rails framework, one can simply install it by running following command.

```
Gem install  'tool-name'
```

Rails, server, generate, destroy are few command line tools used in rails (Guides.rubyonrails.org 2018c). The use and description of the above mentioned command line utility are sighted below along with their use.

- Rails: The Rails is one of the widely used command line tool among rails framework community. It is handy whenever, a web developer has to create a new application. The example of the use of this tool is as follow. rails new application-name;

This command creates a new application with name 'application-name'. The Rails command sets up a basic building layout by generating files and folders of the newly created web application. The Rails utility is used in combination with other command tools to complete many actions.

- Server: Rails framework uses different server such as WEBrick and Puma to render web pages in the browser. The Server command utility is used in the rails framework to start and stop a server used by the rails application. This command comes along with the installation of ruby. This utility is used in rails application in the following way.

  ```
  rails server;
  ```

- Generate: The Generate utility in rails application is helpful in many ways. The Generate utility can create a controller, view and model templates by running the following piece of command in console window or terminal.

```
rails generate controller welcome;

rails generate model Comments title:string text:text;
```

This will generate a controller named Welcome for a rails application. The following line will create a model or database named Comments. The Comments model has a two column title and text and the data type is string and text. The Generate command tool is extremely handy and saves web developer time from doing extra task.

- Destroy: The Destroy is another important tools in rails framework. It is basically the opposite of generate. This utility comes in handy to undo all the generated piece of work done using generate command tools.

```
rails destroy controller Welcome;
```

This command line wipes out controller named 'Welcome' from rails application.

Console, rake, plugin, about and **rake** are some of the other command line tools in rails to interact with application and perform many desired task. In conclusion, it is important for any rails developer to explore this command line utility in the best possible way. The clever use of this command line tool can help to develop web application in a speedy way. The task of rails developer becomes comparatively easy by gaining clear knowledge on applying the rails command line utility.

1.4 Rails caching

Caching is the process of storing previously computed data in the hardware or software component generated by request-response sequence (En.wikipedia.org 2018b). The modern browser such as Mozilla, Firefox implements the cache system. The objective of the cache implementation is to serve the requested data straight from the browser cache. Caching is implemented in rails application to speed up the requested data transaction.

Rails caching helps to serve thousand of application users hence, improving the performance of application. Rails exercises three basic caching technique : fragment, page, and action caching (Guides.rubyonrails.org 2018d).

- Fragment caching: The Fragment caching is the default technique in rails application. Dynamic web pages are built of different components or parts. Fragment caching is applicable whenever different parts of the page needs to be cached individually. This technique uses unique id and time stamp to keep the record of the cache files. The time stamp takes care of the updated portion of the page. New Id is generated and saved in cache file every time specified portion is updated.

- Page caching: The Page caching is suitable to cache the whole page of the website. Dynamic web pages are built of different types of pages according to the need. Some pages are related to information whereas, some are related to authentication. Page caching is applicable for those pages where action of the users is not needed. Web pages that serve the information do not need any kind of user action . In this case page caching is wisely used. This caching is enabled in rails by setting true value in the local environment .

```
config.action_controller.perform_caching=true;
```

- Action caching: The Action caching is relevant where page needs user action. Pages involving authentication is where user needs to do some action. This scenario is perfect to apply action caching. The following setting should be enable in local environment to enable action caching.

```
config.action_controller.perform_caching=true;
```

The Caching concept was developed for the speedy serving motive of web request by storing data in local browser. In conclusion, Rails is feasible to develop high performing rails application by engaging rails caching technique.

1.5 Rails debugging

Rails provides a number of debugging options to find and eliminate the existing error in the code block. Debugging in rails is executed for different purposes. Inspecting variables and objects and locating breaking point are some areas of debugging . Rails facilitates debugging through view helper, Logger, bye bug gem, breakpoints and exceptional handling. These mentioned debugging options are briefly discussed below.(Guides.rubyonrails.org 2018e.)

- View helper: The View helper debugging uses three different ways to inspect the object used in code. Debug, to_yaml and inspect are three different debugging option in the view helper. The Debug helper renders the object using the human readable YAML format. Errors are located by inspecting the readable file of the particular object considered for error cause. The Inspect helper is used to identify the error associated with arrays. The inspect method is injected in the array identified for debugging. (Guides.rubyonrails.org 2018e.)

- The Logger: The Log file is created by rails while building a first new application. This log file stores information about the run time environment of the application. In other words, the error information that appears during run time is assigned to this file. The 'ActiveSupport Logger ' class is used to record log information in rails. Rails uses this log file to debug the error occurred. (Guides.rubyonrails.org 2018e.)

- Bye bug gem: The Bye bug in rails is installed by running 'gem install Bye bug' command. Bye bug can be executed in any source code of the application to set the breaking point . The invoking is made by calling a Bye bug method. (Guides.rubyonrails.org 2018e.)

- Catching exceptions: Rails is provided with try and catch expression to identify error within the source code. The Try and catch statement can be used where ever coder has to identify the problem. It is mostly used to find exception type error in the code. (Guides.rubyonrails.org 2018e.)

Thread and Breakpoints are another debugging option in rails to locate errors. Rails debugging options makes it easy to debug the bug of different nature. Hence, error isolation and elimination using rails debugger makes the life of web programmer easy.

1.6 Rails Deployment

Rails is a platform independent web application frameworks. Rails can be deployed in different operating system like Linux, Windows and Mac OSX operating system. A web programmer do not have to worry about the operating system used. A package manager called 'Ruby gem ' is used to install rails. A command provided by Ruby Gem package to install rails through command line terminal is illustrated below.

```
gem install rails;
```

Rails installation is verified using command 'rails --version' (Guides.rubyonrails.org 2018f). The version command display the rails version number installed in the system.

