

**CONTENT MANAGEMENT SYSTEM (CMS)  
TO MANAGE DYNAMIC CONTENT FOR WEB APPLICATION**

**BY  
FARUK AHMED ANY  
ID : 141-15-3084**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor  
of Science in Computer Science and Engineering

Supervised By

**Dr. Fernaz Narin Nur**  
Assistant Professor  
Department of CSE  
Daffodil International University.

Co-Supervised By

**Md. Sazzadur Ahamed**  
Lecturer  
Department of CSE  
Daffodil International University.



**DAFFODIL INTERNATIONAL UNIVERSITY  
DHAKA, BANGLADESH  
SEPTEMBER, 2018**

## **APPROVAL**

This Project titled “**Content Management System(CMS)**”, submitted by Faruk Ahmed Any , ID: 141-15-3084 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering (BSc) and approved as to its style and contents. The presentation has been held on 10 June 2016

## **BOARD OF EXAMINERS**

---

**Dr. Syed Akhter Hossain**  
**Professor and Head**

Department of CSE  
Faculty of Science & Information Technology  
Daffodil International University

**Chairman**

---

**Dr. Sheak Rashed Haider Noori**  
**Associate Professor and Associate Head**

Department of CSE  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

---

**Md. Zahid Hasan**  
**Assistant Professor**

Department of CSE  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

---

**Dr. Mohammad Shorif Uddin**  
**Professor**

Department of Computer Science and Engineering  
Jahangirnagar University

**External Examiner**

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. Fernaz Narin Nur, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

### Supervised by:

---

**Dr. Fernaz Narin Nur**  
Assistant Professor,  
Department of CSE  
Daffodil International University

### Co-Supervised by:

---

**Md. Sazzadur Ahmed**  
Lecturer  
Department of CSE  
Daffodil International University

### Submitted by:

---

**(Faruk Ahmed Any)**  
ID: 141-15-3084  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Dr. Fernaz Narin Nur, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Object Oriented Programming and Networking to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Prof. Dr. Syed Akhter Hossain** Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

## **ABSTRACT**

Content Management System(CMS) is a tool for authoring on web and Managing content on behalf of a administration panel. Content Management Systems (CMS) are systems that assist in organizations to collect, support, organize and publish information on the Internet, intranet and extranet. CMS are gaining popularity in both large and small businesses due to the increasing need to handle the expanding amount of information on the web today. According to Hallikainen et al (2002) the rising interest in CMS can be traced to timing, cost and quality considerations and numerous problems related to this are e.g. delays in publishing .It will enhance the power of super admin. Both users and admin can publish pages and posts as well as . Voyager is a large set of collection of settings and web management tool that can enhance your web exprience .This Project is focused to develop a web application to make easier for all web administrator to manage all site users and their information in one place. The central place to manage your content is the Site Tree. Set up your menu structure, edit your pages and manage widgets. All in one interface, with simple drag and drop features.

# TABLE OF CONTENTS

<b>CONTENT</b>	<b>PAGES</b>
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v

## CHAPTER

<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1-3</b>
1.1 Introduction	1
1.2 Examples	1
1.3 Objectives	2
1.4 Expected outcome	2
1.5 Report Layout	3
<b>CHAPTER 2: BACKGROUND.....</b>	<b>4-6</b>
2.1 Introduction	4
2.2 Related Works	4
2.3 Comparative Studies	5
2.4 System Requirements	6
<b>CHAPTER 3: RERUIREMENT SPECIFICATION.....</b>	<b>7-9</b>
3.1 Introduction	7
3.2 Business Process Modeling	7
3.3 Requirement Collection and Analysis	8
3.4 Use Case Modeling & description	8
3.6 Design Requirement	9
<b>CHAPTER 4: DESIGN SPECIFICATION.....</b>	<b>10-12</b>
4.1 Font-end Design	10-11
4.2 Backend Design	12
4.3 Interaction Design & UX	12
4.4 Implementation Requirements	12

<b>CHAPTER 5: IMPLEMETATION AND TESTING.....</b>	<b>13-22</b>
5.1 Implementation of Database	13
5.2 Implementation of Font-end	14-20
5.3 Implementation of Interactions	20
5.4 Testing Implementation	21
5.5 Test Result and reports	22
<b>CHAPTER 6: CONCLUSION AND FUTURE SCOPE.....</b>	<b>23-24</b>
6.1 Discussion and conclusion	23
6.2 Scope for further developments	23
6.3 References	24

## **APPENDICES B: RELATED DAIGRAMS**

<b>List of Figures</b>	<b>PAGES</b>
Figure 3.1: Business Process Model	7
Figure 3.2: Use case Model	8
Figure 4.1: Laravel default directory structure	11
Figure 5.1: Sqlite Database Storage	13
Figure 5.2: Development Commannd	14
Figure 5.3: Login Panel	15
Figure 5.4: Dashboard Area	15
Figure 5.5: Feature Menu Section	16
Figure 5.6: Dynamic Database Creation	17
Figure 5.7: Drag and Drop Menu Builder	17
Figure 5.8: Api Functionality	18
Figure 5.9: Media Storage System	19
Figure 5.10: User Management System	20

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

A dashboard, in web administration, is typically the index page of the control panel for a website's content management system. A dashboard typically indicates item which require urgent actions at the top of the page, moving into less important statistics at the bottom.

A well-built admin panel can be turned into a dashboard to either display business metrics or analytics without the viewer writing a single bit of code, or to export weekly data which can be used to create reports. Add to the list that any QA member of the team will love an admin panel for the ease it gives them in verifying data, quickly entering or deleting data, and determining if an issue is in the frontend or the backend. Functioning without an admin panel can lead to a lack of insight into your data, and add greater difficulty for the testing team.

### 1.2 Examples of some CMS

There are some of most and commonly used CMS in web content management field. Depending on web programming languages admins can select multiple kinds of CMS. As an example of these are listed below.

1. The most common and widely used a CMS system is “WordPress” (<https://wordpress.org>) [4]
2. Drupal (<https://drupal.org>) [5]
3. Joomla (<https://joomla.org>) [7]
4. ExpressionEngine (<https://expressionengine.com>) [3]
5. TextPattern (<https://textpattern.com>) [5]
6. Radiant CMS ([www.radiantcms.org](http://www.radiantcms.org)) [3]

These are some examples of managing web content direct from their hosting site.



