# Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Mr. Niman Maharjan sir, for providing me an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although was busy.

I would not forget to remember Mr. Sudeep Bajimaya sir, for his encouragement and more over for his timely support and guidance till the completion of my project work.

Table of Contents

[Acknowledgement 1](#_Toc14967132)

[**Chapter-1** 6](#_Toc14967133)

[Introduction 6](#_Toc14967134)

[Project Introduction 6](#_Toc14967135)

[Justification of Project 6](#_Toc14967136)

[A. Background of Project 6](#_Toc14967137)

[B. Problem statement 7](#_Toc14967138)

[Aims 7](#_Toc14967139)

[Project Objectives 7](#_Toc14967140)

[Overview of the project. 7](#_Toc14967141)

[**Chapter - 2** 9](#_Toc14967142)

[Analysis 9](#_Toc14967143)

[2.1 10](#_Toc14967144)

[2.2 Feasibility Study. 12](#_Toc14967145)

[2.3 Requirement Analysis. 13](#_Toc14967146)

[2.3.1 Functional requirements 13](#_Toc14967147)

[2.3.2 Non-functional requirements 14](#_Toc14967148)

[2.3.4 Moscow prioritization 15](#_Toc14967149)

[2.3.5 Software Requirement specification 15](#_Toc14967150)

[2.4 Use Case Diagram 16](#_Toc14967151)

[2.5 NLA (Natural language Analysis). 18](#_Toc14967152)

[**Chapter-3. Design** 21](#_Toc14967153)

[Introduction 22](#_Toc14967154)

[3.1 Structural: 22](#_Toc14967155)

[3.1.1 Class Diagram 22](#_Toc14967156)

[3.1.2. Flowchart 24](#_Toc14967157)

[2.2 Flowchart for user. 25](#_Toc14967159)

[3.2 Behavioral: 27](#_Toc14967162)

[3.2.1 Activity Diagram 27](#_Toc14967163)

[3.2.2 Activity Diagram 28](#_Toc14967164)

[3.3.1 Sequence diagram 29](#_Toc14967166)

[Database 32](#_Toc14967167)

[1. Data dictionary 32](#_Toc14967168)

[Architecture 34](#_Toc14967169)

[Architecture (User Design). 35](#_Toc14967170)

[Prototyping 35](#_Toc14967171)

[1.2 Home page 36](#_Toc14967172)

[1.3 – Blog 37](#_Toc14967173)

[1.4 – Our project 38](#_Toc14967174)

[1.5 – Gallery 39](#_Toc14967175)

[1.6 – Contact 40](#_Toc14967176)

[1.7 - login page 41](#_Toc14967177)

[1.8 **-** Admin page. 41](#_Toc14967178)

[**Chapter – 4 Implementation** 43](#_Toc14967179)

[Introduction 44](#_Toc14967180)

[**Chapter- 5 Testing** 45](#_Toc14967181)

[Introduction 46](#_Toc14967183)

[1. Unit testing. 46](#_Toc14967184)

[2. Black box testing**.** 53](#_Toc14967185)

[**Chapter- 6 Others Project Issue** 64](#_Toc14967186)

[1. Risk management 65](#_Toc14967188)

[Introduction 65](#_Toc14967189)

[2. Configuration management 67](#_Toc14967190)

[Introduction 67](#_Toc14967191)

[3. Scheduling 69](#_Toc14967192)

[A. Time Estimation table 69](#_Toc14967193)

[B. Gantt chart 70](#_Toc14967194)

[C. Future Work 70](#_Toc14967195)

[D. Project limitation 71](#_Toc14967196)

[E. User manual. 71](#_Toc14967197)

[**Chapter 7- Conclusion** 72](#_Toc14967198)

[**Appendix** 73](#_Toc14967199)

# Tables of Figures

Figure 1: Picture of SWOT analysis. 12

Figure 2 Use case diagram of Dhurmus Suntali foundation. 17

Figure 3: notation used in use case diagram. 18

Figure 4: Class Diagram. 20

Figure 5: Notation of class diagram. 21

Figure 6: notation used while designing class diagram. 23

Figure 7: class diagram of Dhurmus Suntali foundation. 24

Figure 8: Flowchart for admin. 25

Figure 9: Flowchart for user. 26

Figure 10: Notation used while designing flow chart. 27

Figure 11: Activity diagram for admin. 28

Figure 12: User’s activity diagram. 29

Figure 13: Notation used in activity diagram 29

Figure 14: sequence diagram of user. 30

Figure 15: sequence diagram of admin. 31

Figure 16: notation used while making sequence diagram. 32

Figure 17: data dictionary of blog. 33

Figure 18: data dictionary of feedback. 33

Figure 19: data dictionary of gallery. 33

Figure 20: data dictionary of donate. 33

Figure 21: Entity relationship diagram of Dhurmus Suntali foundation. 34

Figure 22: three tier architecture. 35

Figure 23: Digital prototyping about us page. 36

Figure 24: Digital prototyping of home page. 37

Figure 25: Digital prototyping blog page. 38

Figure 26: Digital prototyping of our project page. 39

Figure 27: Digital Prototyping of gallery page. 40

Figure 28: Digital Prototyping of contact page. 41

Figure 29: Digital prototyping of admin’s login page. 42

Figure 30: Digital prototyping of admin page. 42

Figure 31: Donation Form. 43

Figure 32: Insert blog form. 43

Figure 33: post image and video form. 43

Figure 34: Picture of unit testing of login. 48

Figure 35: Picture of unit testing of donation. 49

Figure 36: Picture of unit testing of contact. 50

Figure 37: Picture of unit testing of deleting contact details. 51

Figure 38: Picture of unit testing of Blog–add. 52

Figure 39: Picture of unit testing of deleting blog. 53

Figure 40: UI of donation page before donation. 54

Figure 41: picture of database after successfully donated 55

Figure 42: UI of donation page with validation. 55

Figure 43: UI of Contact page before insertion of data. 56

Figure 44: picture of database after successfully insertion of data from contact form. 56

Figure 45: pictures of login form page with database picture before successfully admin logged in. 57

Figure 46: UI of dashboard pages after successfully admin logged in. 57

Figure 47: UI of Blog page before admin posting blog. 58

Figure 48: picture database page after admin successfully posted blog and data inserted into database. 58

Figure 49: UI of Image page before posting image. 59

Figure 50: picture of database pages after admin successfully posted image and data inserted into database. 59

Figure 51: UI of blog-update page before updating blog. 60

Figure 52: picture of database after admin successfully updated blog. 61

Figure 53: UI of blog-list after admin successfully updated blog. 61

Figure 54: UI of image-list page before deleting image. 62

Figure 55: UI of image-list page after admin successfully deleted image. 62

Figure 56: UI of blog-list page before deleting blog. 63

Figure 57: UI of blog-list page after deleting blog. 64

Figure 58: picture of dashboard after deleting blog. 64

Figure 59: Risk management. 66

Figure 60: Risk likelihood. 67

Figure 61: Risk consequences 67

Figure 62: Table of risk consequences 67

Figure 63: All documents uploaded on GitHub. 68

Figure 64: Tree structure of Dhurmus Suntali foundation on cmd. 69

Figure 65: Time estimation table for task. 70

Figure 66: Gantt chart of Dhurmus Suntali foundation. 71

Figure 67: Model of donation page. 74

Figure 68: view of donation page. 74

Figure 69: Controller of donation page. 75

Figure 70: model of image page. 76

Figure 71: view page of insert, delete of image page. 76

Figure 72: Controller page of image. 77

Figure 73: Model of contact page. 78

Figure 74: View page of insert, delete of contact. 78

Figure 75: Controller of contact page. 79

Figure 76: Model of Password Change page. 80

Figure 77: view page of Password change page. 80

Figure 78: Controller page of password change. 81

Figure 79: Middleware of authentication. 82

Figure 80: view page of authentication. 82

Figure 81: Model page of Blog. 83

Figure 82: view page of blog. 83

Figure 83: View of Blog listing page. 84

Figure 84: View page of blog add, update, delete page. 84

Figure 85: Controller page of Blog. 86

# Chapter-1

# Introduction

## 

## Project Introduction

Dhurmus Suntali foundation is a non-profitable charity website working for foundation specially developed for raising funds, Awaring people from social problems, displaying the problems or critical situation happening in Nepal through website. This website provides the major problems with justification in Nepal in different fields such as education, health, poverty raising up due to natural disaster like storm, earthquake, flood, landslides and so on in details.

## Justification of Project

### Background of Project

The system is developed to raise up maximum amount of fund to help for landslide, earthquake flood victim people. So that user can donate everything to foundation as their wish including money. Which will support foundation to raise up funds for victims.

Furthermore, user can see the every projects which is divided into three different basis that is completed, running or upcoming in details. Not only this, user also can see the condition of the victims, the foundation progress and the victim’s problems through the images in the gallery page.

In addition, for national and international donators they have to transfer their money for donation in foundations bank details which is provided in websites home page and can see their name in donator’s details page. Recapping, if the foundation will organize any events then you can obviously see it day by day. And for this project I have used PHP programming language with laravel and MySQL to handle database over the web.

## Problem statement

There are a small number of problems related to donation through online. Those who provide e-banking facilities for instance PayPal in Nepal, and to bring that facilities into Dhurmus Suntali website for direct transaction, yet that agreement has not been done. So in that case for international donators they have to transfer their money through bank to foundations bank details which are provide in home page of the website. Or, they should move in office and handover the donation by cash or by cheque or anything else as their wish.

## Aims

The aims of this project is

* To attract high amount of national or international investors or donators by displaying the social problems in website.
* To develop fully dynamic website for CRUD operation like insert, update, delete like blog, images etc.

### Project Objectives

The main objective of the Dhurmus Suntali foundation are:

* To display more problems in website so that fund can be easily collected.
* To build a system fully automated and responsive.
* To raise maximum amount of fund.
* To save the donators time.
* Donators should see the name in donator’s details page.
* To collect the reviews from the user to improve the foundations website.

## Overview of the project.

* The Dhurmus Suntali foundation website is developed for a means of message to display the problems with details to the world and to ask for donation.
* It will make easier for donators to donate and for foundation to collect fund to help victims.
* This website show different problems, events which are now happening in Nepal with details and as well as it provide the details of the donators who donate in need, also display the information about the different projects which divided into different parts as completed, ongoing and future projects where user can see all of these.
* User also can see the condition of the victims, progress of foundation and the victim’s problems through the images in the gallery page. And can send the review of website with their details to the admin about the website.
* But there is a small problem regarding with this website that is, in the current situation direct donation facility through this website is not available but founders has done agreement with those who provide online transaction services like e-sewa, PayPal etc.
* Still PayPal is also not available so for international donators it will be little bit difficult to donate so in this case they have to transfer money through the bank to the foundation bank details which is also provided in the home page of the website.

# Chapter - 2

# Analysis

# 2.1

The breaking of the subsequent elements into simple one is known as analysis. We need to perform analysis to confirm the necessities can be included in the software without any problem.

Requirement, Natural language analysis (NLA), Use Case, class diagam and activity diagram are included in this part. The explanation of those analysis are as follows:

**Requirement:** it is the process of gathering all the information of client expectation before developing the system. This is the starting and most important and this features called requirement should be quantifiable, relevant, and then should be in detailed.

**NLA:** Natural language analysis is done after the requirement. This is done by separating the noun, verbs and objectives to made class diagrams and to made classes and objects.

**Use Case diagram:** Use case diagram define the essentials of actors and their collaborations with the system. It is used to increase extra imaginative textual use case.

**Class diagram:** It is the structured diagram that defines the system’s classes, attributes, operations and the relationship of the objects which is used for data modeling.

**Activity diagram**: It is the process of the system which is used to describe the flow of note from one activity to another.

Here in my project I am applying the SWOT analysis because it determine which features are essential to the website’s accomplishment and what can harm my website and also it focus on both internal and external factors. By performing this analysis I as well as other choice makers can understand my website’s weakness, strength, opportunity and threats that can affect my organization.



Figure 1: Picture of SWOT analysis.

The strengths and weakness focus on **internal issues**, while opportunities and threat focus on **external issues**. Likewise, PEST analysis only focus on the external factors whereas CATWOE only provides means to decide the issue of different view. So, I have decided to implement SWOT analysis.

The factors of my project that comes under SWOT analysis are as follows:

**Strength:** Responsive design with full mobile support, operative call to action, suitable and important contents, quick and easy checkout process, and banking details other extra details for donation and direct donation through website and also through e-sewa. .

**Weakness:** language translator.

**Opportunity:**  including features like language translator.

**Threats:** Deceitful activity, browser could upgrade.

# Feasibility Study.

Feasibility study is the early design period of any project, which brings together the elements of knowledge that specify if a project is possible or not. They are used underneath many conditions like whether the company has enough money and human resources for a project? To find out will the created product will be sell or not and so on.

The types of feasibility study are as follows.

1. **Economic feasibility:** it is used to identify the financial resources of the company and it is also the cost benefit analysis because it derived the total budget for development of a new system and benefits derived from the system. My project comes under the budget and there won’t be any economic problem while developing this system. Without any doubt it is possible to develop the system.
2. **Technical feasibility:** it is used to identify that is technical resources are available or not to create a project. And also measure will the hardware and software of the current system support the proposed project. Yes the current hardware and software will support this system and also all technical resources are available so, there is no problem in this feasibility.
3. **Schedule feasibility:** it is used to identify that does the company have enough time to start the project or can the project will be completed at the time? I have already maintained the time schedule and they are accurate that defines the project can be completed in time.
4. **Operational feasibility:** this will measure how well your company can identify and solve the problem and also measure the solution of the problems. As I have implemented SWOT analysis for this system and this can identify all the internal and external problems which I can solve it easily quickly.
5. **Legal feasibility:** this will identify that does the company met all the permitted and ethical requirement? The system that I have planned to develop this is legal and under the laws of my country and in feature my project won’t create problem which is legally possible to develop this system.

# Requirement Analysis.

Requirement analysis is the process of identifying the client’s expectation for a new or modified product.