The Database is an essential part of web application to store data. Rails comes with default SQLite database. Accordingly, SQL or PostgreSQL could be used if needed. Apart from this, the application developed using rails can find a range of hosting company. Amazon web services, Heroku, Digital ocean are some of rails application hosting companies. Besides hosting companies, There is a range of choices for a hosting server. Phusion passenger, Rack, Puma, Thin, Unicorn are hosting servers for rails application (Digitalocean.com 2018).

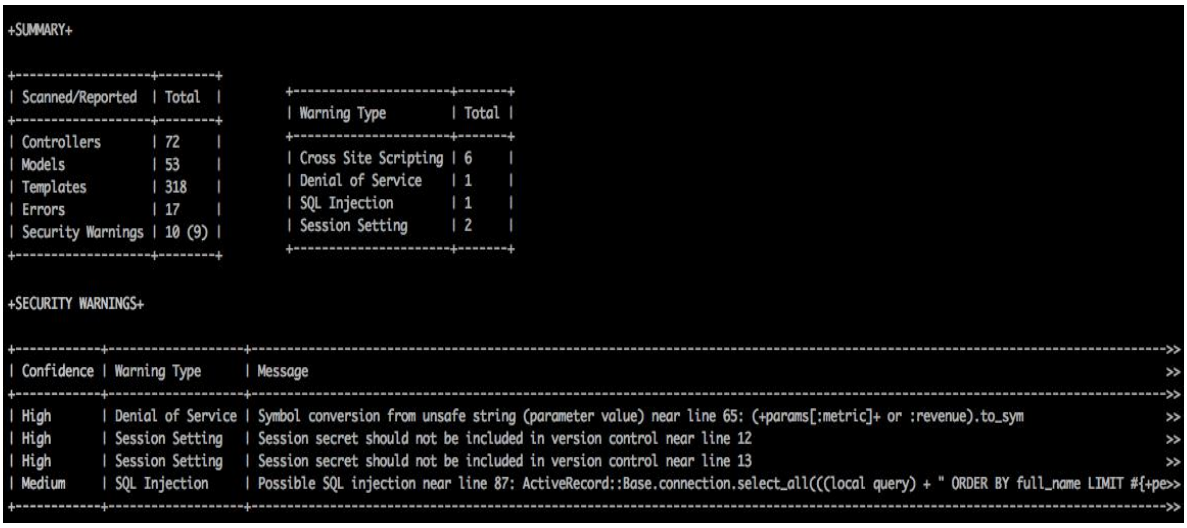
1.7 Rails Tools

Rails is sophisticated and bear enough tools that helps rails developer build a productive web application.

Tools are needed in different phases of web development to add more features in an application. Rails tools can be installed using the Ruby Gem package. Rails is well equipped with tools for abstraction, admin interface, analysis, authentication, command line interface, code analysis and metric and many more. Some of the popular tool used for rails development are Rubocop, ruby critic, trace route, rack-mini-profiler, Brakemen, devise and warden (Infinum.co 2018).

- Rubocop: Rubocop is a static code analyzer. It makes sure the code written is as per the ruby community code guidelines. Code violations are reported through the command line interface.
- Ruby critic: Ruby critic reports code quality. This tool aids to make properly structured HTML files.
- Trace route: Trace route is used to find dead routes and action in the controller.
- Warden: Warden is used for authentication purpose for example login.
- Devise: It is an alternative tool like warden, if a developer wants to make any authentication system. Web developers are able to customize Devise to meet the requirement of authentication structure (Awesome-ruby.com 2018).
- Rack mini- profiler: The Rack mini profiler testing tool is used for speed test in rails application. It shows how long it takes for a request to be processed or for database enquiry. This type of tool is used to check and maintain the performance of application.
- Brakeman: Brakeman is a security analysis tool for rails application. Applications are scanned and security issue are grouped according to their risk level that is low, medium or high.

Picture 1 below is security analysis output of brakeman tools.



```
+SUMMARY+

+-------------------+--------+
| Scanned/Reported  | Total  |
+-------------------+--------+
| Controllers       | 72     |
| Models            | 53     |
| Templates         | 318    |
| Errors            | 17     |
| Security Warnings | 10 (9) |
+-------------------+--------+

+----------------------+-------+
| Warning Type         | Total |
+----------------------+-------+
| Cross Site Scripting | 6     |
| Denial of Service    | 1     |
| SQL Injection        | 1     |
| Session Setting      | 2     |
+----------------------+-------+

+SECURITY WARNINGS+

+------------+-------------------+------------------------------------------------------------------>>
| Confidence | Warning Type      | Message                                                          >>
+------------+-------------------+------------------------------------------------------------------>>
| High       | Denial of Service | Symbol conversion from unsafe string (parameter value) near line 65: (+params[:metric]+ or :revenue).to_sym >>
| High       | Session Setting   | Session secret should not be included in version control near line 12 >>
| High       | Session Setting   | Session secret should not be included in version control near line 13 >>
| Medium     | SQL Injection     | Possible SQL injection near line 87: ActiveRecord::Base.connection.select_all(((local query) + " ORDER BY full_name LIMIT #{+pe>>
+------------+-------------------+------------------------------------------------------------------>>
```

Picture 1 Brakeman security analysis (Source : https://infinum.co/the-capsized-eight/top-8-tools-for-ruby-on-rails-code-optimization-and-cleanup)

1.8 Rails Security

Rails is well aware of various kind of web threat. Rails recognizes the security threats in different layers of rails application. Rails is able to mitigate the web risk related to Session hijacking, session fixation, SQL injection, replay attack for cookie store, cross site request forgery, intranet and admin security threats. Rails applies various countermeasures to prevent websites from being tampered.