### **1.3 Objectives**

This system will provide a dynamic content management services and some kind of web management facilities. These are written below.

- A dynamic Users & Admin management.
- Post handling.
- Page creation.
- Site management.
- Multiple admin management.
- Login System
- Dynamic Database Management.
- BREAD (Browse, Read, Edit, Add, Delete) System.

### **1.4 Expected Outcome**

It's easy for the non-technically minded people. Not all users have the same comfort level with technology, but the basic CMS function of writing and publishing content, and slightly more advance ones of adding media are usually easy for to grasp.

In fact, anyone who can use word processing software can use a CMS for the basic function – so you don't even have to spend much time on training.

It allows multiple users, it improves site maintenance, and design changes are simple. It helps you to manage content from several platform. Instead of being reliant on an external vendor, with a CMS you are in control, with the ability to assign task and roles and to check progress at any time. That puts you in the driven seat when it comes to this important business tool.

## 1.5 Report layout

In Chapter 1 : Introduction

- Introduction
- Examples
- Objectives
- Expected outcome
- Report Layout

In Chapter 2 : Background

- Introduction
- Related Works
- Comparative Studies
- System Requirements

In Chapter 3 : Requirement Specificaation

- Introduction
- Business Process Model
- Requirement Collection and Analysis
- Use Case Modeling
- Design Requirement

In Chapter 4 : Design Specification

- Font-end Design
- Backend Design
- Interaction Design & UX
- Implementation Requirements

In Chapter 5 : Implementation and Testing

- Implementation of Database
- Implementation of Font-end
- Implementation of Interaction
- Testing Implementation
- Test result and reports

In Chapter 6 : Conclusion and Future Scope

- Discussion and conclusion
- Scope for further developments
- References
- Appendices

## CHAPTER 2

### BACKGROUND

#### 2.1 Introduction

Content Management Systems (CMS) are systems that assist in organizations to collect, support, organize and publish information on the Internet, intranet and extranet. CMS are gaining popularity in both large and small businesses due to the increasing need to handle the expanding amount of information on the web today. According to Hallikainen et al (2002) the rising interest in CMS can be traced to timing, cost and quality considerations and numerous problems related to this are e.g. delays in publishing, erroneous or out-of-date content, pressures on staff and infrastructure associated with web environment. Hallikainen et al (2002) [6] also emphasize the potential of CMS to work as a strategic tool that effects the organization on a company wide or even global level. The existing research on CMS is poor and Bergman and Ryman (2004) [5] mean that both the strategically and technological aspects of CMS are unexploited.

#### 2.2 Related Works

There are some kind of works are related to my work and here the breif link included to check-out the examples of related work and description. These sites are providing the management tools and services for the web users for a service provider. Thats why these systems are too popular for as the web development service. These are not framework but a controll panel.

1. The most common and widely used a CMS system is “Wordpress” (<https://wordpress.com>) [4]
2. Drupal (<https://drupal.org>) [5]
3. Joomla (<https://joomla.org>) [7]
4. ExpressionEngine (<https://expressionengine.com>) [3]
5. TextPattern (<https://textpattern.com>) [4]
6. Radiant CMS ([www.radiantcms.org](http://www.radiantcms.org)) [5]

These are some examples of managing web content direct from their hosting site.

## 2.3 Comparative Studies

In our comparative studies we found that many of the CMS benefits will be described in future sections aligned with the useful effects generated in the organization by the CMS implementation. Since the study is made on global organization and the system is used across the organization we believe that these useful effects are common enough to appear in other large organizations as well.

- Voyager is pretty much different from WordPress and Drupal. It has some cool features and programs for the users and admins. Both will feel comfortable to use and customize the sites with this application.
- WordPress has much popularity and many professional features but Voyager has some complex features that are much better than WordPress.
- Drupal has some important features which can be replaced by WordPress on the other hand Voyager has a complete package management tool to reduce the uses of WordPress.
- Custom theme and media storage tool will create a powerful usability of this application
- Many other apps have much more professional diagnostic tools but it can find errors and complex functions between users and admin functionality.

## 2.4 System Requirements

This application can be run on several kinds of server

- Web server.
- Microsoft Azure.
- Vultr.
- Google Cloud Platform etc. app hosting platform.

If it is running on windows for development process you have to install

- Git Bash or
- CMD (Windows).
- MySQL or SQLite Database
- Php 5.6 or 7

# CHAPTER 3

## REQUIREMENT SPECIFICATION

### 3.1 Introduction

Requirement analysis is concerned with discovered and deciding what the new system is required to do? In this chapter we will discussed about the requirement for the proposed system.

### 3.2 Business Process Modeling

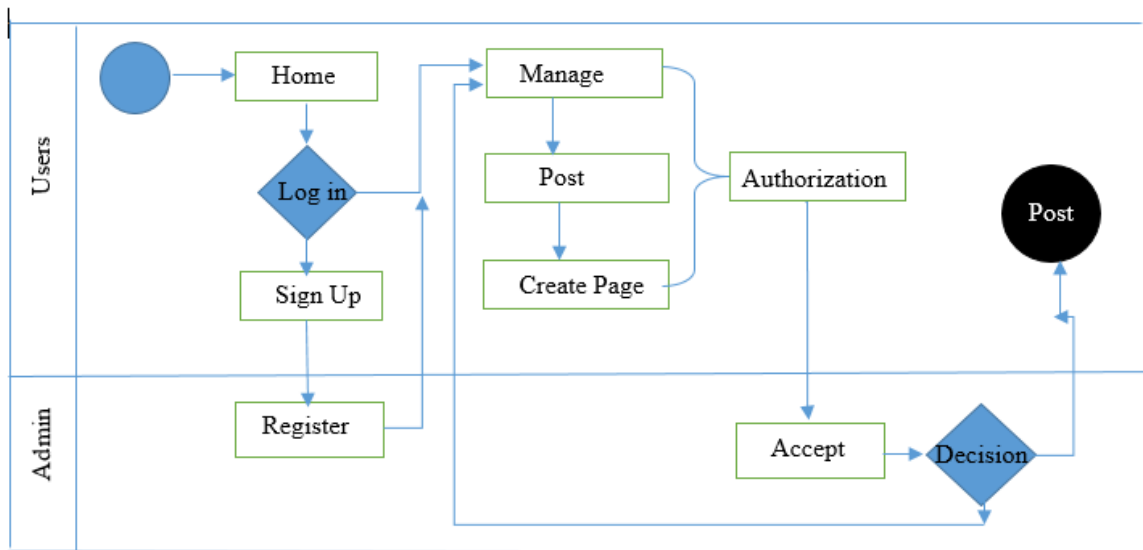


Figure 3.1 : Business Process Model

Business process demonstrating is mapping out standard business procedures and discovering approaches to enhance them. It is a piece of the act of business process administration [5]. The Business process modeling for our system is given in figure 3.1