The type of requirement analysis are with it explanation are as follows:

* + 1. Functional requirements:

It specifies that what system should do for example in my system the functional requirements are:

* Interface requirements.
* Blogs
* Gallery
* Authentication
* Sending feedback by user.
* Admin panel.
* Donator’s details.
* Causes.
* Contact information including google map
* Pop up information.
* Donation form
* Call to action button

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F.ID | Function | Data | Rational | Dependency | Remarks |
| F01 | Login | Admin’s username and password | Security | F01 | Login for accessing admin panel |
| F02 | Admin Panel | No data needed | Posting activities | F01 | Posting images, blogs, donator’s details. |
| F03 | Blogs | No data needed | Visiting page | F02 | General user visiting page. |
| F04 | Gallery | No data needed | Visiting page | F02 | General user visiting page. |
| F05 | Donators details | No data needed | Visiting page | F02 | General visiting page. |
| F06 | Feedbacks | No data needed | Sending feedbacks page | N/A | Simple feedback sending page. |

## Non-functional requirements

It is used to derive that how the system works or behave. For example: scalability, performance, availability, security, maintainability and so on.

|  |  |  |  |
| --- | --- | --- | --- |
| N.ID | Function | Rational | Remarks |
| N01 | Responsive | Support different resolution | Fits in different sizes of devices and also support in every browser. |
| N02 | Reusability | Easy to use. | The system code can be reused or can be taken for references. |
| N03 | Scalability | To be able to handle overflow of workloads | The system can handle the flow of incoming data. |
| N04 | Security | To protect the system form being hacked and cracked. | Security precautions shall be taken. |
| N05 | Performance | To maintain system’s productivity | Good design and regular tests to be focused |
| N06 | Usability | Easy learning and user friendly. | Codes are simple and interface are user friendly. |
| N07 | Maintainability | Easy to change. | Easy maintenance and optimization. |
| N08 | Reliability | Exact output | Gives accurate output as per input given. |

## Moscow prioritization

It is the most widespread procedure for handling requirements and normally used to help basic stakeholders recognize the meaning of advantages in a particular statement. It has three types which are explained below in detail:

|  |  |  |  |
| --- | --- | --- | --- |
| **Must have** | **Should have** | **Could have** | **Won’t have** |
| Bank details for donation,  Post blog,  Donator’s details. | login,  Call to action,  Feedback,  Contact,  Responsive. | Gallery. | Videos. |

The Moscow prioritization of non-functional requirements are listed below:

**Must have:** Responsive, Reusability, Scalability, Security, Performance, Usability, Maintainability and Reliability.

## Software Requirement specification

**Software requirements**

* Window vista, 8.1, 7, 10 or Linux.
* Any browser such as: Firefox, internet explorer, chrome and so on.

**Hardware requirements.**

* Processor: i3, i5, i7.
* Memory: minimum 4 GB RAM.
* Video card: Intel HD 5500.

# Use Case Diagram

Use case diagam is pictorial representation of user’s interaction with the system that express the relationship between the operators and different use cases in which the operators is involved. Here in this diagram I have defined two actors as admin and user where admin can perform CRUD operation and login whereas user can send feedback and then view all the things that admin has posted in the website.

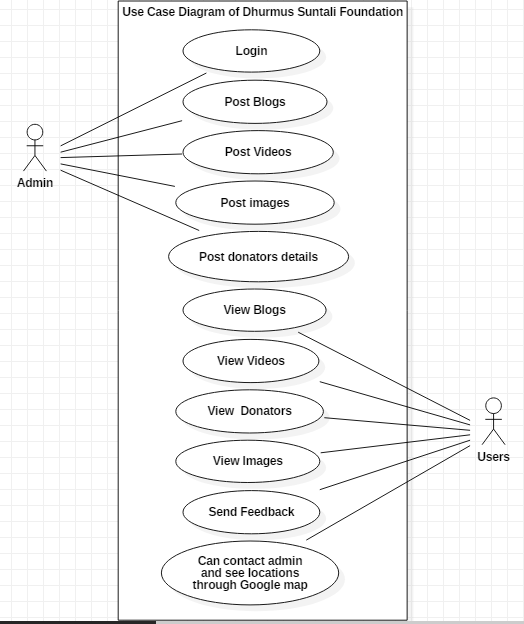


Figure 2 Use case diagram of Dhurmus Suntali foundation.

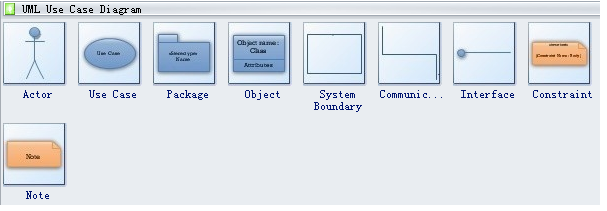


Figure 3: notation used in use case diagram.

|  |  |  |
| --- | --- | --- |
| ID | Title | Describe |
| 1 | Login | Authentication |
| 2 | Post blog | Displaying information |
| 3 | Post image | Display image |
| 4 | View blog | User can see all the information posted by admin. |
| 5 | View images on gallery | User can view all the images on gallery page posted by the admin |
| 6 | View donators details | User can see their as well as others donation details. |
| 7 | Can contact admin and view the location through the google map. | User directly can contact to admin through email or phone number or can track the location through google map. |
| 8 | Can send reviews of website. | User can send review of the website from the contact page. |
| 9 | View review. | Admin can view their website review in admin panel. |

Table of explanation of the cases.

# NLA (Natural language Analysis).

Dhurmus Suntali charity Foundation website show different problems, blogs which are now happening in Nepal with details and as well as it provide the details of the donators who donate in need, also display the information about the different projects which divided into different parts as completed, ongoing and future projects where user can see all of these. User can see pictures from the gallery pages and can send their reviews. Except this all admin can post blogs, videos, images, donator’s details and so on.

But there is a small problem regarding with this website that is, in the current situation direct donation facility through this website is not available which means founders has not done agreement with those who provide online transaction services like eSewa, PayPal etc. but in this case they have to transfer money through the bank to the foundation bank details which is also provided in the home page of the website.

Nouns: Nepal, blogs, donators, donate, information, gallery, admin, images, donation, eSewa, PayPal, money, bank, transaction, home, website.

Possible class: blog, gallery, Login, feedbacks, users.

From the above scenario we can classify the potential classes and methods as:

|  |  |
| --- | --- |
| **Potential classes** | **Potential methods.** |
| Users | View donators details |
| Admin | Delete images. |
| Feedback | Add, update and delete Blogs. |
| Gallery | Add Images to the gallery |
| Blog | Give feedbacks or send messages. |
| Admin login |  |

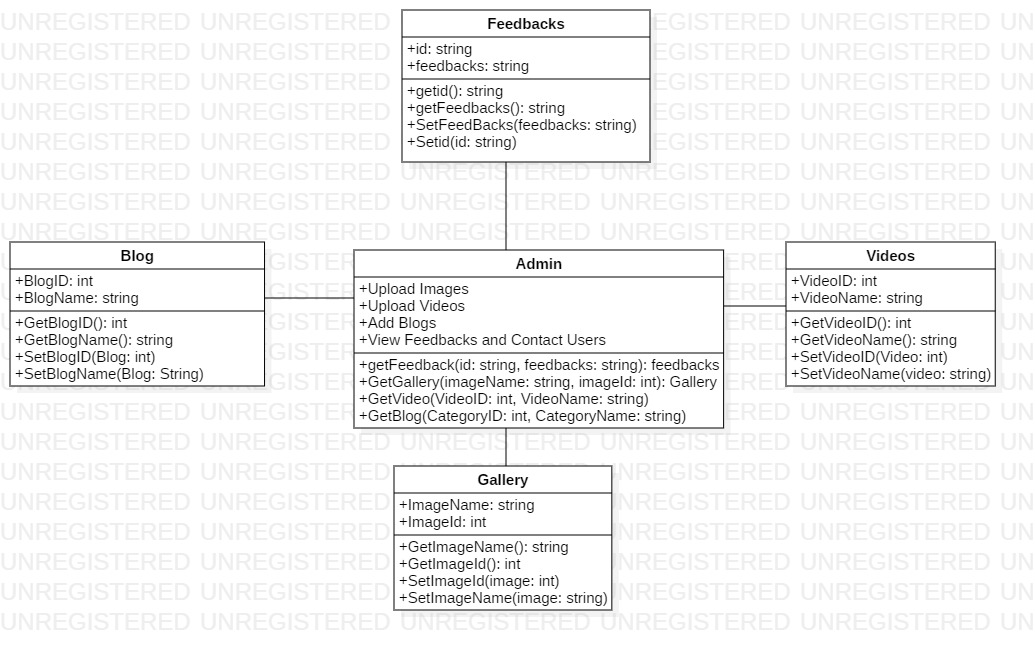


Figure 4: Class Diagram.

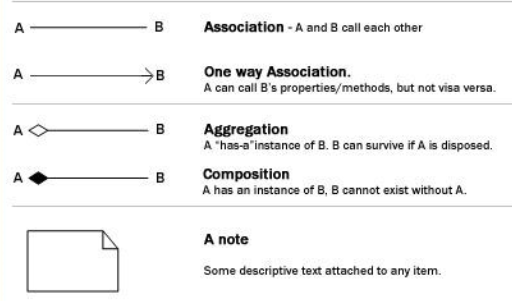
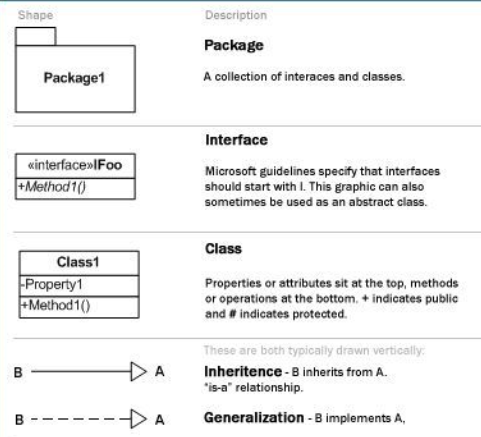


Figure 5: Notation of class diagram.

# 

# Chapter-3

# Design.

# Introduction

In Software development lifecycle design is the subsequent phase where the system is developed based on the user requirements. It is divided into three parts which include structural, behavioral and database design. here I have made diagrams like activity, flowchart, class diagrams etc. using tools like star uml to design the project according to users requirements by taking references from analysis phase.

## 3.1 Structural:

Structural design is the pattern that easiness association between entities and also displays the constant view of the system.

### 3.1.1 Class Diagram

The constant diagram which defines the system classes, attributes, methods and then relationships among the objects. I have designed class diagram for the Dhurmus Suntali foundation involving methods, classes and attributes.

* The admin class perform CRUD operation over further objects.
* The blog class have operations to display the daily journal on the website, read the comments and then add, edit and delete them.
* The gallery class have another operations to display the images of the foundation and can add and delete it.
* The videos class is also same like gallery which display the videos related to foundations which is responsible for adding and deleting the videos.
* Feedback class shows the feedback sent by the user and through which user will be contacted by the admin.

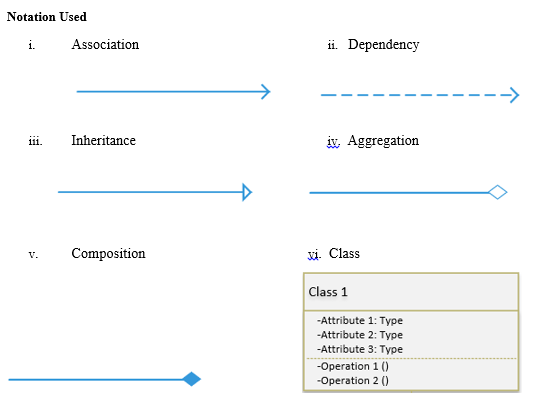


Figure 6: notation used while designing class diagram.

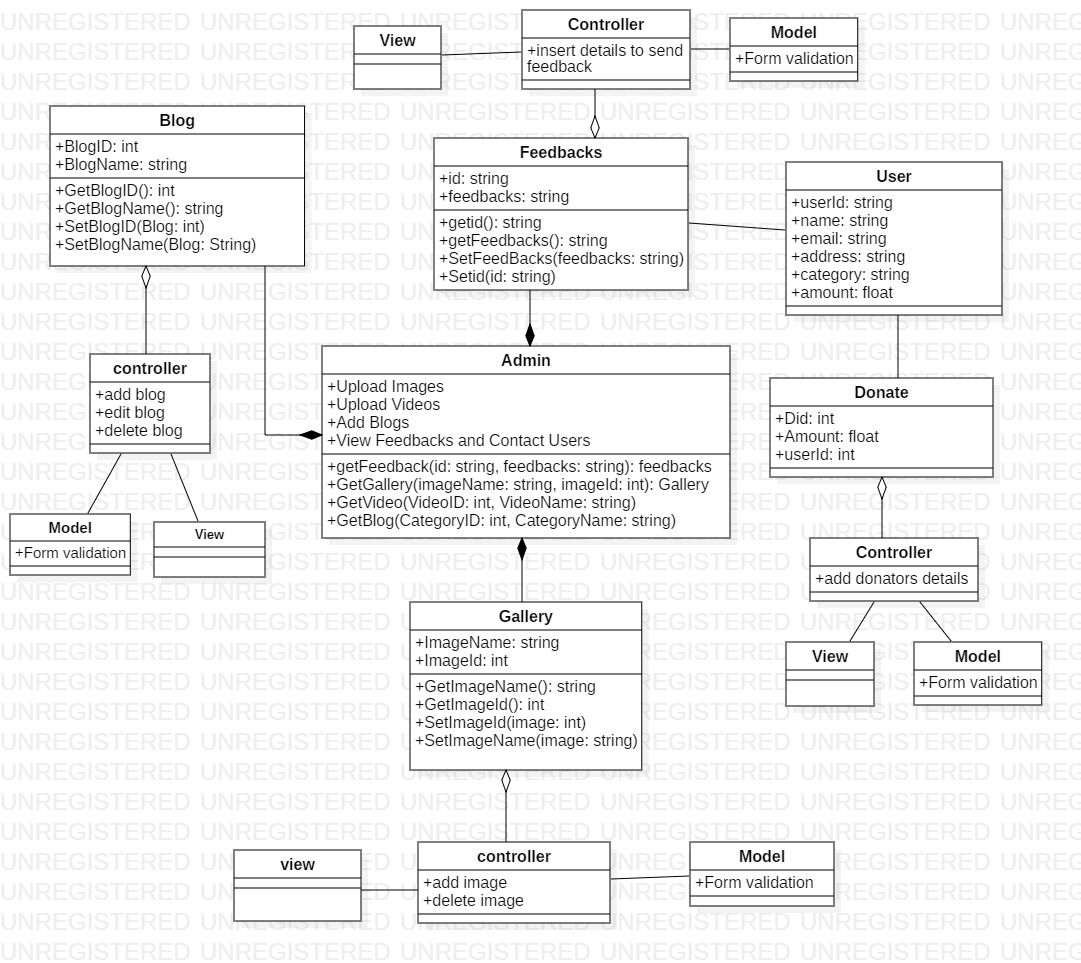


Figure 7: class diagram of Dhurmus Suntali foundation.