Rails also discusses how an application developer needs to apply this technique to make threat intolerable applications. The detail discussion of rails security is made on the Rails optimization topic.

In conclusion the Ruby programming language and its features, rails command line utility, debugging and caching features in rails, rails deployment and security help to make a proper website using rails framework.

2 RAILS PRINCIPLES

A principle is the fundamental guidance followed by any framework. The principles helps to sketch the working map. In other words, principles govern the operating behaviour of framework. Web developer has to understand the founding theory clearly to work with rails. Rails is designed by molding distinct concept into one platform. The rails principles are discussed in the following section.

2.1 MVC design (Model-view-Controller)

The MVC was implemented for the first time in 1970 by Trygve Reenskaug for small talk -76. Model-view-controller shortly known as MVC, is architectural design applied for user interface design. MVC pattern took various form considering different context like model-view-presenter, model-view-adapter, hierarchical model-view-controller after its introduction. MVC dissect user interface application into three interdependent section. The three section has its own data and information. Information from these section are wrapped together and presented in the user interface for the final view.(En.wikipedia.org 2018c.)

Rails has adopted MVC structure to design its web application. Following MVC principle, rails divides its application into three stand alone components. In Rails M stands for model and handles user data. V component know as view manages the user interface. C stands for controller component and it handles the control logic in web application. This three component model, view and controller works hand to hand to display end result in the browser.(Goodrich 2018.)

Rails simplify the web application by applying MVC design and separating user data, control logic and user interface.

Figure 1 illustrates the MVC architecture.

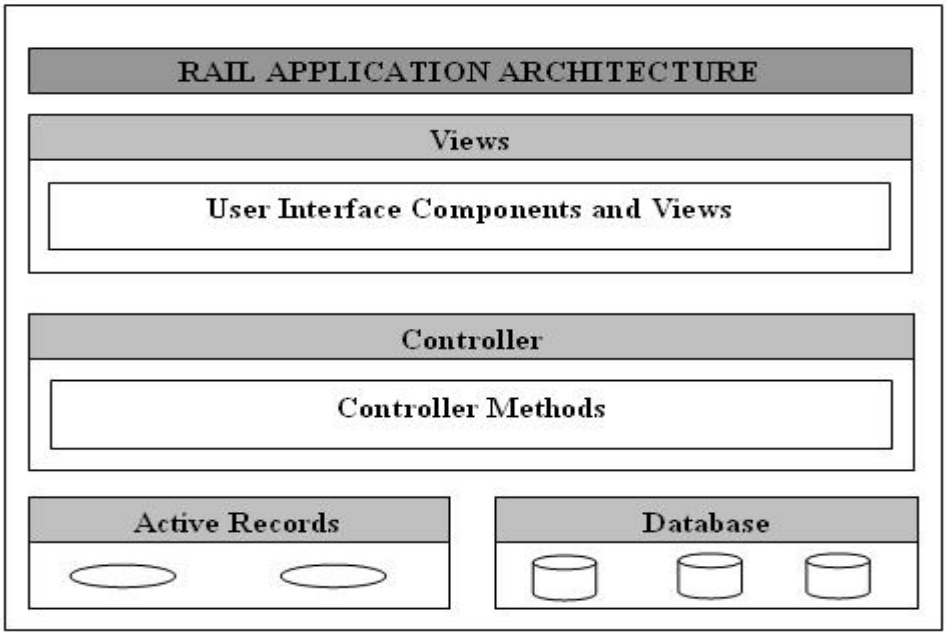


Figure 1 MVC architecture. (Source : https://www.tutorialspoint.com/ruby-on-rails/rails-framework.htm)

The three component of MVC architecture are described below according to their roles.

- Model: The Model is the vital component in rails MVC architecture. The model is associated with data and information of web application. In simple words, model refers to the database of an application. A Model component in rails is implemented in active record library (www.tutorialspoint.com 2018). The Active record library is the Application programming interface that provides binding and interface required to maintain the relationship between database table. In general, a model is responsible for storing data that user input. The model retrieves data from database and direct it to the user interface, whenever there is query for particular data.

- View: The View represents the user interface in MVC structure . HTML,CSS,JSON makeup users view in the screen. Web application deals with the collection of data. This information are displayed in user browser in particular format. A view component of rails is responsible for presenting

this information in specific format. It is implemented in rails by the Action View library. The template required to show the data in particular format is defined in Action view library. (www.tutorialspoint.com 2018.)

- Controller: The controller is the brain or coordinator of the three component. A model and view work under the supervision of controller.In other words, it is a middleman between view and model component. It handles the transaction of data between view and model. The controller is implemented in action controller library. The user request, user data submission and fetching in this triangular architectural design is handled by controller.

2.1.1 Benefits of MVC architecture

Adaptation of MVC pattern and division of MVC architecture into three separate components has several benefits. The development and deployment of website becomes fairly easy. The beneficial point of implementing MVC pattern are pointed below.

- Easy maintenance: The MVC pattern divides application into three distinct section. The clear map of application in the mindset of developer helps to reach the component where the error occur. The fixing of bug is easy if developer knows the specific component zone. In short, One component can be assessed and maintained with out affecting the other.
- Reusable data and code: The model is the storage for the data. Web application uses data from a model to prepare the view for the query made. Hence, the model data can be used multiple times for same kind of query.

2.1.2 Working theory in MVC architecture

Figure 2 depicts the working theory of MVC pattern adopted by rails.