### 3.3 Requirement Collection and Analysis

At the beginning of this project, I have identified the stakeholder for my project. Web admins are the main stakeholder of this system. Many of the CMS systems are not independent to use and developing the system, but using this system admin will really satisfy their tools. Some web administrators are feeling hesitant to choose their correct web platform to maintain the site for their office or company. This tool can help them to discover the new platform.

### 3.4 Use Case Modeling and Description

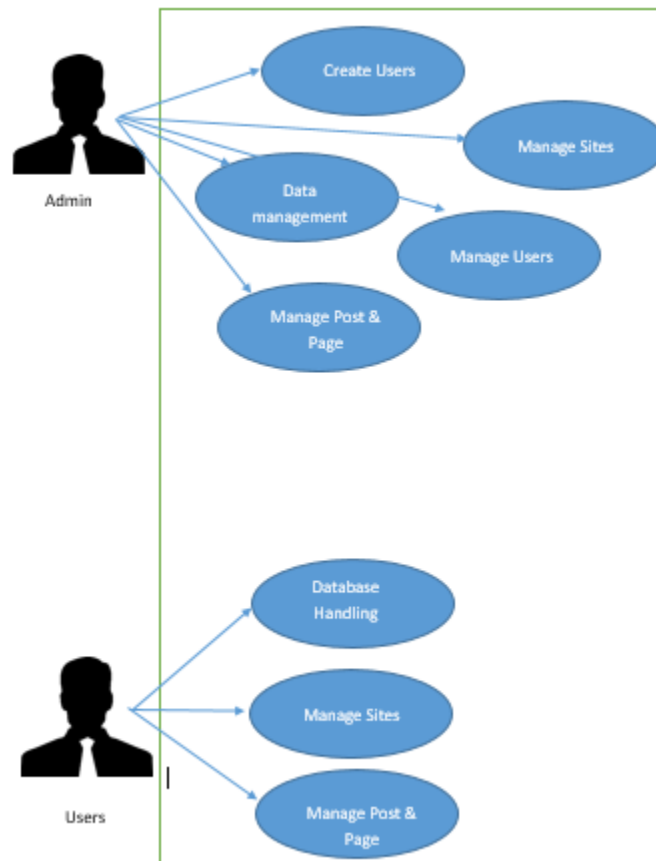


Figure 3.2 : Use Case Model

Figure 3.2 Shows the use case diagram of the system provided in the project. There are two types of actor in the application one is Admin and another is User. Both Users and Admin can post online and publish content on web.

### **3.5 Design Requirements**

- The system will be used by the 2 types of user. One is Super Admin and other is Admin or User.
- Super admin will create a admin account for a user.
- Admin and user can post and mantain the site both at a time in one platform.
- System admin will setup basic setting and tools management.
- System admin will develop the site user exprience and most of the cases.
- Admin will authorize the user post, and other management.



## CHAPTER 4

### DESIGN SPECIFICATION

#### 4.1 Font-end design

Voyager is built with most popular php framework Laravel 5.5 . Laravel is free, open-source Php web framework created by Taylor Otwell and intended for the development of web applications following the MVC (Model-View-Controller) architecture pattern based on Symfony.

Laravel 5 was released in February 2015 [5] as a result of internal changes that ended up in renumbering the then-future Laravel 4.3 release. New features in the Laravel 5 release include support for scheduling periodically executed tasks through a package called Scheduler, an abstraction layer called Fly system that allows remote storage to be used in the same way as local file systems, improved handling of package assets through Elixir, and simplified externally handled authentication through the optional Socialite package. Laravel 5 also introduced a new internal directory tree structure for developed applications.

## Laravel Directory Structure

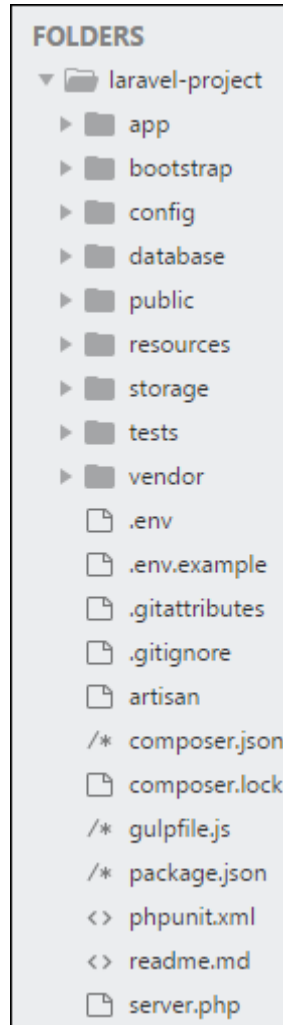


Figure 4.1: Laravel default directory structure.

Figure 4.1 shows the default directory folder structure of Laravel Framework. The “App” folder contain the front-end design content. The “database” folder contain the full database for the application.

## 4.2 Backend Design

Voyager application backend designed with “Php” which can run by blade template engine by default provided by laravel framework. Blade is a template design engine compile php, with html. I have used SQLite3 database to create data tables for users, admin, and site management system.

SQLite is popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets. SQLite has a small code footprint, makes efficient use of memory, disk space, and disk bandwidth, is highly reliable, and requires no maintenance from a Database Administrator.

How to browse data from SQLite

- DB Browser for SQLite installed in your machine.

## 4.3 Interaction Design and UX

Interaction Design is essential for making a system easy. He has done lots of things to make our system easy. We use buttons and icons for that people can easily interact with the system. We implement many options for selection. Users don't need to input text manually in areas; instead, they can choose options.

## 4.4 Implementation Requirements

To implement this application on anyone's system, he/she will have a complete server system on machine.