Justification

* The above class diagram is designed in star uml software in MVC (Model, View, and Controller) pattern which is clearly shown in the diagram and also shows that how these pattern are related with each other.
* To show relationship between classes I have used aggregation, composition and association. The composition is used because the composed object like controller cannot exist without another object like gallery. Likewise aggregation is used because without gallery admin can exists to so that composition is used in every main classes.
* And from the above shown class diagram admin has the ability to perform all the operation so all classes are connected with it.

### 3.1.2. Flowchart

Flowchart is the diagram that solve the task by step by step method. It is also used to clarify the logic behind the program before starting the code. I have choose this diagram rather than DFD (Data Flow Diagram) to spell out the logic of my system because it is easy to design and easy to understand make other understand.

## 

Figure 8: Flowchart for admin.

Justification

* The above flowchart diagram is for admin to show how admin can perform overall all operations after login in the system.
* To show the flow of diagram I have used flow line, decision box for if else statement and then shown process in rectangle box.
* After login in the dashboard opens and then he can add, edit and delete the task like image, videos and so on. if the task is successfully completed then the admin logout and check out in home page otherwise admin repeat that operation again and again.

## 2.2 Flowchart for user.

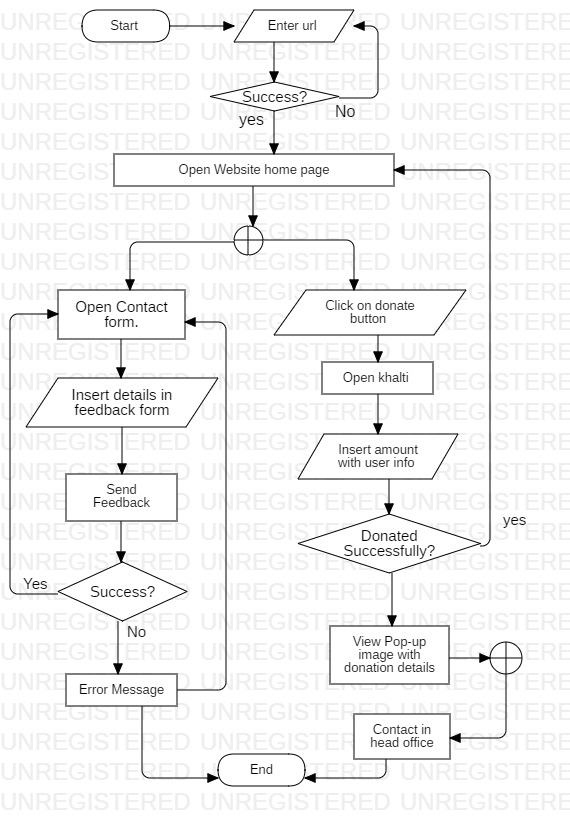


Figure 9: Flowchart for user.

Justification

* The above flow chart diagram shows that flow of system after accessing the website. First user either open the donation form to donate or send feedbacks to the admin about the website so circle shape with cross sign is used to show as ‘or’ option.
* If user donate the amount and if it is donated successfully then he return homepage after seeing successful message.
* Otherwise he return to the view page to see the pop up image to see donation details and then donate according to that details or the donators contact in the head office. So here again that circle shape with cross sign is used to show as ‘or’ option.

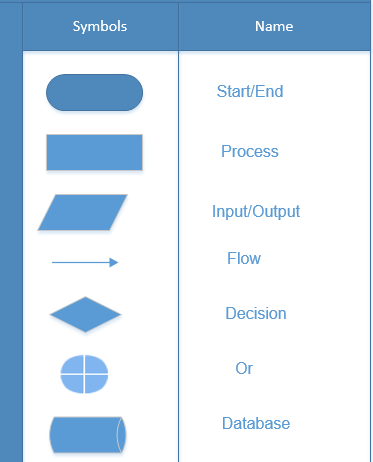


Figure 10: Notation used while designing flow chart.

## 3.2 Behavioral:

The behavioral design pattern shows how the system interact with each other with one object to another object which signifies the relations among the structural diagrams and also show the dynamic environment of the system. It consists of various diagrams like activity, use case and sequence diagrams.

## 3.2.1 Activity Diagram

Activity diagram shows the movement from activity to activity inside the system which also address the dynamic view of the system.

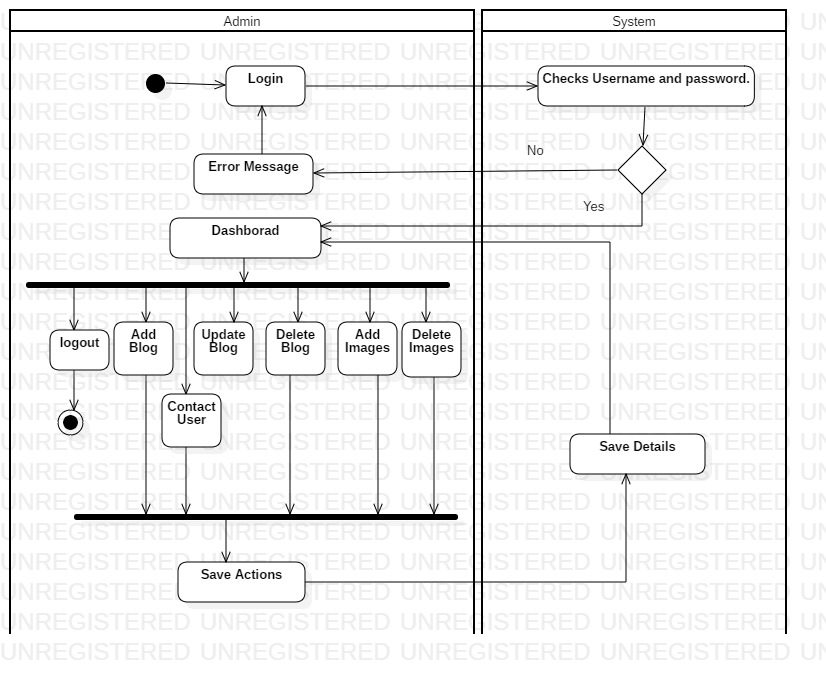


Figure 11: Activity diagram for admin.

Justification

* The above activity diagram is for admin’s activity and also display the whole activities involved during the duration of the system.
* After login in the dashboard opens and then he can add, edit and delete the task like image, videos and so on. If the task is successfully completed then the performed action is saved and also save the saved actions details in the system and then he return back to dashboard and logout. Hence overall task ends according to the above shown activity diagram.

## 3.2.2 Activity Diagram

# 

Figure 12: User’s activity diagram.

Justification

* The above activity diagram is for user’s activity and also display the whole activities involved during the duration of the system.
* The above activity diagram illustrates that when user access home page then the user do various operation like send feedbacks, donate etc. but when the user donate and if it is successfully donated then the details is saves and then return back to the page otherwise they see the details to donate from the home page and donate from the bank or contact in the head office.

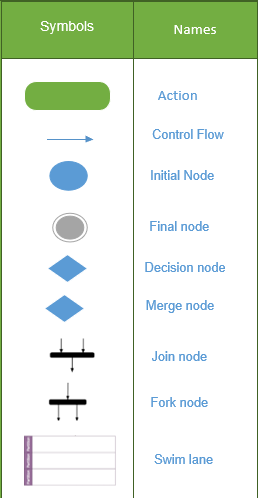


Figure 13: Notation used in activity diagram

## 3.3.1 Sequence diagram

Sequence diagram are the diagram that display interaction in the time order and show how the operation are carried out.



Figure 14: sequence diagram of user.

Justification

* The above sequence diagram illustrates that when the user access website the they either donate or do other things but if the donate then they donate from the system which flow of line is connected with the activation box from user and then donation is save in the database. If the donation is successful the user return to home page otherwise retry that process again.
* And in the case of sending feedback if the user want to send null data then it won’t let it to send that because before sending form validate whether there is null data or not so there is curved arrow touched in the activation box which is also called as self-message.

## 

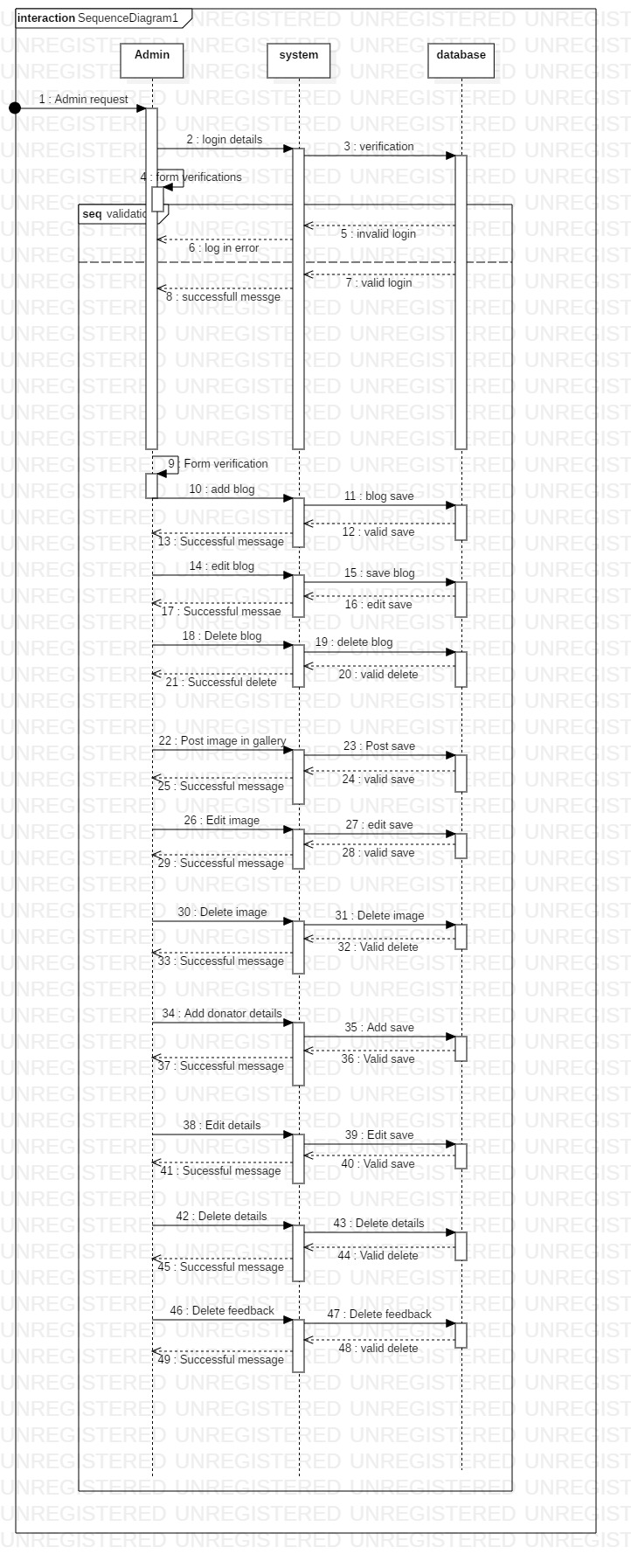


Figure 15: sequence diagram of admin.

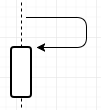
Justification

* The above sequence diagram display the overall operation of the admin. First of all, the admin log in to access the admin page if the login is successful then he can do rest of the work otherwise admin return with error message in login form again. This process repeats until and unless the admin successfully logged in.
* And when the admin is successfully logged in then admin can perform the task like add, update and delete blogs, images etc. here again form validation is performed in every form to prevent admin from sending null data so if admin wont fill any field then it won’t let it to send that data. After successfully completing the task then admin return back to the dashboard and do other work or log out from there.







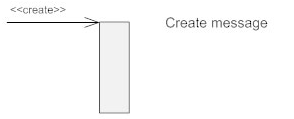


Figure 16: notation used while making sequence diagram.

# Database

## Data dictionary

Data dictionary is the collection of the information that shows the name, data type, length, key while creating table on the database. Here I have use MS Excel to design data dictionary of each classes.

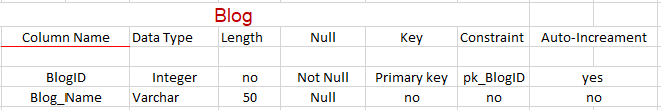


Figure 17: data dictionary of blog.

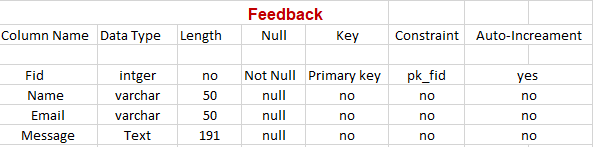


Figure 18: data dictionary of feedback.

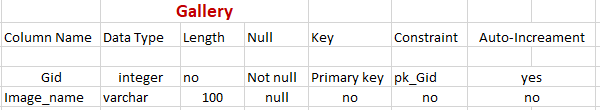


Figure 19: data dictionary of gallery.

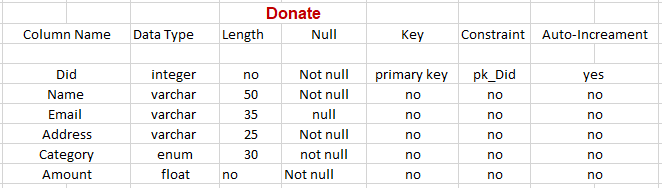


Figure 20: data dictionary of donate.