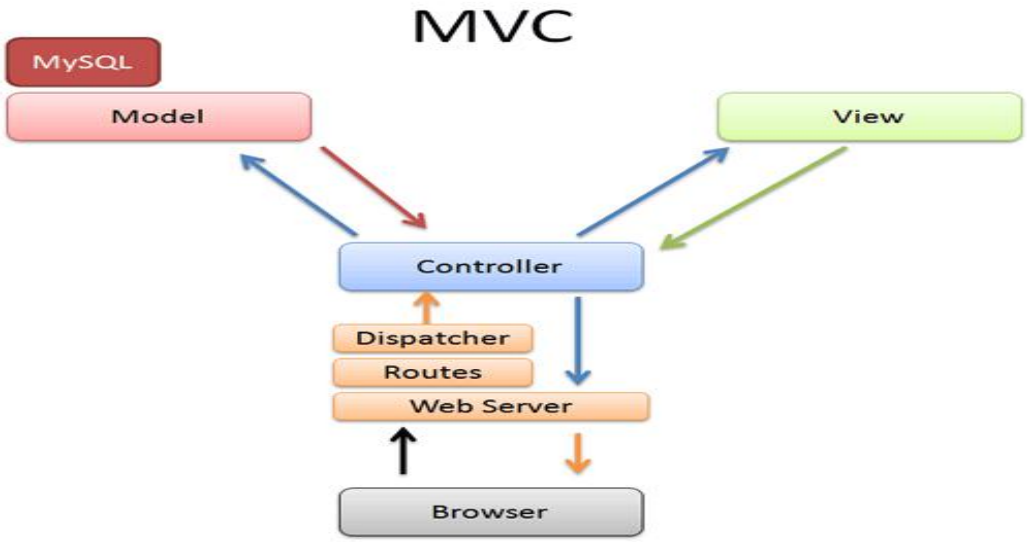


Figure 2 MVC working theory. (Source : https://betterexplained.com/articles/intermediate-rails-understanding-models-views-and-controllers/)

MVC working theory Explanation: Whenever, the query is made using URL in the browsers. The rails server such as Mongrel, Puma or WEBrick receives the client request. The server checks the routes name in the file called 'config/routes.Rb. This file is created when the new application is created using 'rails new' command. The server maps the controller and associated action related to requested query. When the user request reach the associated controller, it assess if the particular action needs data from the model or not. If data needs to be fetched, controller order data from model and wrapped it with template from view component. The final view is rendered to the users screen in the form of HTML or XML file in the browsers screen. (Betterexplained.com 2018 .)

2.2 Convention over configuration

Convention is the distinct sets of concepts or thought pattern applied to complete specific task. Configuration is the manual ideas or input to do a task. Convention over configuration is the design logical rules adopted by rails

framework to minimize the decision making workload of developer with out tampering the flexibility of an application. This philosophy was introduced by David Heinemeier Hansson in rails framework. The idea such as “sensible defaults” and “principle of least astonishment” in user interface design resembles the philosophy of convention over configuration adopted by rails. Convention over configuration saves time and ease the workload especially during the creation of controller and model in the application files. (En.wikipedia.org 2018d.)

The explicit configuration in rails is only required when one deviates from convention over configuration philosophy. When a controller and model is created for object called ‘article’ using convention over configuration. The first alphabet is capital and name is plural for controller. The controller for article is written as ‘Articles’. The model for the article object automatically becomes ‘Article’. The first alphabet has to be capital and name is singular. The corresponding name of the database table becomes ‘articles’. In conclusion, Convention over configuration is the logical way of doing things.(Guides.rubyonrails.org, 2018g .)

2.3 Do not repeat yourself (DRY)

Do not repeat yourself principle states that “every piece of code must have single representation within the system”(Guides.rubyonrails.org 2018h). DRY concept stand opposite to write everything twice idea. DRY theorem in ruby on rails prohibits the duplication of similar piece of code in several places in application files. Dry principle motivates to reuse the same piece of code whenever possible through out the application development. The objective of Dry principle are as follow.

- The same piece of code with similar function and method are needed sometimes in various place of application. The duplication of same piece of code in almost every places might lead to massive application code structure. This big chunk of code might be unclear and time consuming to

debug and fix error if needed. Therefore, the piece of code written in one place and reused again gives the clean structure and view for the web developer to check and fix the error.

- The multiple use of a same code in almost every places means same memory has to be allocated in every places. This approach might not be wise to waste memory storage for the same piece of code. In cases, memory management plays vital role in the web development and deployment. Thus, Dry principle helps to save memory storage by motivating web developer to reuse the same piece of code whenever possible.

In conclusion the do not repeat yourself principle inject the idea of single representation of one piece of code in application system. In addition, it helps to make the application code clean and less memory consuming by suggesting the reuse of code and demotivating the code duplication.

2.4 Open source software

Ruby on rails is open source software. Open source software runs in the market with the idea of everyone should access the related knowledge or software free of cost. Hence Ruby on rails adopt the trend of open source software allowing anyone to use it for free. The Ruby on Rails software can be download online without making any payment. Apart from this, The software can be modified, merged and sub-licensed. The modified software should again be entitled to modify and reuse for the upcoming third party. However, it is necessary to give credit to proprietor with some kind of acknowledgement. It is a tribute to those whose hard work and creativity has helped in the learning process and played part in developing ones work.(GitHub, 2018)

4 RAILS OPTIMIZATION AND DEPENDENCY

The act or procedure of making the best use of a resources for productive and effective implementation of products is defined as optimization. Software Company builds application based on specific web programming language and framework for the use of their clients. It is a challenging task to satisfy their customers by maintaining their critical business applications. In modern business world, it is very essential the applications or software have high performance level and options for scalability whenever the business grows. Growing business demand sometimes makes it hard to maintain the application. Low speed and scalable issue leads to crippling productivity. Company customer might not stay for long if the demand is not met. Rails has considered this issue closely. Rails analyze different factors for the optimization of rails web application. Performance, scalability, maintenance, security are some clearly recognized factor that assist in facilitating the optimization of rails products. Putting these factor in the foreground, this part of thesis will emphasize on how the rails helps to optimize its application.