- Php : Need php 7 or higher on system.
- Framework : Laravel 5.5 is required to implement on system.
- Web Server : Any kind of web server required to run application.
- Terminal : A command terminal to run up the server.
- SQLite : SQLite3 database and its browser.

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation of Database

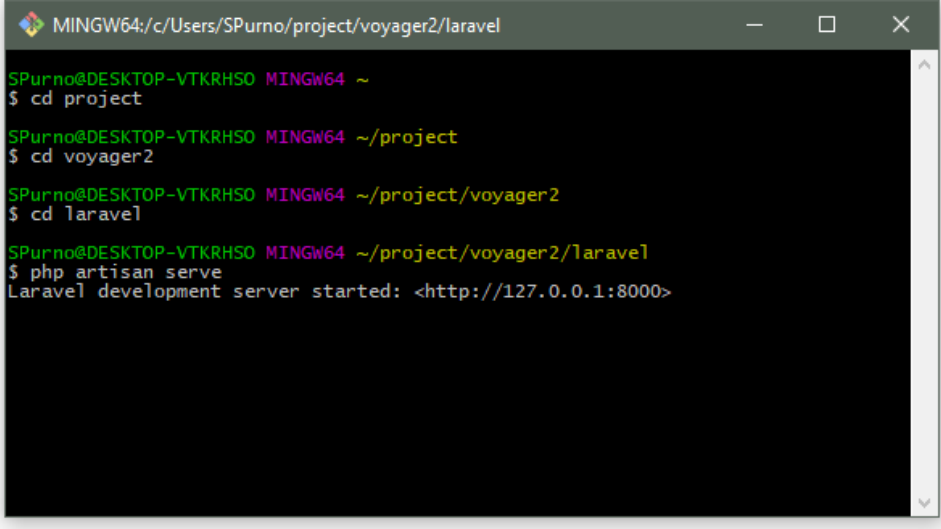
The implementation phase is where we install the DBMS on the required hardware, optimize the database to run best on that hardware and software platform and create the database table and load the data. Need to establish database security in this phase and give the various users by role that you've identified access applicable to their requirements. According to the role of users, DBMA will assign the permission for tables individually (if needed).

DBMS system follows sequential process to handle a BD request. Process demonstrated bellow.

- SQLite request goes to SQLite parser and stores it into code cache.
- The parser code goes to SQLite executer.
- Then the request checks the user permission for respective table.
- After the transaction being start.
- Until finish the transaction respected table's stores in cache.
- Once transaction done, the updated information of the table stores in Disk

## 5.2 Implementation of font-end

As it is management system of web application at first copy the full directory in the current system server location's root of the directory. Admin will create an account for a certain user and then he can login the management system with email and password otherwise can't login. The listed figure can be a user documentation of application. Users can find the development site at <https://127.0.0.1:8000>. Admin can access by <https://127.0.0.1:8000/admin>

A screenshot of a terminal window titled "MINGW64:/c/Users/SPurno/project/voyager2/laravel". The terminal shows the following commands and output:

```
SPurno@DESKTOP-VTKRHSO MINGW64 ~  
$ cd project  
SPurno@DESKTOP-VTKRHSO MINGW64 ~/project  
$ cd voyager2  
SPurno@DESKTOP-VTKRHSO MINGW64 ~/project/voyager2  
$ cd laravel  
SPurno@DESKTOP-VTKRHSO MINGW64 ~/project/voyager2/laravel  
$ php artisan serve  
Laravel development server started: <http://127.0.0.1:8000>
```

Figure 5.2: Development command and service

Figure 5.2 Shows the terminal where we need to command for running the application on either web server or the own machine.

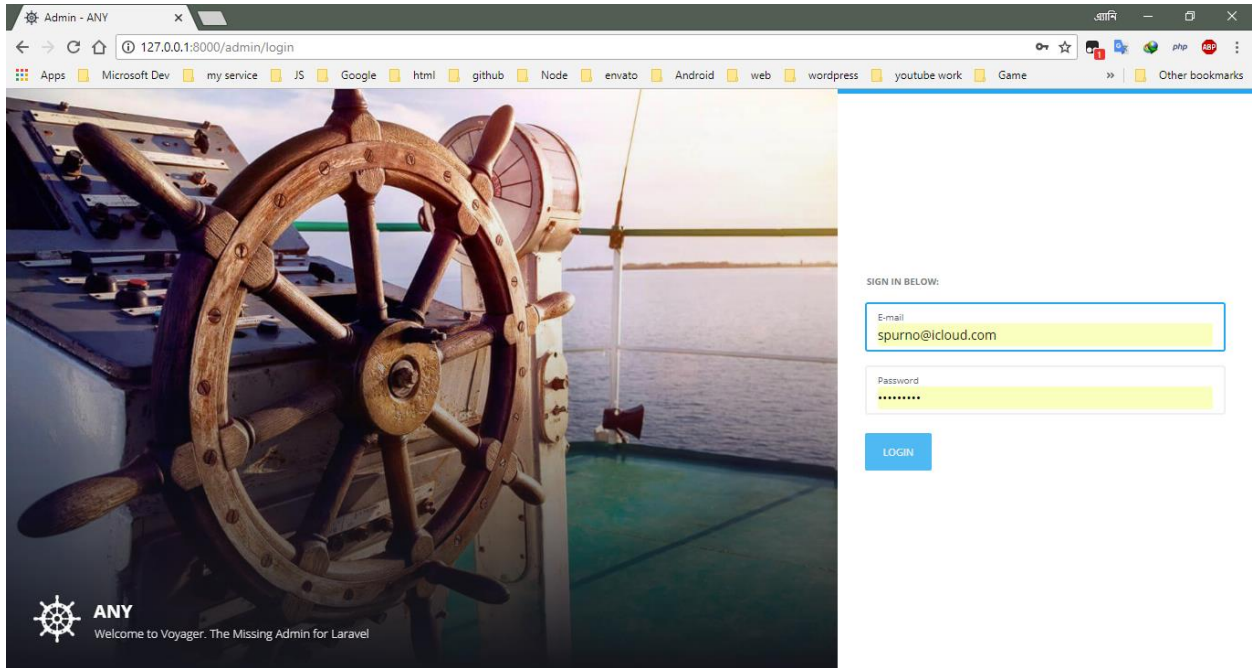


Figure 5.3 : Login panel

Figure 5.3 represents Login system to the administration panel. Need recognized email and password by admin to login.