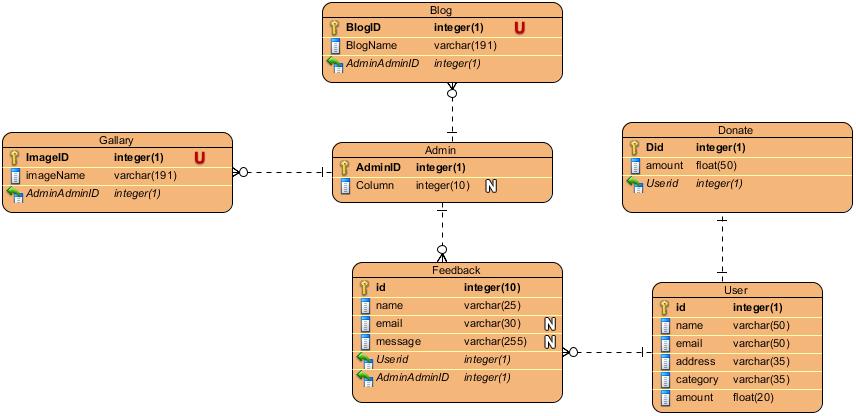


Figure 21: Entity relationship diagram of Dhurmus Suntali foundation.

**Justification.**

* Here I have made the Entity relationship diagram of my project in visual paradigm. Because many professionals use this software to design ER diagram and also I have used this software before too so that I feel comfortable to use it.
* And the above diagram shows that different table connected to each other. All the tables are connected with admin with one to many relationships because admin can perform the task multiple times like admin can create many blogs, can post many images into the gallery so they are linked with one to many relationships.
* There is also another table having relationships because one user can donate only one time so it is linked with one to many and also one user can send many feedbacks do it is connected with one to many relationship.

# Architecture

I have choose three tier architecture to do this project because

* It individually helps to sustain the elements on single platforms and then it develop system’s functional procedure, graphical user interface, system storage.
* I have chances to restructure my project or product and it not only look for present requirements but also for futures requirements, which will be easy for me to adapt changes made by the customer. So, I don’t think that I can get this features by using others architecture. That’s why I chose three tier architecture.

Basically, it is divided into three parts:

**Presentation layer** which help to display the web page using HTML/CSS/JS and whatever data is send by the user from the form which is contained in web page those data are delivered to the **application layer** and it runs the query and finally distributed to the database layer and at last the **database layer** perform query in database management system and then return the results to the application layer which setups it into the web page. It is also explained in the figure:

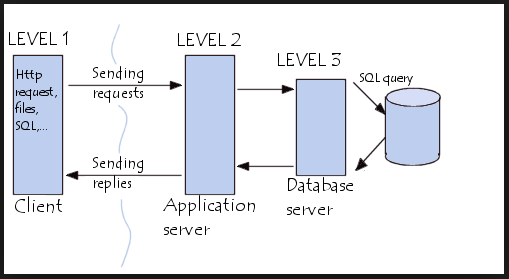


Figure 22: three tier architecture.

The advantages of three tier architecture are:

* Complex and enormous projects can be easily understand and maintain.
* The one tier can be easily update or maintain without impacting other part of the application layer.
* The modules can be reused.

# Architecture (User Design).

Architecture design is the way of representing the design of the software which also weighs all top level of design. Making user design makes easier to understand the system and also makes the judgement making method well-organized so that it is important to design.

## Prototyping

To design the prototype I have used software name balsamic mock up because

* It can be used in any type of device like laptop, tablet, desktop etc.
* It is fast, trustworthiness software and also can make prototype of any kind of software.
* It has features of drag and drop which saves time and resources and also it is secured than paper prototyping. I can redesign if the prototype is wrong by using balsamic but designing in paper I have to draw in another new paper which lost time and resources.



Figure 23: Digital prototyping about us page.

## 1.2 Home page

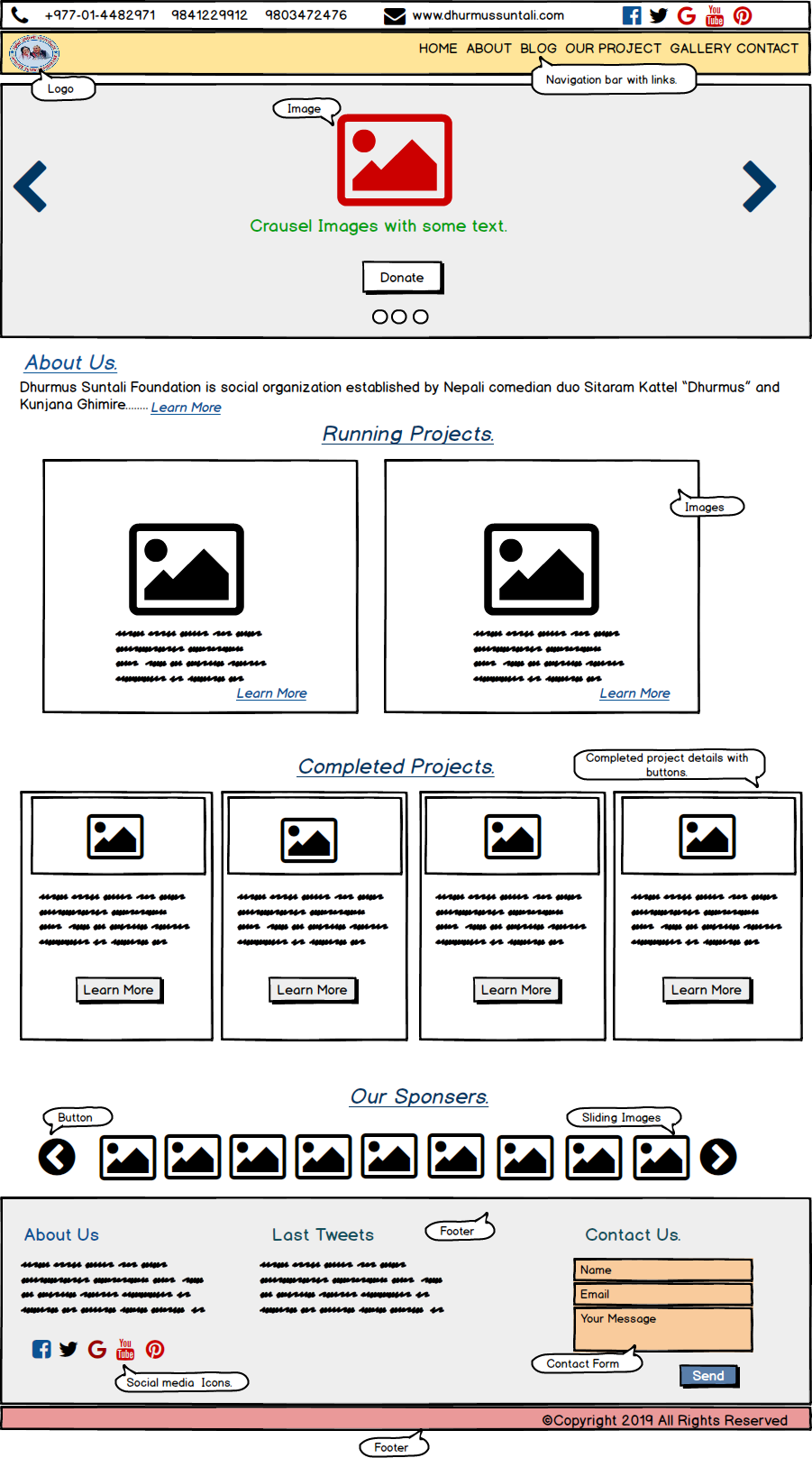


Figure 24: Digital prototyping of home page.

## – Blog

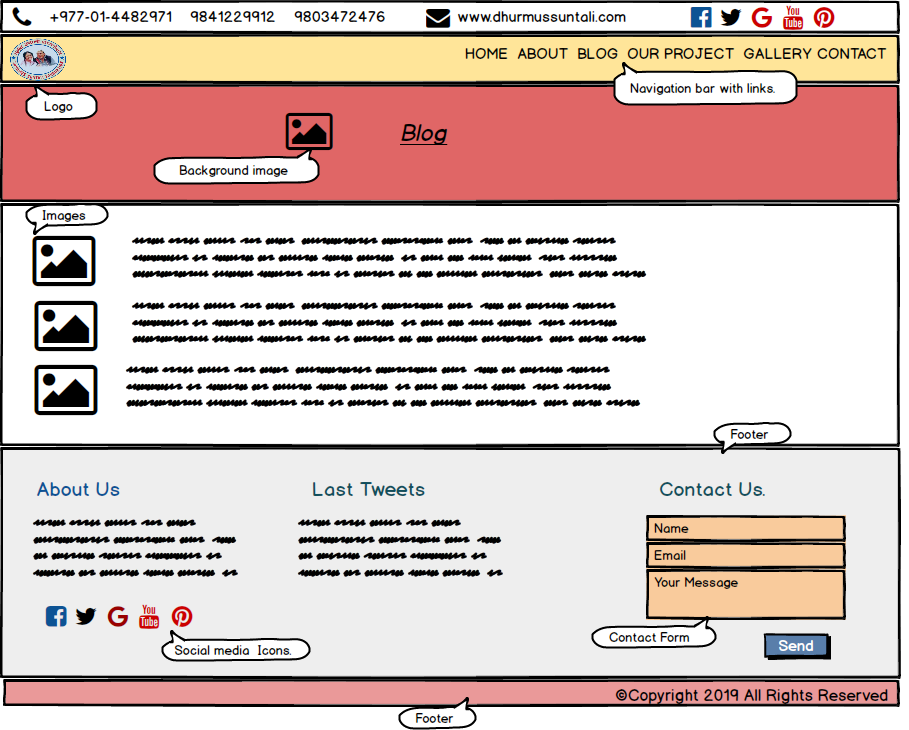


Figure 25: Digital prototyping blog page.

## – Our project

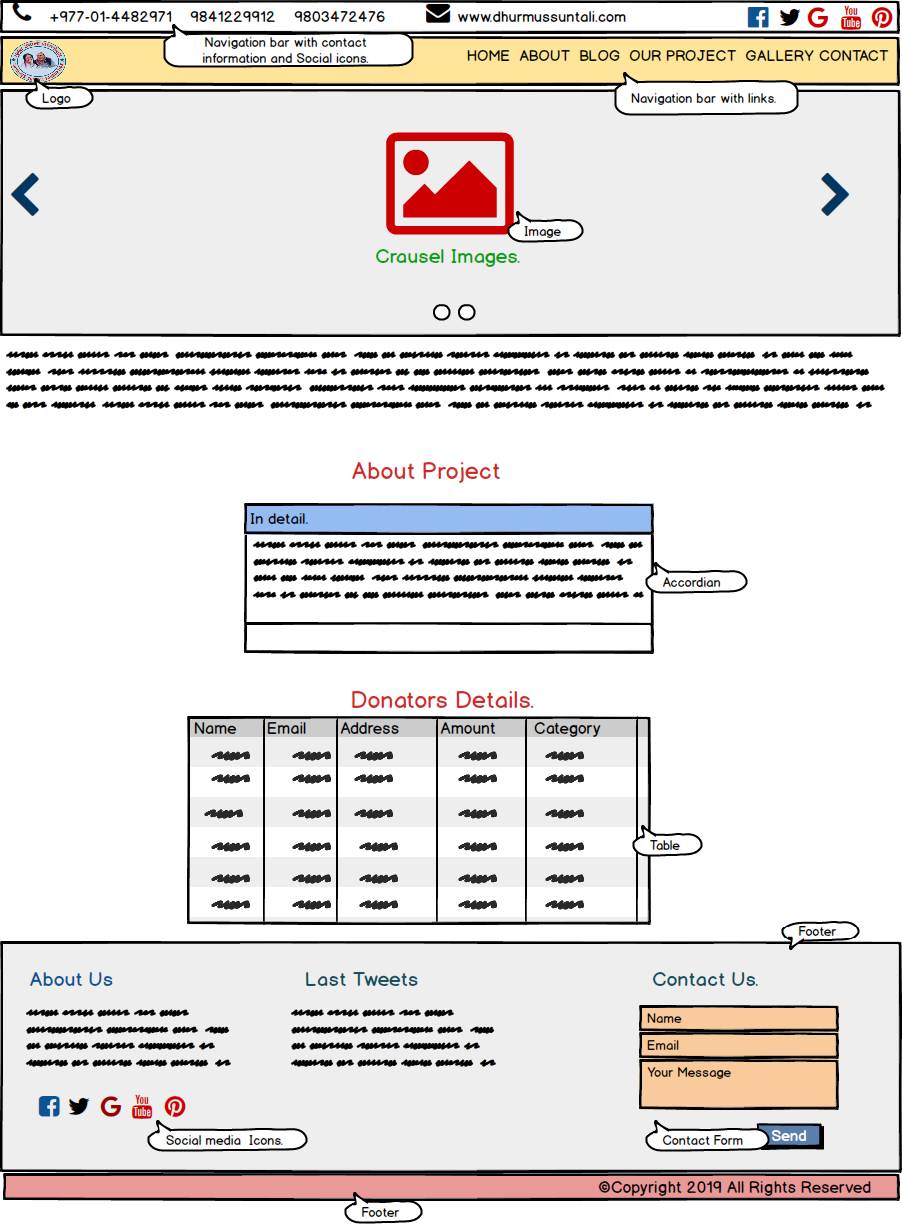


Figure 26: Digital prototyping of our project page.