4.1 Performance optimization

In Rails, a performance testing is the vital part during the development phase of an application. The performance test is the pre-homework done by testing team before the deployment of application in real world. Performance testing needs tools and specific sets of skill to carry out effective testing. Appropriate tools and techniques helps to build application that can satisfy the speed and pleasant browsing experience demand of the customer. Unnecessary cost related to hardware addition such as memory usage is possible to avoid by carrying proper performance test before deployment phase. The performance test helps to figure out the region in application associated with speed latency and memory problem. The textual output and graphical diagrams are heavily used to identify and solve the performance issue with in application.

Rails has various ways of doing performance tests. The two focal ways of accomplishing performance test is benchmarking and profiling . Prior to rails version 3.0.0, the performance test library or gem were used to be standard library for testing purpose. After version 3.0.0, one has to import 'rails-perftest' gem into the developing application to perform benchmarking and profiling test.. The rails-perftest gem can be downloaded from various sites that host gem and tools needed to develop rails application.

The 'https://rubygems.org' is one of the widely used online site to download and import required gems. Downloaded gems are included in Gem file folder of the newly created application to implement performance testing using rails-perftest. The below mentioned line includes rails-perftest gem in the application's Gem file.

```
gem 'rails-perftest' ;

gem 'prof-test';
```

A Rails performance test is carried out in two ways, benchmarking and profiling (Guides.rubyonrails.org 2018i). Both test are used to test whole application system or for specific unit of code. Specific unit of code refers to individual controller, model or database and view template code of newly generated application project. These tests figure out the request/response time of that particular unit of code or whole application building block of code and the memory storage it occupies. Rails facilitates a generator known as 'performance_test' to initiate new test (Guides.rubyonrails.org 2018j). The command to create new test is:

```
rails generate performance_test testname;
```

4.1.1Benchmarking

In computer programming, benchmarking is the execution of series of computer program to assess the relative performance of application. Rails Benchmarking test consists of various metrics. Metrics are measurement unit that needs

proper understanding and evaluation skills. System and subsystem of application performance can be properly analyzed through metrics evaluation. The test in benchmarking mode, is initiated using below mentioned command in command line interface.

```
Rake test:benchmark;
```

Bench marking test can be tuned to evaluate specific metric. The tuning test can be achieved by referring specific metrics in the test code. After the test is executed, the gathered metrics such as wall time, memory and CPU process time used by tested system or subsystem in application is shown in the form of text output in command line interface. In addition to this, test create number of CSV files under tmp/performance in rails project application folder. Rails has CSV class that provides interface to read and write data from strings or Input/output object. (Guides.rubyonrails.org 2018k)

In conclusion Benchmarking testing mode aims at performance optimization of rails application by observing various metrics involved in benchmarking test.

4.1.2 Profiling

Profiling is another performance testing mode in rails applied to optimize rails application performance. In computer programming, profiling is a dynamic way of program analyses aimed to measure the memory space, time period of request response process and duration of function call of the software codes (En.wikipedia.org 2018e). In rails, as ruby is the programming language, the native profiler is Jruby and Rubinius. MRI is an example of non-native profiler (Guides.rubyonrails.org 2018i). Profiling in rails use different kind of tool to evaluate its data. Rack mini-profiler, Devtrace, railspanel are some of rails profiling tool used for in depth analysis of rails application. The result of profiling is shown in command line. Graphs and trees are alternatives to command line to print test output in profiling.

4.1.3 Metrics used in benchmarking and profiling

Benchmarking and profiling use various metrics during the performance test. The list of metrics that are observed during the performance test are briefly outlined below.

- Wall time: The Wall time is the real time taken by website to appear in the user's browsers. It is affected by all the processes running in the system (Guides.rubyonrails.org 2018L).
- Process time:The Process time is the time taken for the specific processing. Irrespective of machine workload, process time is usually constant. Unlike wall time, it is not affected any other process running in the system (Guides.rubyonrails.org 2018L).
- CPU time: This is the image of process time. This is time taken by the CPU processor of the tested system (Guides.rubyonrails.org 2018L).
- Memory: It is the amount of storage space occupied during the performance test case (Guides.rubyonrails.org 2018L).

Metrics and its availability always depend on the interpreter used by rails. Jruby, MRI, Rubinius and REE are some of the interpreter used by rails application. Some interpreter supports these metrics and some does not. So, testing team should have knowledge about the metric support in the interpreter used. Above all, rails provide vital testing mode and support to develop better performing application.

4.2 Scalability

The ability of any application to change its size using various technique is called scalability. Scalability is one of the most essential feature a modern web application should have to survive in the growing market. Scalability issue depend on the architecture of application. Another factor affecting the scalability

of entire system is the structure of server implemented for application. Considering request load and architecture of system setup, scalability issue and solution is discussed below.

The deployment of new project or application serves its users. The situation might change over the period of time regarding the number of application users. Implementation of single tier server architecture in the initial phase responds specific number of request per minute in certain time period.The increase in size of users means the workload of server to respond request per minutes also rise. This scenario might lead to slow response to users request. Figure 3 illustrates the single tier architecture server initially deployed for start up application. (Rubygarage.org. 2018.)

Figure 3 Single tier server architecture (Source : https://rubygarage.org/blog/ruby-on-rails-is-scalable)

Sorting out this issue is carried out by implementing vertical or horizontal scalability in the system setup.

4.2.1 Horizontal scalability

The load distribution method applied by introducing two or more layers architecture of server instead of single architecture is called horizontal

scalability. Figure 4 shows horizontal scalability. The Nginx server deployed here has slightly different work responsibility.