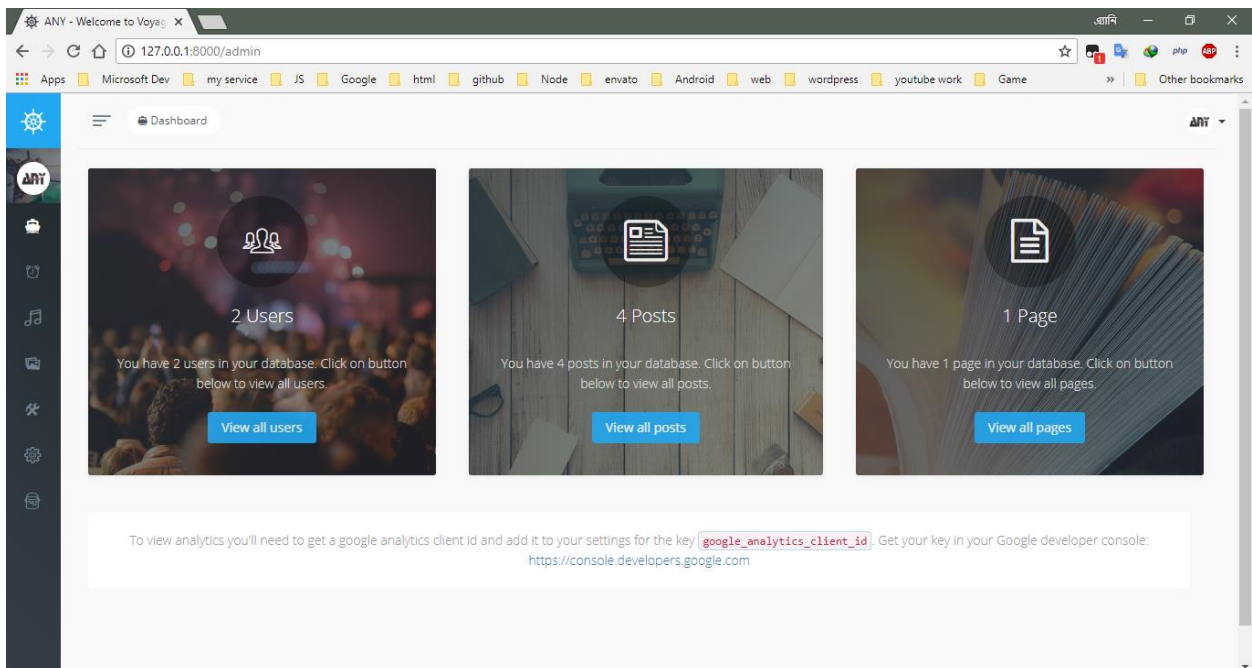


Figure 5.4 : Dashboard Area

Figure 5.4 represents the Dashboard panel of voyager. Including Users, Posts & Page counter.

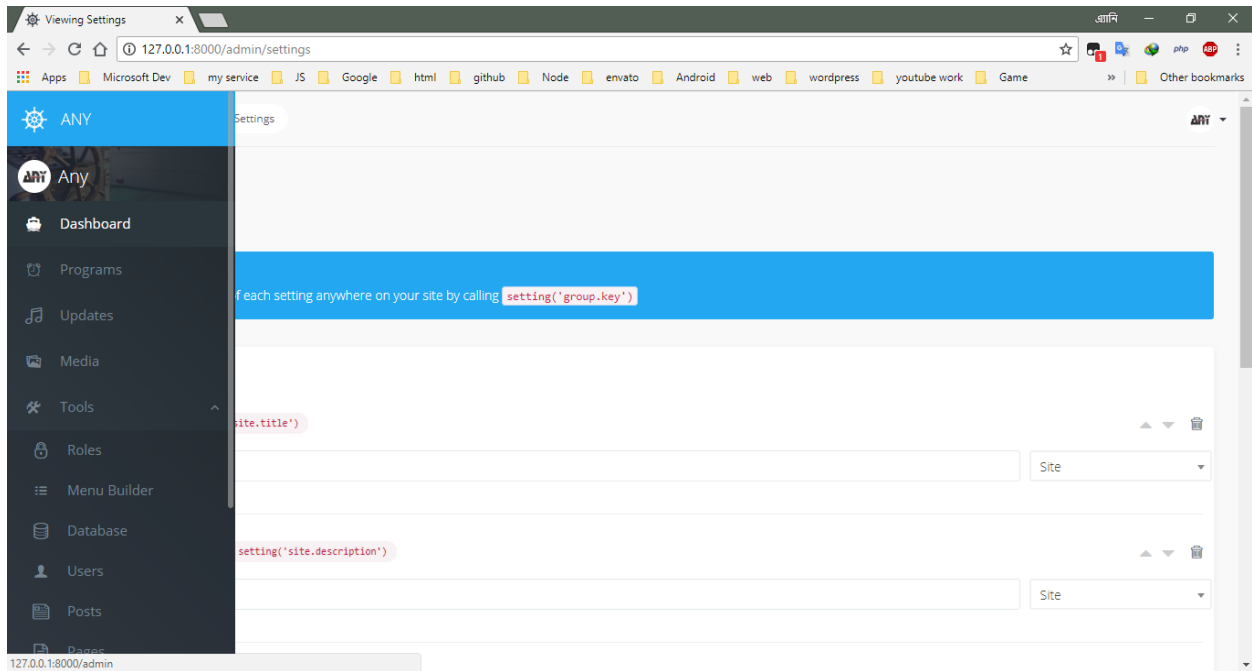


Figure 5.5 : Feature Menu Section

Figure 5.5 represents the feature menu section in left sidebar which is build with some most efficient couple of menus.