## – Gallery

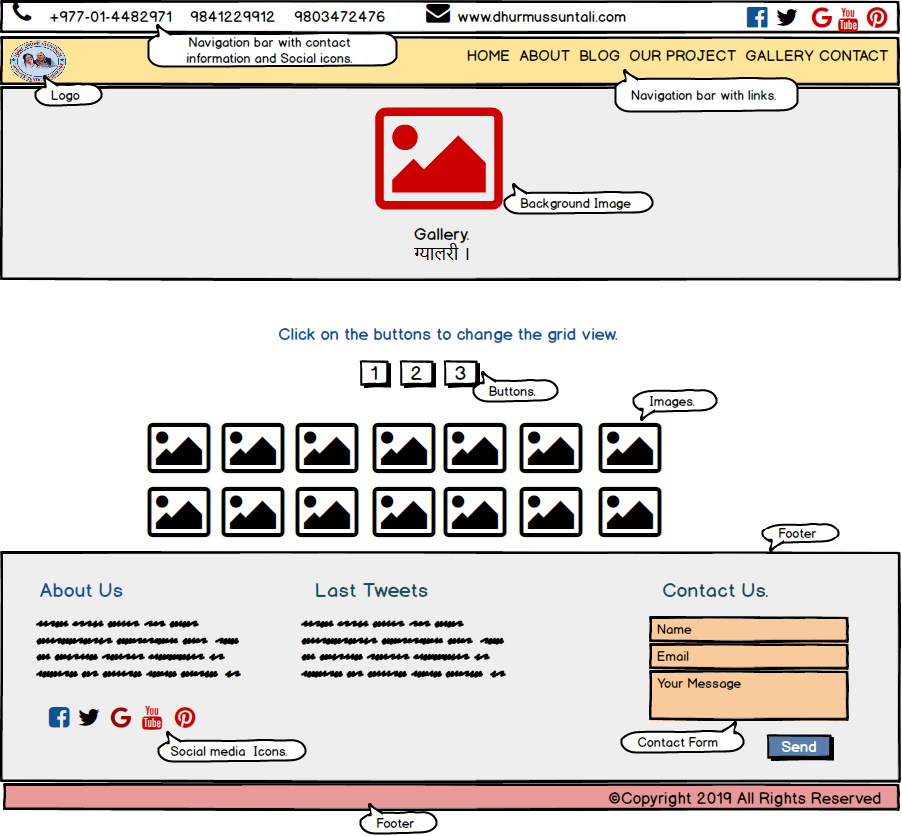


Figure 27: Digital Prototyping of gallery page.

## – Contact

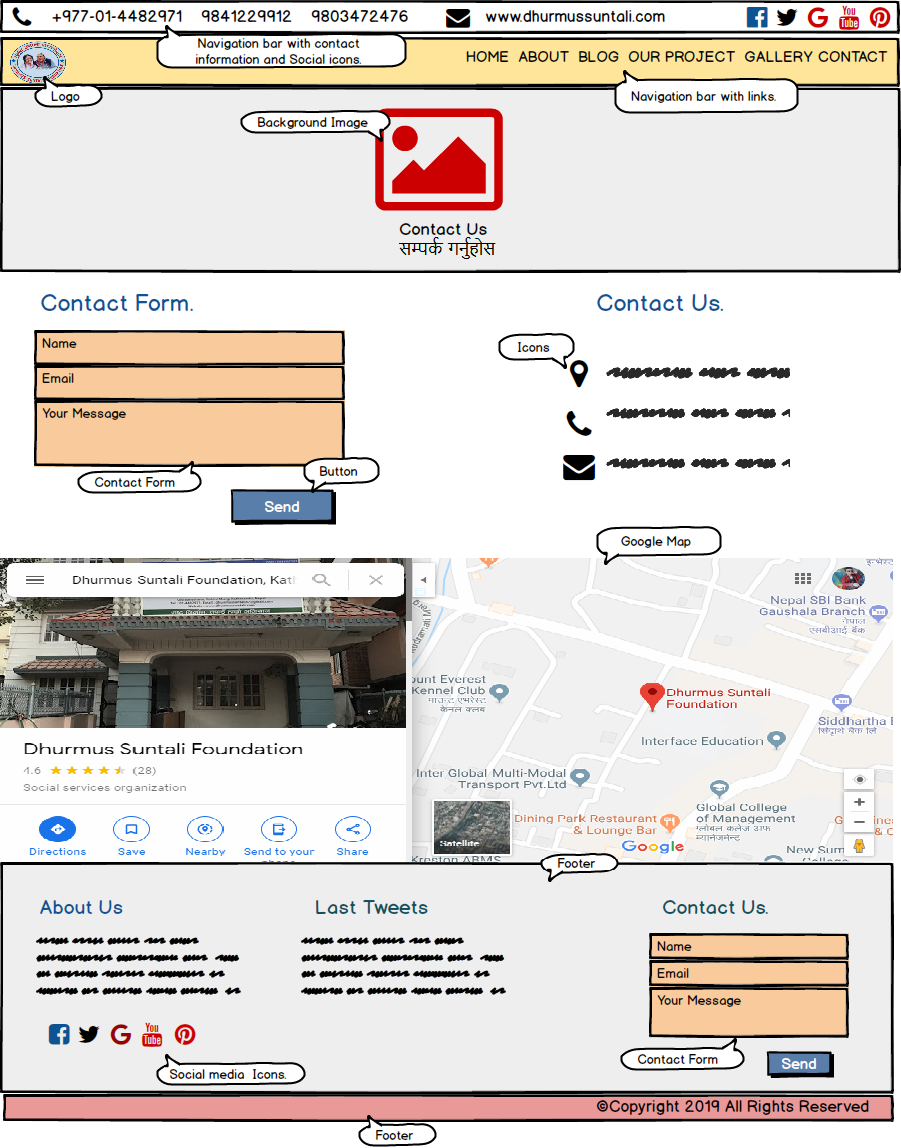


Figure 28: Digital Prototyping of contact page.

## 1.7- login page

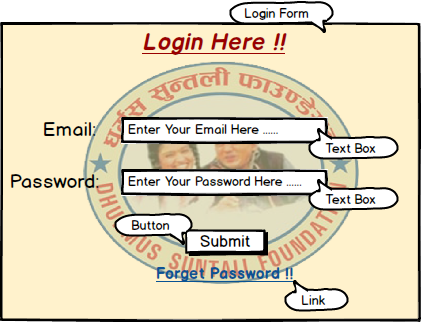


Figure 29: Digital prototyping of admin’s login page.

## **- Admin page**.

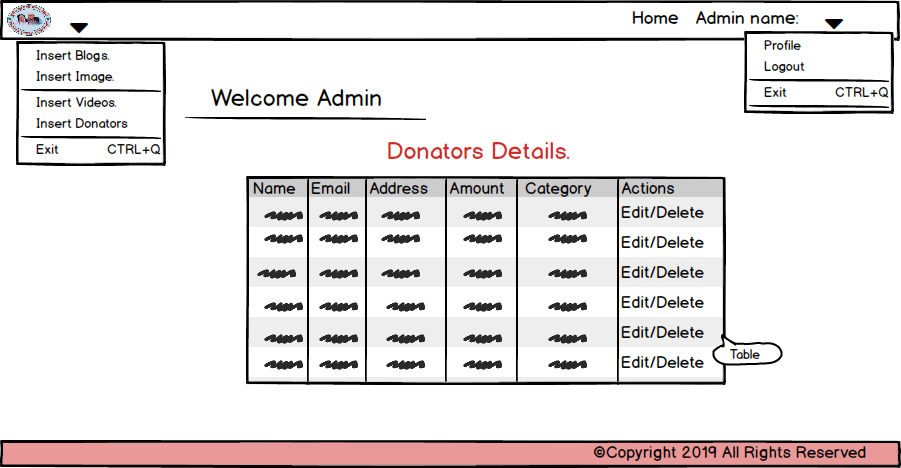


Figure 30: Digital prototyping of admin page.

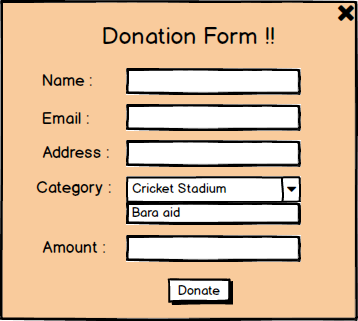


Figure 31: Donation Form.

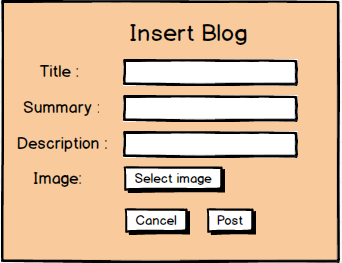


Figure 32: Insert blog form.

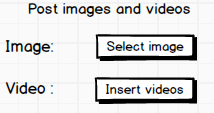


Figure 33: post image and video form.

# 

# Chapter – 4

# Implementation

# Introduction

The implementation is the fourth phase of process in software development life cycle (SDLC) which is also stated in the design file into workable programming language code. This phase includes three parts: coding, combination and testing and finally installation. And in implementation phase I have used PHP as programming language in laravel framework which is explained in more detailed below:

**Language, code editor and database.**

I have used PHP programming language with HTML, JavaScript and Css for coding in visual studio code editor to develop this project because I am familiar with this language and fell easy to do coding in visual code editor instead I could use python, java also but I don’t know those programming language also and I am more familiar with PHP and fell easy to write so I used this programming language for coding. Likewise, to handle the website database I have used PHP myadmin database because it is easy to use and popular too and JavaScript for the animation.

**Framework**

I didn’t wrote code in core PHP rather than that I chose laravel to do it because developers made laravel for both beginners and for professionals.

While doing coding in MVC pattern in core PHP It takes time and the time given to complete this project is also not enough so I chose laravel framework, where MVC pattern is already inbuilt inside it.

Not only MVC pattern almost all pattern are inbuilt here just we have to write code inside it. It has made so easy for us that it has its own authentication. If I write one line of code in terminal to create authentication then it will automatically create login and registration part which reduce our time to complete the project. But in core PHP I have to write all this code.

In laravel debugging is the very easy than in core. Here I should not search errors line by line like in core PHP, in laravel it directly shows the errors in which part of line there is error. And also almost all IT companies today use laravel framework to develop website. It is the most popular programming language in 2019 to build website.

So, these are the reason that I had chosen laravel framework instead of core PHP.

# Chapter-5

# Testing.

# Introduction

Testing refers to the process of checking actual result with expected result of the system to confirm that the system works nicely and is bug free.

Due to following reasons software testing is important:

* Defects in software can cause economic, time and reputation loss.
* It improve reliability, performance and security.
* Also it improve quality of software and satisfy customer.

There are various types of testing like UI testing, regression testing, black box testing, white testing, unit testing and so on. From the above type of testing I am doing

* Unit testing and black box testing.

## **Unit testing.**

## 

It is the process of testing each module to make a decision whether the developer has made any mistake or not.

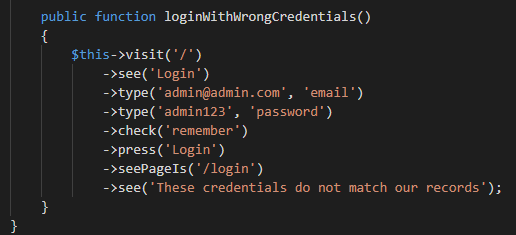
The reason behind choosing unit testing are as follows:

* It reduce bugs in the software.
* The software also can be tested before developing it completely.
* It improve code architecture and helps developers to attain their aims quickly.

Test Name: Login Test.

Pre-condition: User must successfully logged in.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 1 | To check if the value inserted is working properly. | Email: admin@admin.com  Password: admin123. | The data should be inserted into the database and user should be redirected to the donation page. | Successfully inserted data is matched form the database and admin is able to access the dashboard page | Pass |



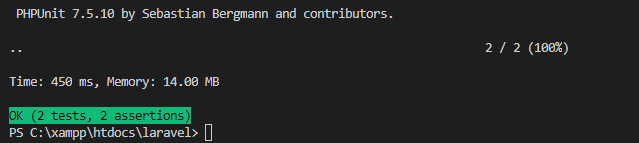
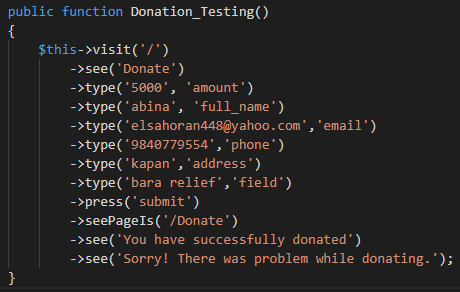


Figure 34: Picture of unit testing of login.

Test Name: Donation Test.

Pre-condition: User must successfully donate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 2 | The value inserted for login should work properly. | Amount: 5000  Name: abina  Email:  Elsahoran448@yahoo.com  Phone: 9840779554  Address: Kapan  Field: Bara Relief. | The data should be successfully inserted into the database. | Successfully data is inserted into the database. | Pass |



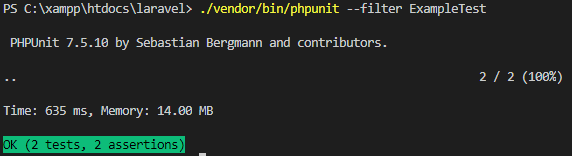
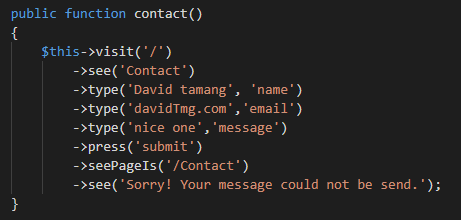


Figure 35: Picture of unit testing of donation.

Test Name: Contact Test.

Pre-condition: User must successfully send message with user details to contact.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 3 | To check if the value inserted is working properly. | Name: David tamang.  Email: davidTmg.com  Message: nice one. | The data should be successfully inserted into the database. | Successfully data is inserted into the database. | Pass |



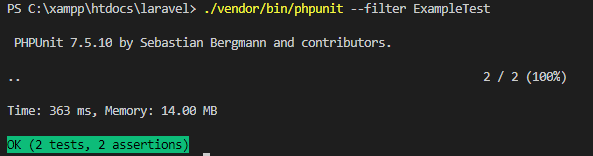
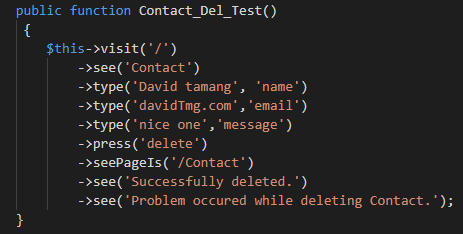


Figure 36: Picture of unit testing of contact.

Test Name: Contact-Del Test.