The Nginx is responsible to divide the workload across three different layers. The set up here shows three instances of rails application and database. Each of these instances perform their own task that is request from users. The basic idea that Nginx adopt to assigned the task is through load distribution. The initial request is assigned to first machine, second request is forwarded to second. In the same way third request is handled by third machine. Likewise, the fourth request is sent to first machine even though it is serving first request. The same is applied for remaining two machines. Hence this multi-layer architecture works on the principle of load balancing. At the same time, it increases responsive ability of rails application. (Rubygarage.org. 2018.)

Figure 4 Three tier architecture server. (Source : https://rubygarage.org/blog/ruby-on-rails-is-scalable)

4.2.2 Vertical scalability

Vertical scalability is a way of handling the increased request per minute or hour by giving additional memory storage or processing power to the server. In other

words, adding RAM memory and upgrading server processor in the server computer is the vertical scalability. Additional RAM and processor gives more computational ability to current server.Hence more user requests can be processed. Hence more user requests can be processed. This scalability method works fine at the initial stage of server scaling. When the request increases after first scaling, addition of RAM and processor for second time is technically difficult or impossible. Vertical scalability is ideal only when initial and final scaling of server is required. It is always good idea to look for horizontal scalability method whenever vertical scalability technically fails to sort scalability issue. (Rubygarage.org. 2018.)

In conclusion, rails application is scalable. One must do all the planning and analyzing homework before deploying one of the scaling technique. In addition, one needs to analyze the architecture of server, application and database for proper scalability of rails application. Apart from this, there are scalability issues, that are very tough to find manually. There are handful of software that are designed to find out issue regarding scalability. Splunk, new relic, flame graphs are the software that collect data and statistics of an application. Statistics in turn helps to find out what to change. For example, flame Graphs is used to learn about the CPU load when running the application. Within an application, CPU process are like tree branch, if one finds out in which part the processing is slow, one can work to find the solution.

4.3 Rails security system

Web applications are prone to different external web threat. User account hijacking, bypass of access control, modifying and twisting sensitive data, presenting tricky content are some reported security problem in the web applications. Rails applications are no exceptions to such threat. Web application environment consist of different layer: web server, back end storage, session layer, transport layer, web application layer itself and possibly many more. It is critical to understand this all layers and the edge point of that

particular layer which have high risk of getting exploited. A secured web application can be built by isolating all points of attack in a different layers and applying correct countermeasure.The various kind of web threat and associated countermeasures are discussed in the following topic.

4.3.1 Session layer security

In web networking, session is a conversation between two communicating devices (En.wikipedia.org 2018f). In general language, it is an interaction among two parties, client as browser and server. The relationship between two parties is maintained using session ID. New session along with session id is created when user visit site for the first time. Existing session is loaded. if the user has visited the site more than once. For the objective of speedy connection, rails save the session as a cookie in the client's browser. The cookie and session are sensitive data. Session hijacking or simply stealing user session id can tamper the site. It is vital to protect this layers weak points while developing rails application (Guides.rubyonrails.org 2018m).

SecureRandom.hex is used in rails to compose the session ID. It generates cryptographically secure random numbers used as session Id (Guides.rubyonrails.org 2018). To prevent session Hijacking, rails facilitates secure connection implementing secure socket layer. SSL or secure socket layer connection is achieved using following line in configuration file.

```
Config.force_ssl=true;
```

Rails application stores session as cookie in the client's browser for speedy connection. Session can be another attacking point and must apply some security measurement. Rails facilitates EncryptedCookieStore for it.

It enables the encrypted session storage before being saved in cookie. It is accomplished using the server-side secret key secrets.secret_key_base. Apart from session id stealing, session id fixation could be attacker's motive. Session id fixation enable attacker to use the known id in the browser for session fixation.

Session fixation can be countered using reset_session (Guides.rubyonrails.org 2018). Cookie session delete method should be used to delete session as attacker might try to keep the session alive to tamper the cookie content.

4.3.2 Cross site request forgery (CSRF)

A website or application works on the basis of session established between server and client's browser. The user is authorized to do a lot of stuff in that particular website or application like changing password, changing the content or money transfer. When the session is established, the user might visit some other link or might get email. Those email or link might be crafted to have hidden code directed at doing what attacker intended. When user click link or image, this act might send request to server through victim's session established in the browser. So a CSRF vulnerable site might lead to bigger issues sometimes. (YouTube. 2018.)

Rails has introduced required security token which is encrypted value that rails-based sites know but other sites have no understanding of token. Included security token ensure every request is verified by matching the token in the server. Security token is by default in newly generated rails application.

```
protect_from_forgery with:: exception;
```

Security tokens are compulsorily added in all the forms and Ajax request if above line of code is included in application controller. The security token can be modified and used in more advanced and effective way in rails application. (Guides.rubyonrails.org 2018o.)

4.3.3 Miscellaneous attacks

Rails have made a list of multiple common and day to day web related security threats and gives all sorts of direction to adopt skills to put off those threats. Some of well-known security attacks are listed below.