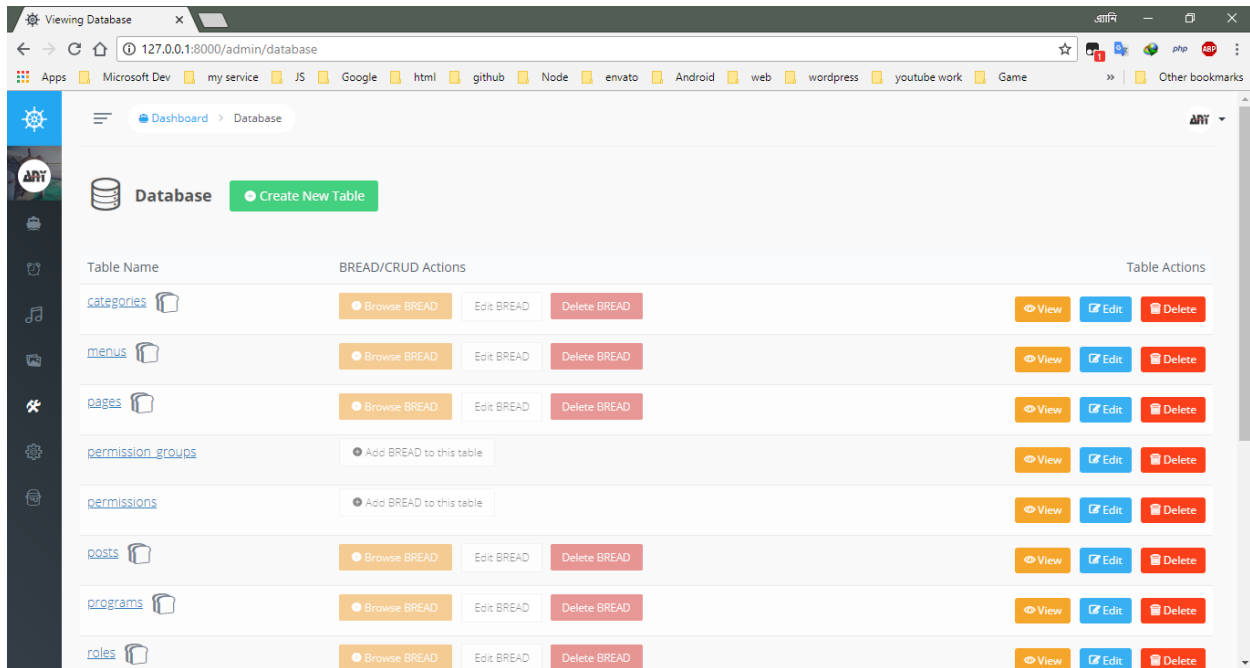


Figure 5.6: Dynamic Database Creation System.

Figure 5.6 represents the Dynamic Database Browse, Read, Edit, Delete & Update System. Users no need to go back into Database for check all this things.

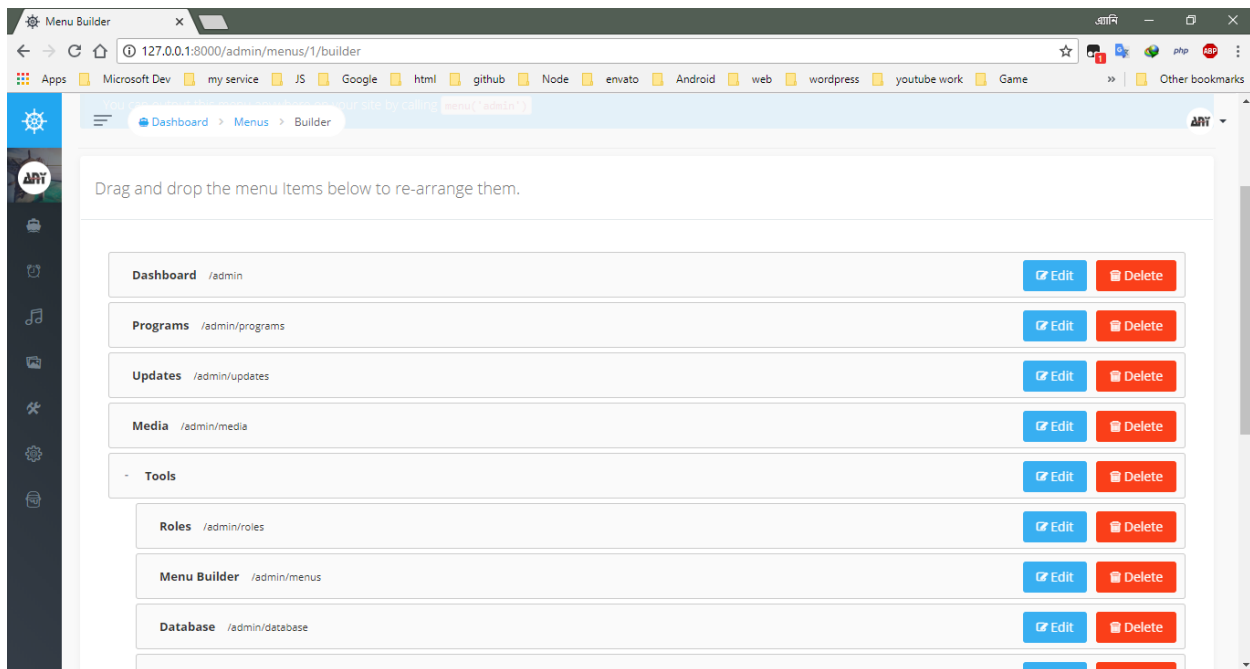


Figure 5.7: Drag & Drop Menu builder.

Figure 5.7 represents the Drag and drop menu builder for the user in the administration panel.