Pre-condition: Admin must successfully delete the contact details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 4 | To check if the selected value is deleting properly from database. | Name: David tamang.  Email: davidTmg.com  Message: nice one. | The data should be successfully deleted from the database. | Successfully data is deleted from the database. | Pass |



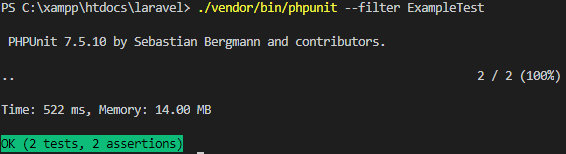
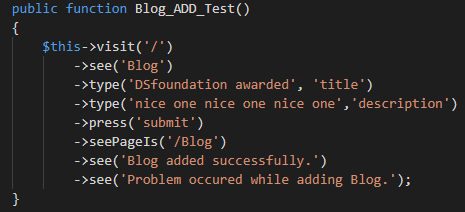


Figure 37: Picture of unit testing of deleting contact details.

Test Name: Blog-add Test.

Pre-condition: Admin must successfully post the blog.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 5 | To check if the value inserted is working properly. | Title: DSfoundation awarded.  Description: nice one….. | The data should be successfully inserted into the database. | Successfully data is inserted into the database. | Pass |



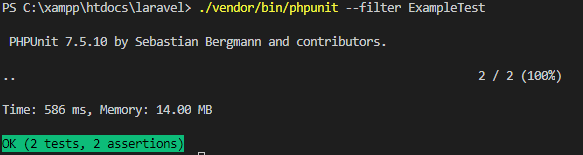
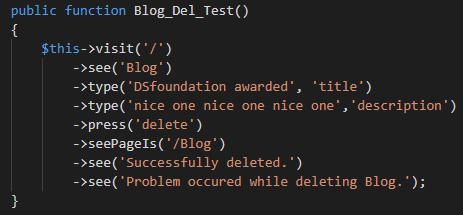


Figure 38: Picture of unit testing of Blog–add.

Test Name: Blog-Del Test.

Pre-condition: Admin must successfully delete the blog.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 6 | To check if the selected value is deleting properly from database. | Title: DSfoundation awarded.  Description: nice one….. | The data should be successfully deleted from the database. | Successfully data is deleted from the database. | Pass |



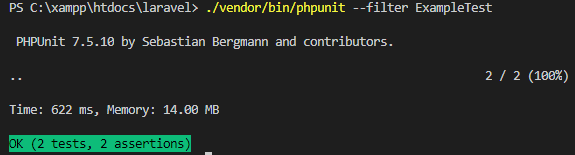


Figure 39: Picture of unit testing of deleting blog.

## **Black box testing.**

The process of testing the input and output without bothering internal working of the system is known as black box testing. The reason behind choosing black box testing are:

* It is easy to do because code is not required and done through the user points of view.
* It emphasis on functional requirements.
* Mainly, it test behavioral and performance errors.

Test Name: Donation Test.

Pre-condition: User must successfully donate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 1 | The value inserted for login should work properly. | Amount: 75000  Name: Abiraj Timalsina  Email:  abirajtimalsina87@yahoo.com  Phone: 9840779554  Address: Kapan  Field: Bara Relief. | The data should be inserted into the database and user should be redirected to the donation page. | Successfully data is inserted into the database and user is redirected to the donation page. | Pass |

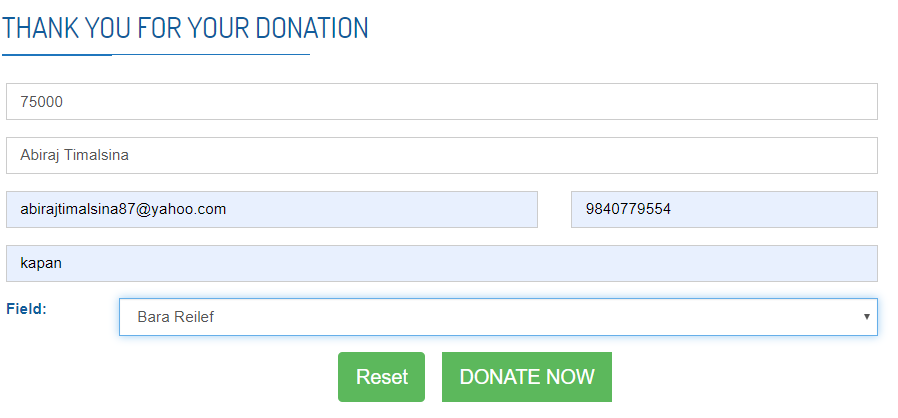


Figure 40: UI of donation page before donation.

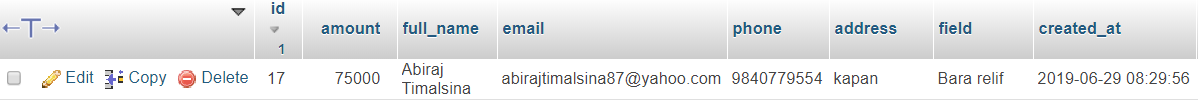


Figure 41: picture of database after successfully donated

Test Name: Negative amount validation Test.

Pre-condition: validation must restrict user from donating negative amount.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 2 | To check if the from validation is working properly from being inserting negative value. | Amount: -15000  Name: abiraj  Email:  abirajtimalsina87@yahoo.com  Phone: 9840779554  Address: Kapan  Field: Gautam Buddha international cricket stadium. | The data should not be inserted into the database and user should be redirected to the donation page with error message. | The Data did not inserted into the database and user is redirected to the donation page with error message. | Pass |



Figure 42: UI of donation page with validation.

Test Name: Contact Form Test.

Pre-condition: User must successfully send message with user details to contact.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 3 | To check if the value inserted is working properly. | Name: Sandesh Thapa  Email: th\_San756@gmail.com  Message: This site is well developed, responsive too and I can get every information that I want about this foundation but if there was a video than it'd be better. | The data should be inserted into the database and user should be redirected to the Contact page. | Successfully data is inserted into the database and user is redirected to the Contact page. | Pass |

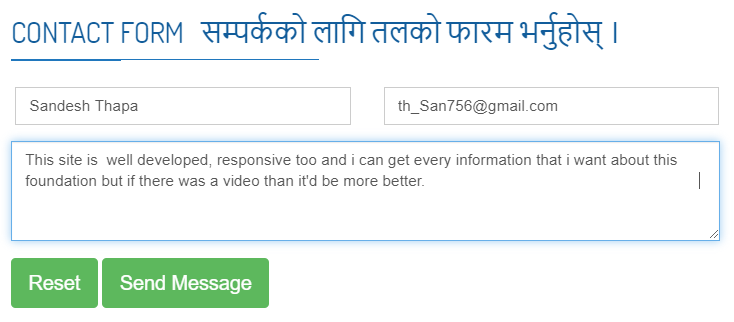


Figure 43: UI of Contact page before insertion of data.

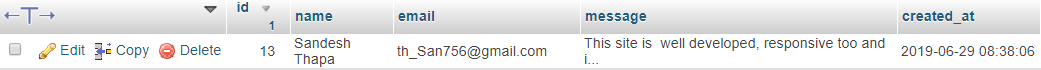
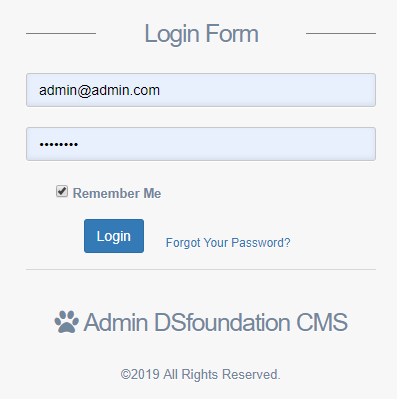


Figure 44: picture of database after successfully insertion of data from contact form.

Test Name: Login Test.

Pre-condition: Admin must successfully logged in and access to the dashboard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 4 | To check if the value inserted for login is working properly. | Email: admin@admin.com  Password: admin123. | The inserted data for login should be matched from the database and admin should be able to access the dashboard page. | Successfully inserted data is matched form the database and admin is able to access the dashboard page. | Pass |



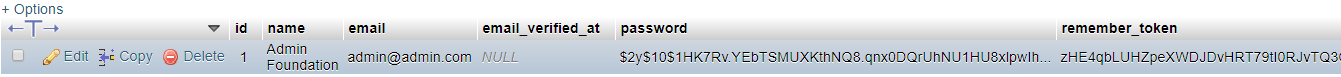


Figure 45: pictures of login form page with database picture before successfully admin logged in.

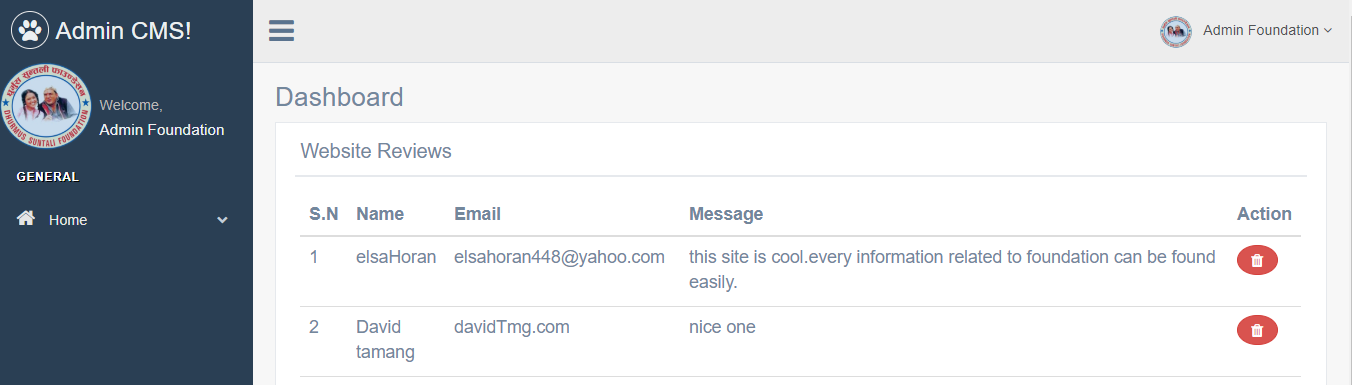


Figure 46: UI of dashboard pages after successfully admin logged in.

Test Name: Posting Blog Test.

Pre-condition: Admin must successfully post the blog and redirect to the post blog page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 5 | To check if the value inserted for login is working properly. | Title: राष्ट्र निर्माण, तपाई हाम्रो अभियान ।  Description: जनस्तरबाट……  Image: cricket.jpg | The data should be inserted into the database and admin should be redirected to the Blog add page. | Successfully data is inserted into the database and admin is redirected to the Blog add page. | Pass |

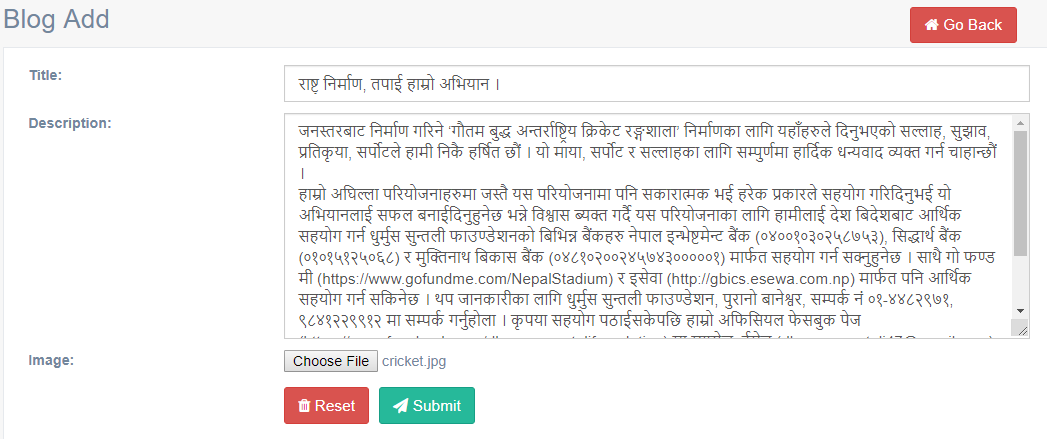


Figure 47: UI of Blog page before admin posting blog.

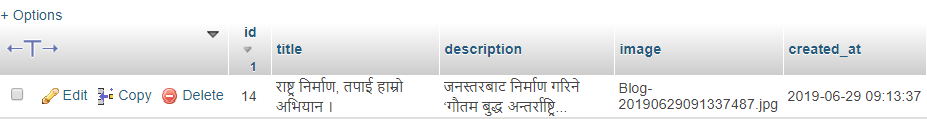


Figure 48: picture database page after admin successfully posted blog and data inserted into database.

Test Name: Posting Image Test.

Pre-condition: Admin must successfully post the image and redirect to the post image page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 6 | To check if the value inserted for login is working properly. | Image: 15.jpg | The data should be inserted into the database and admin should be redirected to the image add page. | Successfully data is inserted into the database and admin is redirected to the image add page. | Pass |

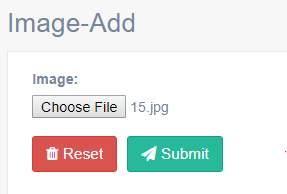


Figure 49: UI of Image page before posting image.



Figure 50: picture of database pages after admin successfully posted image and data inserted into database.

Test Name: Update blog Test.

Pre-condition: Admin must successfully update blog and redirect to the blog-add page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 7 | To check if the selected value for updating and inserting into the database is working properly. | Title: कोरियाको एक समुहले धुर्मुस…..  Description: रोजगारिको...  Image: DS.jpg | The data should be inserted into the database and admin should be redirected to the blog-update page. | Successfully data is inserted into the database and admin is redirected to the blog-update page. | Pass |



Figure 51: UI of blog-update page before updating blog.

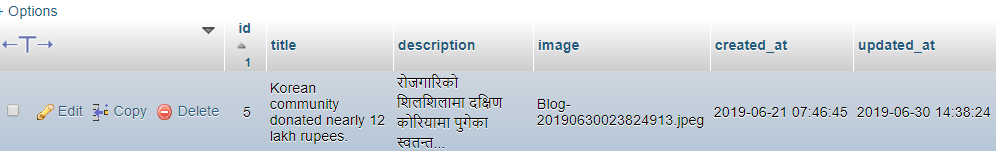


Figure 52: picture of database after admin successfully updated blog.