- File upload threat is common to those application which allows file uploading environment in them. Attacker intends to overwrite the existing file through different tricks. Processing media files asynchronously is a solution to file uploading related threats.(Guides.rubyonrails.org 2018p)
- Account hijacking threat means to steal the user account through various attacks. Rails suggest to make the forms safe against CSRF attacks and use of old password while making a new one.
- SQL Injection influences databases manipulation. The purpose of this type of attack is to bypass authorization. The another objective is to manipulate data and read sensitive data from databases. Ruby on rails uses built-in filters for special SQL character. Model.find(id) is popular countermeasures to prevent this kind of attack. (Guides.rubyonrails.org 2018q)

4.4 Maintenance of rails application

Optimization of rails application depends on how well it is maintained. A properly maintained application contribute to the present and future lifetime of any application. Rails supply support to maintain rails application mainly in four different division. The four groups of support are bug fixes, new features, security issues and severe security issues. It uses semantic version as a method to apply required maintenance. (Guides.rubyonrails.org 2018r)

- Patch z: In this semantic version, bug fixing package is included for the rails framework. It does not include any API changes or new features. This semantic version is applied when any rails framework application need error fixing.
- Minor y: Sometime, developing application need new features or might need kind of change in application interface to adjust the current need of the project. This semantic comes in handy and is applied in rails.

5 PROJECT WEBSITE OVERVIEW

The website was built as a part of thesis project . The idea behind building the website was to become familiar with the rails framework. The website is built to give customer online service about mobile brands. It was built for those customer who are fond of mobile phones, of different brands such as Apple, Android and Huawei. This website was built on rails version 5.0.6 and ruby version 2.4.2. The main feature of this website is to sort out mobile phones according to their brand. JavaScript has been used to add the sorting functionality according to brand.

Website pages and functionality:

There are mainly four pages in this website. Home, Aboutus, Products and the sign-in page. The Home page has the sliding functionality. Users can experience the sliding view of different mobiles phones in this page. When clicking on the picture in slide-show, it will direct the users to the product page. The product page is the main page where users can check variety of mobile phones of different brands. This page can show all mobiles phones of different brand at once. If users are interested in specific mobile brand, they can check specific mobile phones by clicking the button related to that specific brand. Hence the product page has ability to show all mobile brand at once or sort specific mobile brand according to the requirement of users. This page also shows the details of specific phones such as RAM, processor, made-model in case the user is interested. The About us page is the information page of the mobile selling company. This page contains the contact information of mobile selling company. Apart from this page the website has login page. New users can use sign-up page to register themselves to the website. If the user already has sign-up, the user can simply login by using login page. The default SQLite database has been used to keep record of the registered customer. Those customer who has not made an account in the website cannot login. One has to make an account to login in the page.

External tool used: External tool ‘Devise ’ has been used to built sign in and sign up page for the website. The devise is also used as authentication system for sign in and sign up page .This tool or gem was imported in the Gem File folder of the website. Following semantics has been used to import devise gem.

```
gem ‘devise’;
```

Server: Puma has been used as local server to host this website.

Command line tools: Rails and server were the two mainly used command line tools. Rails command lines tool was used to generate two controller and views needed for the website. Server command line tool was used to start the local server.

Text Editor: Sublime text and vim editor were used to write the source code during website development.

Logo: The logo for the website was made online from site 'https://www.freelogodesign.org/index.html'.

6 CONCLUSION

The Rails framework is based on simple ruby programming language. The Ruby language helps programmer to use all the feature such as class, mixin, inheritance, memory management, error handling to write program code. The Rails built-in SQLite database is used for storing data, hence developer do not need to worry about database and record keeping. SQLite database has been used for the record keeping in the thesis project website. Beside this, Rails is equipped with different command line tools to manage the website or application. The website or application management job during development phase becomes easy with the availability of these command line tools such as generate, destroy, server and rails. During the thesis website development, command line utility were very helpful in interacting with website. The deployment of the Rails website to the real world can be done using web hosting company such as Heroku or digital ocean. Apart from this, Rails has plenty of tools such as Rubocop, Rubycritic, Trace route, Warden, Devise that is helpful during development and production phase of rails website. The Devise tool was used in the website to create the form for the customer. The Rails framework is based on MVC principle, the Do not Repeat Yourself and convention over configuration. These principles were adopted to build website in the development phase. Apart from this, HTML was used to create view for the users, CSS was used to add style in the page. The sliding functionality in the website was built from JavaScript. In conclusion, Rails framework can be picked considering ruby programming language, rails principle, database, tools, deployment, security to develop a proper website or application.

REFERENCES

Awesome-ruby.com. 2018. Awesome Ruby. [online] Available at: http://awesome-ruby.com/#-authentication-and-oauth [Accessed 3 Apr. 2018].

Betterexplained.com. 2018. Intermediate Rails: Understanding Models, Views and Controllers. [online] Available at: https://betterexplained.com/articles/intermediate-rails-understanding-models-views-and-controllers/ [Accessed 4 Apr. 2018].

Digitalocean.com.2018. A Comparison of (Rack) Web Servers for Ruby Web Applications | DigitalOcean. [online] Available at: https://www.digitalocean.com/community/tutorials/a-comparison-of-rack-web-servers-for-ruby-web-applications [Accessed 3 Apr. 2018].

Emery, C. 2018. A Brief History of Web Development. [online] Available at: https://www.techopedia.com/2/31579/networks/a-brief-history-of-web-development [Accessed 2 Jan. 2018].

En.wikipedia.org. 2018b. Cache (computing).[online] Available at: https://en.wikipedia.org/wiki/Cache_(computing) [Accessed 3 Apr. 2018].

En.wikipedia.org.2018c. Model–view–controller.[online] Available at: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller#History [Accessed 4 Apr. 2018].

En.wikipedia.org. 2018d. Convention over configuration. [online] Available at: https://en.wikipedia.org/wiki/Convention_over_configuration [Accessed 4 Apr. 2018].

En.wikipedia.org. 2018e. Profiling (computer programming). [online] Available at: https://en.wikipedia.org/wiki/Profiling_(computer_programming) [Accessed 17 Apr. 2018].

En.wikipedia.org. 2018f. Session (computer science). [online] Available at: https://en.wikipedia.org/wiki/Session_(computer_science) [Accessed 5 Apr. 2018].