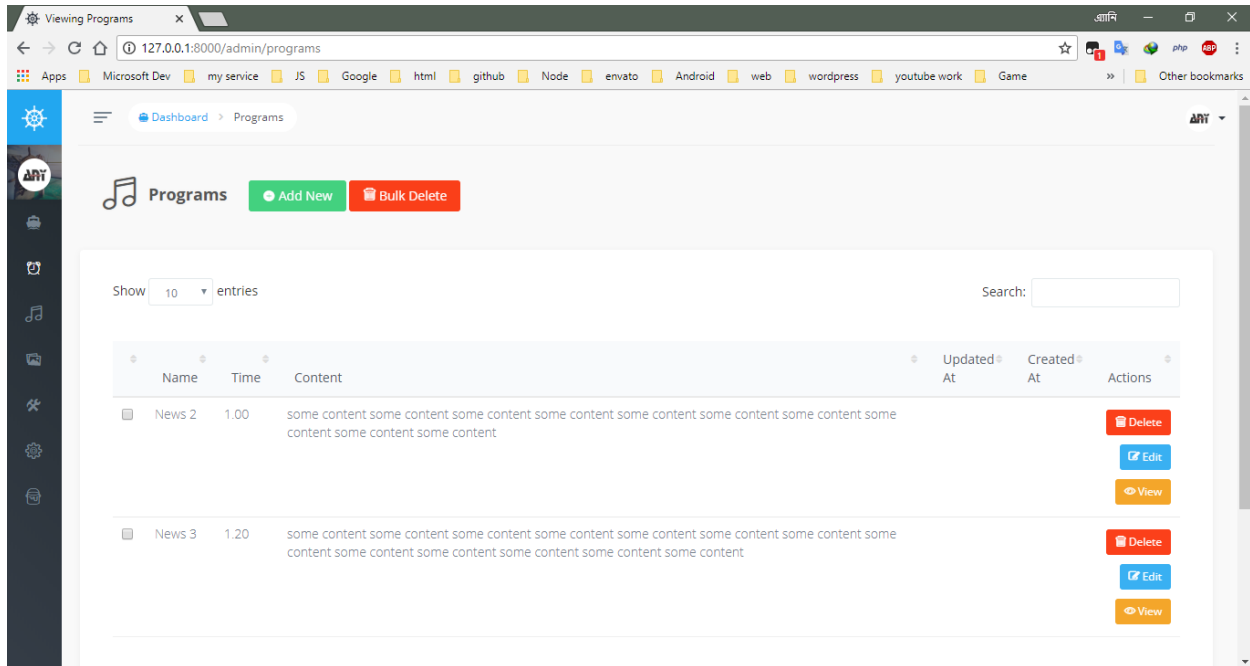


Figure 5.8 : API Functionality

Figure 5.8 Shows the function can export all post on website into a JSON output which can access through web in the other web API request. Can be access at <https://localhost/api/programs>.

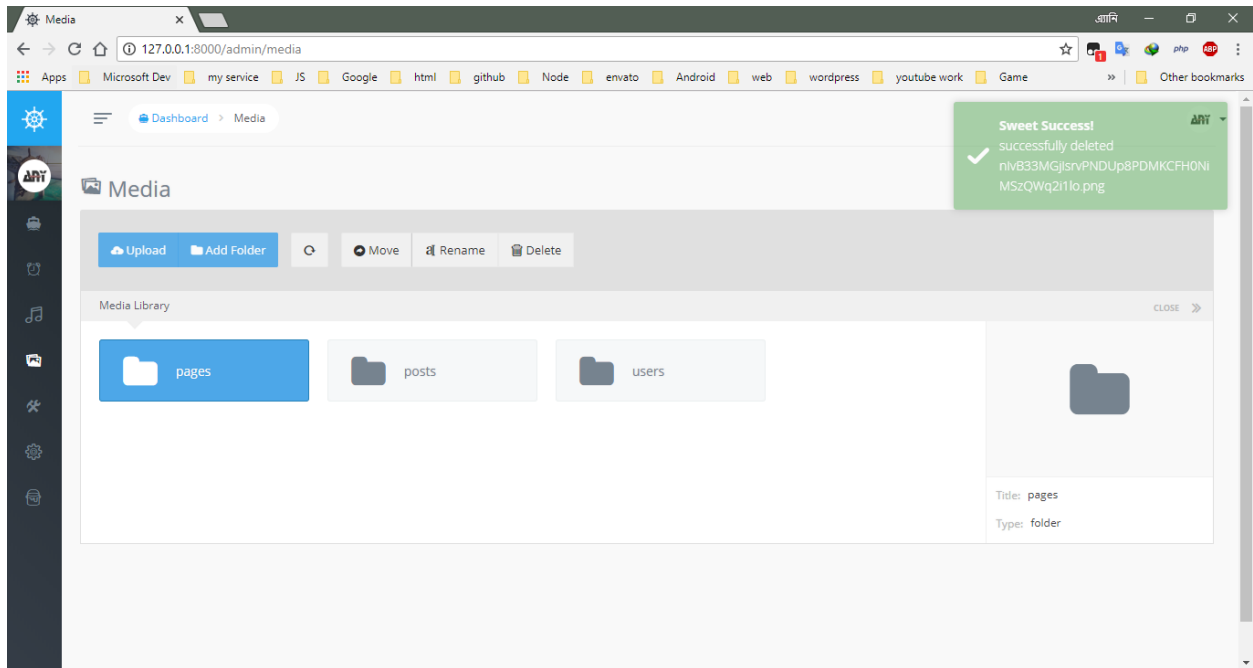


Figure 5.9: Media Storage System

Both admin and users can upload files like images, audio, video and other for managing content on web.