Figure 53: UI of blog-list after admin successfully updated blog.

Test Name: Delete image Test.

Pre-condition: Admin must successfully delete image and redirect to the image-list page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 8 | To check if the selected value for deleting from the image list and from database is working properly. | Image: DS.jpg | The image should be deleted from the database and image list too and redirected to the image-list page. | Successfully data is deleted from the database and image list page and admin is redirected to the image-list page. | Pass |

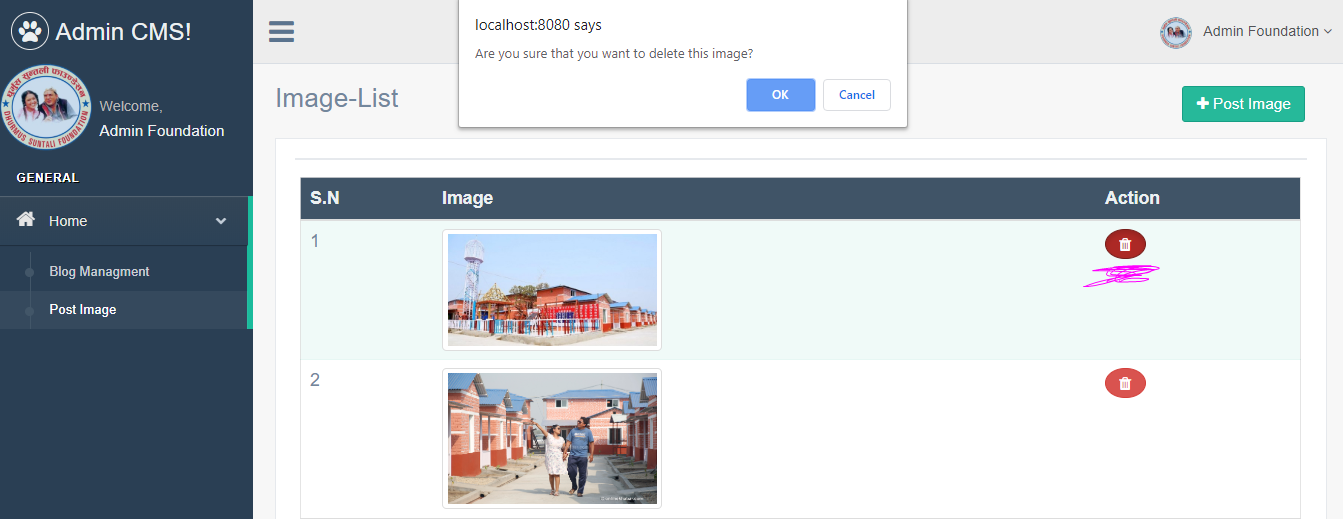


Figure 54: UI of image-list page before deleting image.

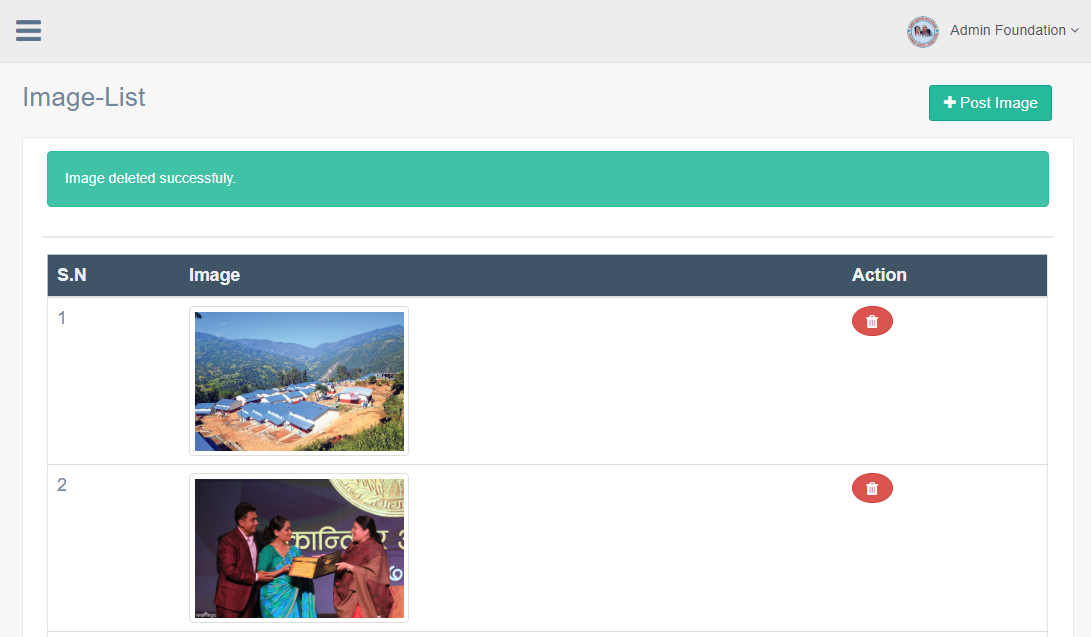


Figure 55: UI of image-list page after admin successfully deleted image.

Test Name: Delete blog Test.

Pre-condition: Admin must successfully delete blog and redirect to the blog-list page.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Test Scenario | Test Input Data | Expected result | Actual result | Result (Pass/Fail) |
| 9 | To check if the selected value for deleting from the blog list and from database is working properly. | Delete S.N 1 blog from blog-list. | The blog should be deleted from the database and blog list too and redirected to the blog-list page. | Successfully blog is deleted from the database and blog list page and admin is redirected to the blog-list page. | Pass |

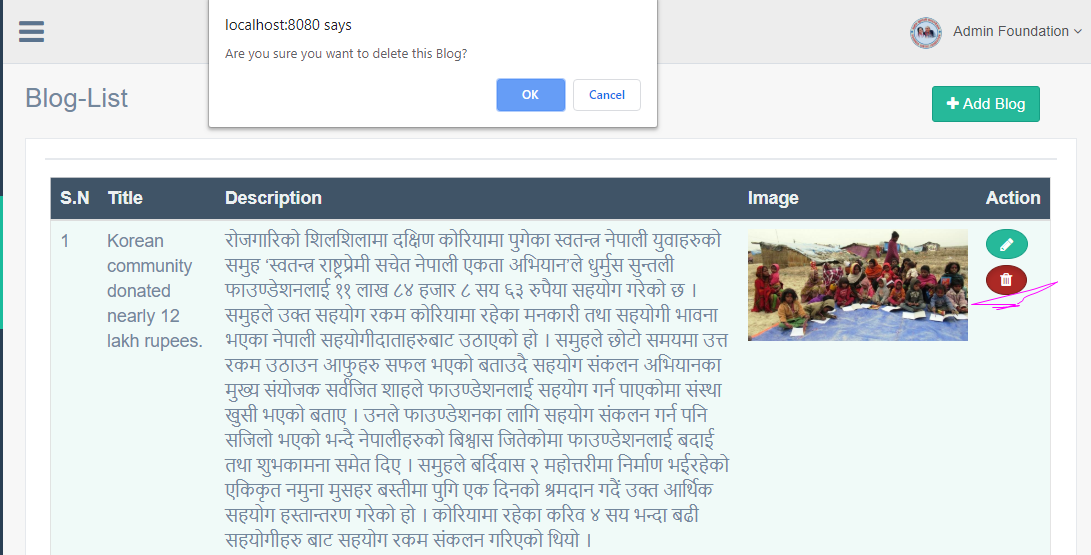


Figure 56: UI of blog-list page before deleting blog.

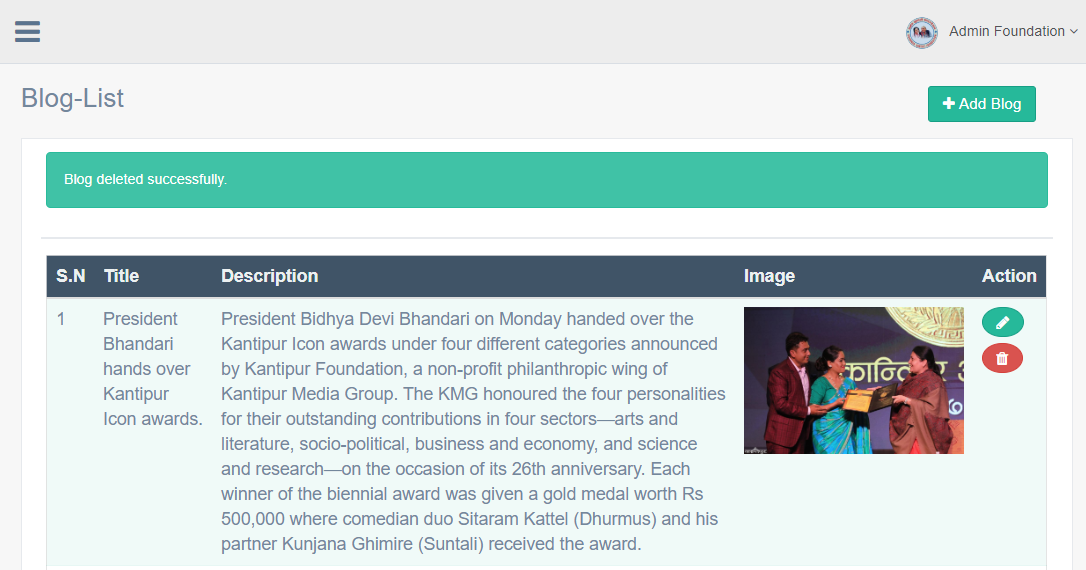


Figure 57: UI of blog-list page after deleting blog.

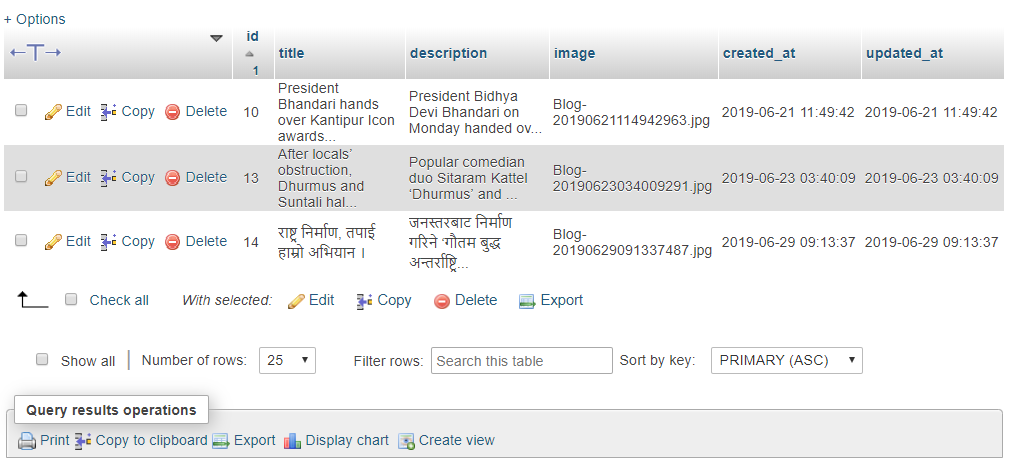


Figure 58: picture of dashboard after deleting blog.

# Chapter- 6

# Others Project Issue.

## Risk management

## Introduction

Risk management the process of recognizing, considering and monitoring intimidations to an administration's assets and incomes. And the below figure shows method to control risk management.



Figure 59: Risk management.

**How did I manage risk for my system?**

* I have identified the risk that might prevent from achieving user goals like: database attack, risk in structure, problems in links and so on.
* And also assess risk that all those risk that I have mentioned above are dangerous.
* So for those risk I have plan to control it such as, I will use the hashing function to encrypt the admin password in database and also making the website responsive and user friendly for user satisfaction and finally checking the links properly before deploying the website to the owner are planning and controlling methods for risk that I have applied.
* Continuing those plans and controlling as review for risk management in future also.

The risk likelihood with its values of this projects are shown below**.**

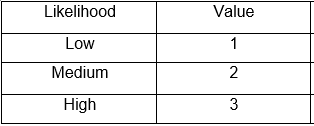


Figure 60: Risk likelihood.

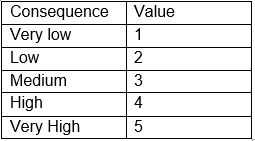


Figure 61: Risk consequences

The table of risk consequences are shown below.



Figure 62: Table of risk consequences

## Configuration management

## Introduction

It is the process of monitoring the modifications and tracing in software in relations of necessities, design, utilities and improvement of the product.

My Link: <https://github.com/AbirajTimalsina/cp>

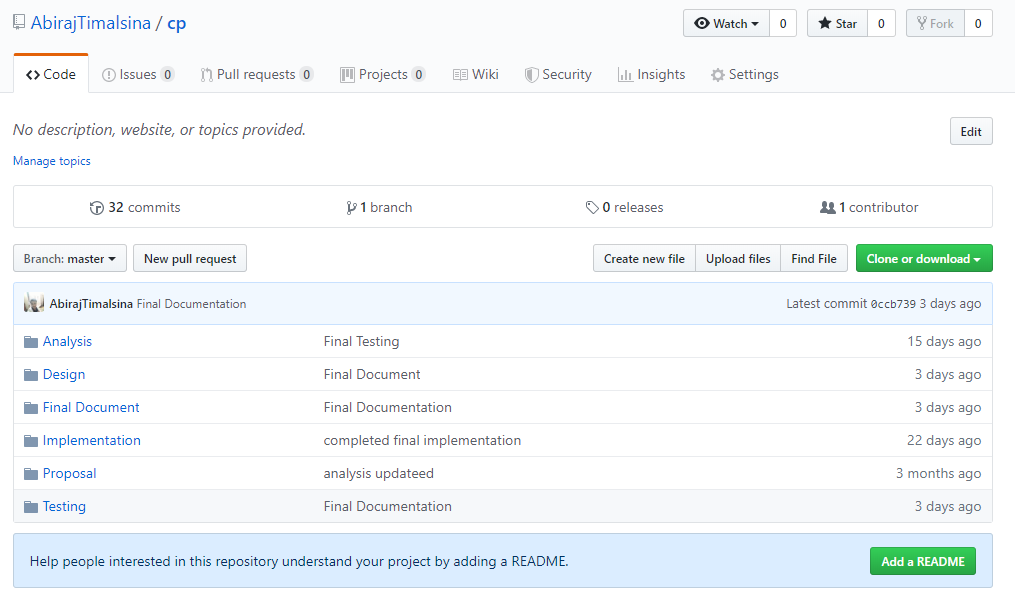


Figure 63: All documents uploaded on GitHub.