GitHub. 2018. rails/rails.[online] Available at: https://github.com/rails/rails/blob/master/activerecord/MIT-LICENSE [Accessed 4 Apr. 2018].

Goodrich, G. 2018. Understanding the Model-View-Controller (MVC) Architecture in Rails — SitePoint. [online] SitePoint. Available at: https://www.sitepoint.com/model-view-controller-mvc-architecture-rails [Accessed 4 Apr. 2018].

Guides.rubyonrails.org. 2018a. Ruby on Rails Guides: [online] Available at: http://guides.rubyonrails.org/v2.3/getting_started.html#configuring-a-database [Accessed 3 Apr. 2018].

Guides.rubyonrails.org. 2018b. Ruby on Rails Guides: Getting Started with Rails.[online] Available at: http://guides.rubyonrails.org/v2.3/getting_started.html#the-components-of-rails [Accessed 3 Apr. 2018].

Guides.rubyonrails.org. 2018c. Ruby on Rails Guides: A Guide to The Rails Command Line. [online] Available at: http://guides.rubyonrails.org/v2.3/command_line.html#command-line-basics [Accessed 3 Apr. 2018].

Guides.rubyonrails.org. 2018d. Caching with Rails: An Overview — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/caching_with_rails.html#basic-caching [Accessed 3 Apr. 2018].

Guides.rubyonrails.org. 2018e. Debugging Rails Applications — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/debugging_rails_applications.html#view-helpers-for-debugging [Accessed 3 Apr. 2018].

Guides.rubyonrails.org.2018f. Getting Started with Rails.[online] Available at: http://guides.rubyonrails.org/getting_started.html#installing-rails [Accessed 3 Apr. 2018].

Guides.rubyonrails.org. 2018g. Getting Started with Rails — Ruby on Rails Guides. [online] Available at:

http://guides.rubyonrails.org/getting_started.html#creating-the-article-model [Accessed 17 Apr. 2018].

Guides.rubyonrails.org. 2018h. Getting Started with Rails — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/getting_started.html#what-is-rails-questionmark [Accessed 4 Apr. 2018].

Guides.rubyonrails.org.2018i. Ruby on Rails Guides: Performance Testing Rails Applications. [online] Available at: http://guides.rubyonrails.org/v3.2.13/performance_testing.html#modes [Accessed 4 Apr. 2018].

Guides.rubyonrails.org. 2018j. Ruby on Rails Guides: Performance Testing Rails Applications. [online] Available at: http://guides.rubyonrails.org/v3.2.13/performance_testing.html#generating-performance-tests [Accessed 4 Apr. 2018].

Guides.rubyonrails.org. 2018k. Ruby on Rails Guides: Performance Testing Rails Applications. [online] Available at: http://guides.rubyonrails.org/v3.2.13/performance_testing.html#understanding-the-output [Accessed 17 Apr. 2018].

Guides.rubyonrails.org. 2018L. Ruby on Rails Guides: Performance Testing Rails Applications.[online] Available at: http://guides.rubyonrails.org/v3.2.13/performance_testing.html#metrics [Accessed 4 Apr. 2018].

Guides.rubyonrails.org. 2018m. Securing Rails Applications — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/security.html#what-are-sessions-questionmark [Accessed 17 Apr. 2018].

Guides.rubyonrails.org. 2018n. Securing Rails Applications — Ruby on Rails Guides.[online] Available at: http://guides.rubyonrails.org/security.html#session-id [Accessed 17 Apr. 2018].

Guides.rubyonrails.org. 2018o. Securing Rails Applications — Ruby on Rails Guides. online] Available at: http://guides.rubyonrails.org/security.html#csrf-countermeasures [Accessed 17 Apr. 2018].

Guides.rubyonrails.org. 2018p. Ruby on Rails Security Guide — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/security.html#file-uploads [Accessed 5 Apr. 2018].

Guides.rubyonrails.org. 2018q. Ruby on Rails Security Guide — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/security.html#sql-injection [Accessed 5 Apr. 2018].

Guides.rubyonrails.org. 2018r. Maintenance Policy for Ruby on Rails — Ruby on Rails Guides. [online] Available at: http://guides.rubyonrails.org/maintenance_policy.html [Accessed 5 Apr. 2018].

Hotframeworks.com. 2018. Web framework rankings | HotFrameworks. [online] Available at: https://hotframeworks.com/ [Accessed 17 Apr. 2018].

Infinum.co. 2018. Top 8 tools for Ruby on Rails code optimization and cleanup. [online] Available at: https://infinum.co/the-capsized-eight/top-8-tools-for-ruby-on-rails-code-optimization-and-cleanup [Accessed 3 Apr. 2018].

Rubygarage.org. 2018. What if I Tell You That Ruby on Rails Is Scalable. [online] Available at: https://rubygarage.org/blog/ruby-on-rails-is-scalable [Accessed 4 Apr. 2018].

Ruby-lang.org.2018. About Ruby. [online] Available at: https://www.ruby-lang.org/en/about/ [Accessed 2 Apr. 2018].

Stewart, B. 2017. An Interview with the Creator of Ruby - O'Reilly Media. [online] Linuxdevcenter.com. Available at: http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html [Accessed 17 Apr. 2018].

Sqlite.org. 2018. About SQLite. [online] Available at: https://www.sqlite.org/about.html [Accessed 3 Apr. 2018].

YouTube. 2018. CSRF Explained. [online] Available at: https://www.youtube.com/watch?v=vrjgD0azkCw&t=312s [Accessed 5 Apr. 2018].

www.tutorialspoint.com. 2018. Ruby on Rails MVC Framework. [online] Available at: https://www.tutorialspoint.com/ruby-on-rails/rails-framework.htm [Accessed 4 Apr. 2018].