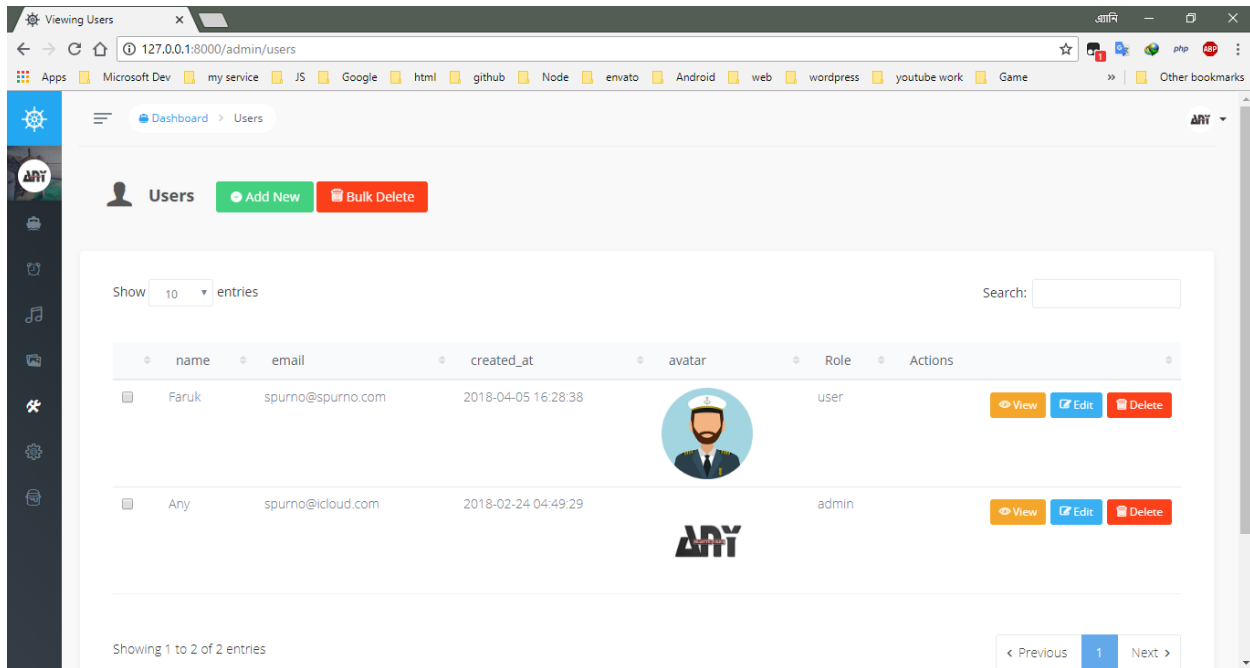


Figure 5.10 : User management system.

Figure 5.10 , An admin can add several users and can define more role to users. Then a user can login with the credential detail provided by admin.

## 5.2 Implementation of interactions

Into voyager app I have implemented interactive UI for easy and better user experience. In many cases I have used interactive icon rather than text or link. The admin will add different types of post, pages, and content upload on application add, edit, delete, & update user, database and browse database. Every database can be created with multiple rules and different kinds of values.

All the work is done very easily and the interaction between system and this application is very fast. All the functionalities is comfortable for an admin to handle the all category of management system. There are several types of common uses of menu for database create, browse, delete and update.

### 5.3 Testing Implementation

There are many strategies that can be used to test Application. An Application could wait until the system is fully constructed and then conduct testing on the overall system in hopes of finding error. This approach, although appealingly simple does not work. It will result in buggy software that disappoints the customer and end user. At the other extreme, a software engineer could conduct testing on a basis, whenever any part of the system is constructed. This approach although less appealing too many can be very effective.

A testing strategy that is chosen by most software teams falls between the two extremes. It takes an incremental view of testing, beginning with the testing of individual programs or units, moving to testing designed to facilitate the integration of the unit and culminating with testing that exercises the constructed system. Each of these classes of tests is described in the section that follows.

**Unit Testing:** Unit testing focuses verification exertion around the smallest unit of programming the product part or module. Utilizing the segment level outline portrayal as a guide. Various control ways are tried to reveal blunders those inside the limit of the module. The relative multifaceted nature of test and the blunders those tests reveal is restricted by the compelled scope built up for unit testing. The unit test concentrates on the inside handling rationale and information structure inside the limits of part.

The objective of the unit testing is to segregate each part of the program and demonstrate that the person's part is right. Unit testing gives a strict, composed set that the bit of code must fulfill a few advantages.

Unit testing help wipe out vulnerability in the unit themselves and can be utilized as a part of a base up testing style approach. By testing the piece of the program first and after that testing the total of its parts, joining testing must easier.

**Integration Testing:** Integration testing is a precise procedure for development the product design while in the meantime leading tests to reveal mistakes related with interfacing. The goal is to take unit test segment and assemble a program structure that has been directed by outline.

There is frequently propensity to endeavor no incremental joining that is to build the program utilizing a huge explosion approach. All segments are joined in cutting edge. The whole program is tried. Furthermore, mayhem more often than not comes about an arrangement of mistakes in

experienced. Redress is troublesome in light of the fact that seclusion of causes is muddled by the huge cost of the whole program.

Incremental approach is the enormous detour direct opposite approach. The program is developed and tried in little augmentation, where mistakes simpler to segregate and right interfaces will probably be tried totally, and a methodical test approach might be connected.

## **5.4 Test Result and Reports**

There were number of testing technique is real time software. But we tested our system is two testing system. Those two systems were unit testing and integration testing. As beginning of the coding section is unit testing. Where we make our different module, we tested by unit testing. After completing our full system, we tested by integration testing system. In every test section we found some problem and at that time we solve that problem. After then we complete our desire system.

## CHAPTER 6

### CONCLUSION AND FUTURE SCOPE

#### 6.1 Conclusion and discussion

In my developed system the goal was to make a proper maintenance a website administration panel with fully functional panel. It can be the most useful panel for a web admin or a user. It can reduce time for a user to maintain a site.

Most of the admin do not know how to control dynamically post and administrative features in one place. Voyager will create a place for those admin such a great way to handle all issues in several effective ways.

No need to depend on other framework or CMS system or no need to heir a developer to maintain the site. So Voyager is master piece of tool that can bring all tools in an arranged ways.

#### 6.2 Scope for Further Development

- In future or the next version of this app I will include many extra features and more tools to use it more simply.
- Next time I will provide two factor authentication to more secure this app's user account.
- This system already have capability to handle thousands of http request in one time. If the next users is increased then the system will provide a user friendly behave to user.
- Voyager will one the best app in application market for its better functionality that will be developed in future.

## REFERENCES

- [1]. Laravel : (<https://laravel.com>) and Laravel documentation page. [Last accessed on 06-05-2018 12.00pm]
- [2]. Laracast : (<https://laracast.com>) for laravel user tutorial. [Last accessed on 06-05-2018 12.00pm]
- [3]. Wikipedia: ([https://en.wikipedia.org/wiki/Content\\_management\\_system](https://en.wikipedia.org/wiki/Content_management_system)) Laravel and CMS information and technology. [Last accessed on 07-05-2018 11.00am]
- [4]. CMS application site to gather more knowledge. (<https://dnssoftware.com>) [Last accessed on 07-05-2018 4.00pm]
- [5]. CMS wiki open source (<https://github.com/cms-wiki>) [Last accessed on 08-05-2018 12.00pm]
- [6]. Github : Laravel library management system and other framework paging system. (<https://github.com/laravel>) [Last accessed on 08-05-2018 8.00pm]
- [7]. Google : To inquiry more information of administration panel features. [Last accessed on 08-05-2018 8.00pm]
- [8]. Voyager : (<https://voyager.com>) for laravel package management and routing system for application. [Last accessed on 08-05-2018 8.00pm]
- [9]. Youtube: (<https://youtube.com/laravel-tutorial>) for laravel proper uses and documentation. [Last accessed on 06-07-2018 12.00am]