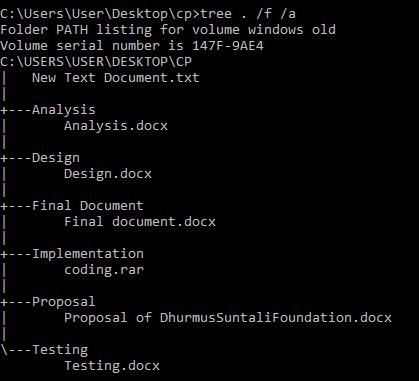


Figure 64: Tree structure of Dhurmus Suntali foundation on cmd.

## Scheduling

### Time Estimation table

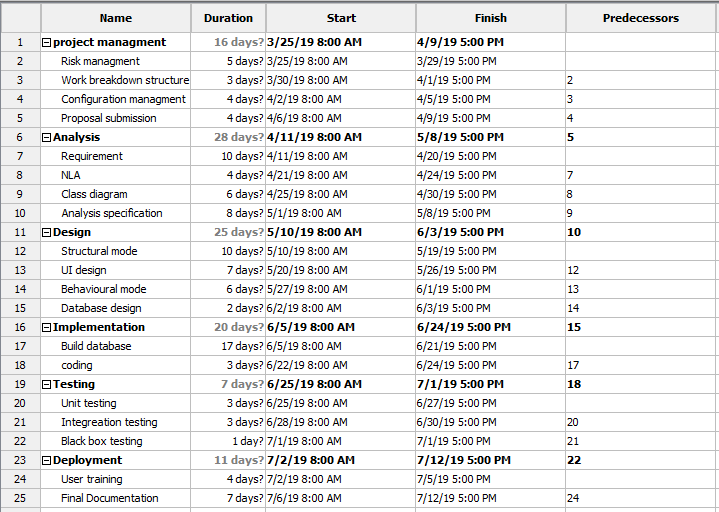


Figure 65: Time estimation table for task.

### Gantt chart

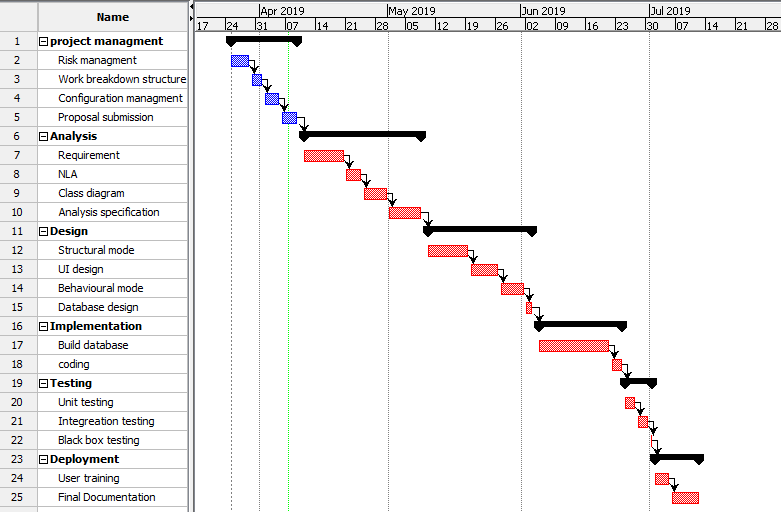


Figure 66: Gantt chart of Dhurmus Suntali foundation.

### Future Work

All most all requirements meet according to needed but there are still some other requirements which are set as future work and are explained below:

* Such as language translator which will help user to read the content of the website.
* For international donators PayPal is not available to donate so now they have to transfer their money directly in the foundations bank account.
* Admin cannot post videos. So it is also set as the future work.

Finally, these all things has been set to the future work of the Dhurmus Suntali foundations website.

### Project limitation

* User cannot directly make donation through the website. They have to donate through the e-sewa.
* No search box in donators details. So, user cannot search name in donator’s details page.
* Admin cannot post videos. So, videos is not available.
* PayPal is not available for international donators.

### User manual.

**To donate**

* To donate first you should open the website.
* Click on the donate button after that it will redirect you to the donation page will open.
* Then fill the every field in the form with amount as your wish because every field is required.
* Then click on the submit button and then see your details in donators details page.
* Or you can donate through e-sewa also for that click on the link which is given in help page.

**To send feedbacks**

* To send feedback first click on contact page in navigation bar.
* Then it will redirect you to the contact page.
* After that fill all the fields which is given in the form then click on the submit button.

# Chapter 7- Conclusion

Dhurmus Suntali foundation in of the one of the most wanted and popular social foundation website which has features of user friendly, responsive for any kind of devices where user can see blogs, images and admin can post news and blogs, but the only one limitation Is user cannot donate directly through the foundation website for that the user have to donate their amount from bank to the foundation’s bank account where the bank details are provided in the home page, not only this user can also see the progress of the foundation through the pictures from the gallery and can see the list of donators from the donators page. They not only can know the blogs but also can know the current problems happening in the country and then the news for help they needed.

I have used PHP programming language with HTML, JavaScript and Css for coding in visual studio code editor to develop this Likewise, to handle the website database I have used PHP Myadmin database because it is easy to use and popular too and JavaScript for the animation.

At last, instead of writing code in core PHP to implement this project I have chosen the laravel framework to do it because developers has made laravel for both beginners and for professionals. Also in the Model view and controller (MVC) pattern so that I won’t get confuse and feel easier to complete this project.

While doing coding in MVC pattern in core PHP It takes time and the time given to complete this project is also not enough so I chose laravel framework, where MVC pattern is already inbuilt inside it.

# Appendix



Figure 67: Model of donation page.

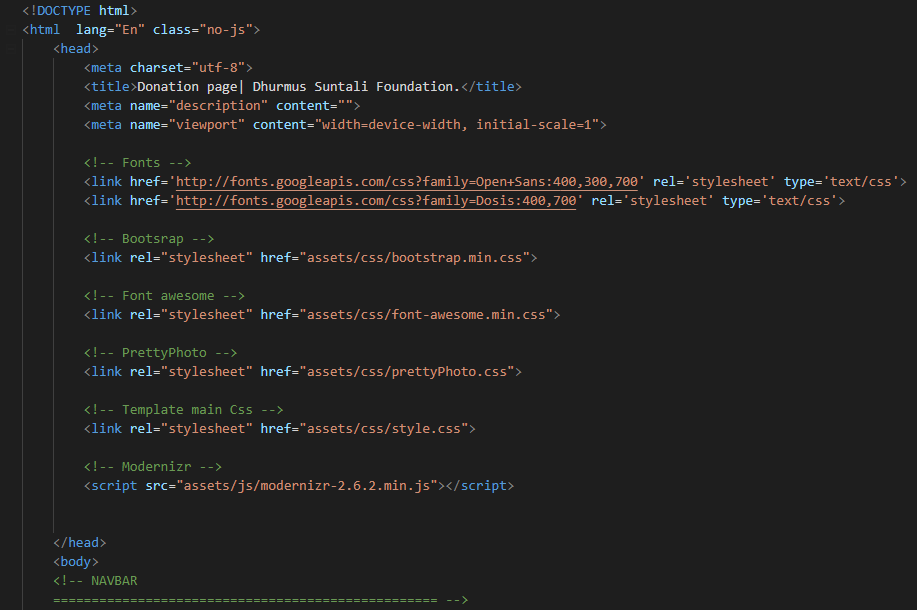


Figure 68: view of donation page.

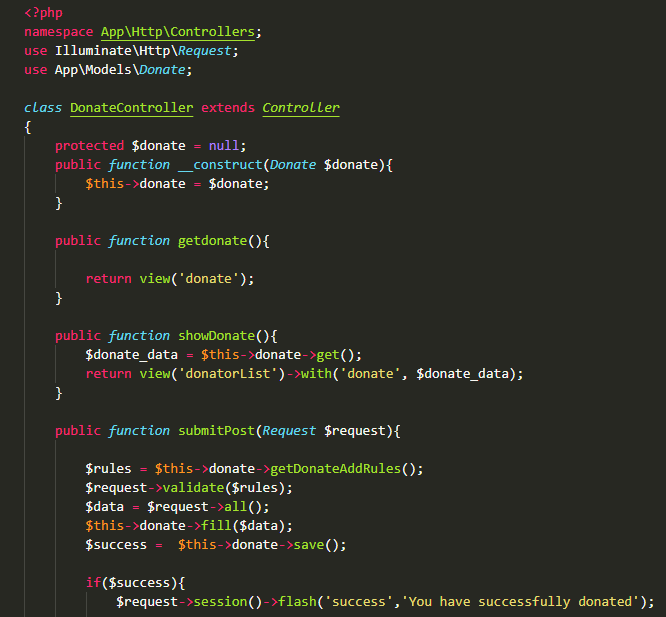


Figure 69: Controller of donation page.



Figure 70: model of image page.



Figure 71: view page of insert, delete of image page.

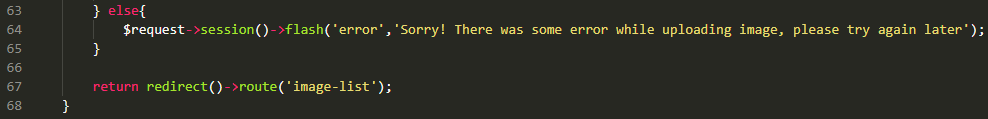
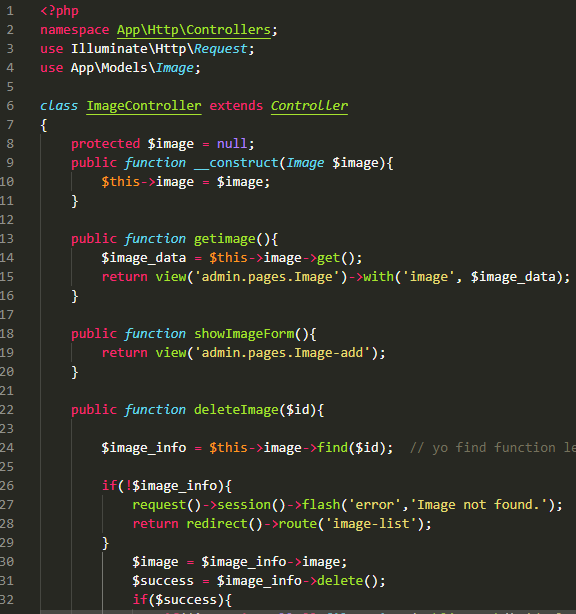


Figure 72: Controller page of image.



Figure 73: Model of contact page.



Figure 74: View page of insert, delete of contact.

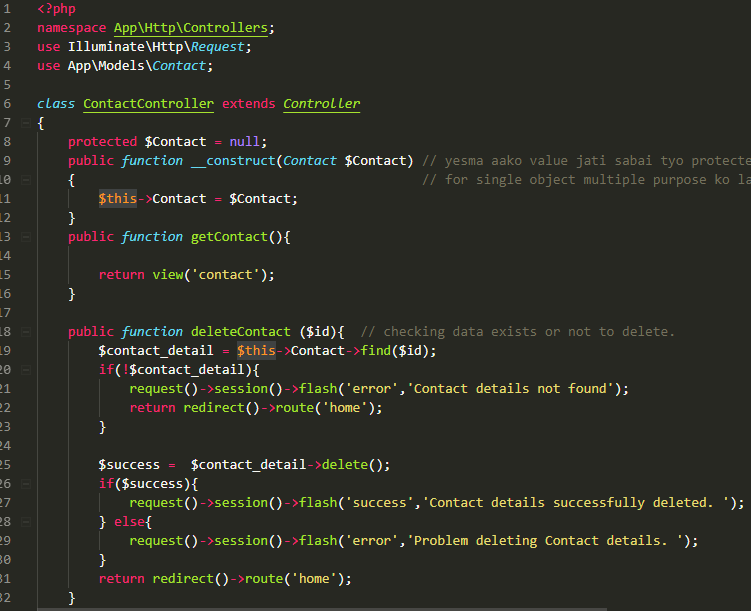


Figure 75: Controller of contact page.

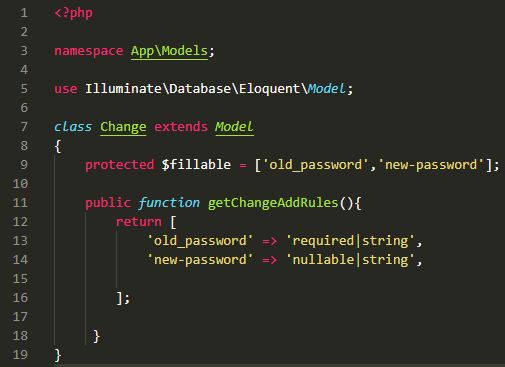


Figure 76: Model of Password Change page.



Figure 77: view page of Password change page.

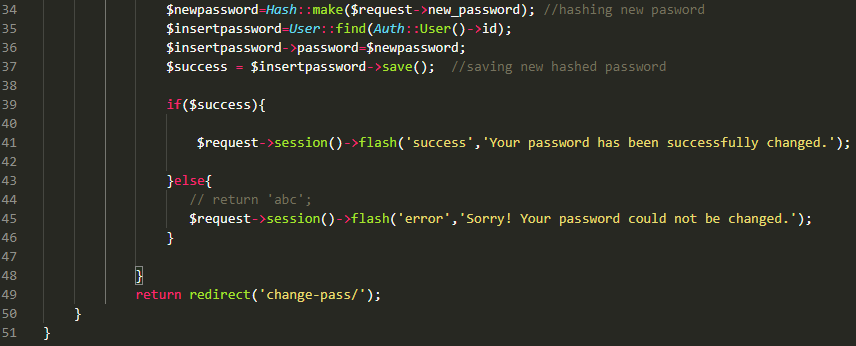


Figure 78: Controller page of password change.



Figure 79: Middleware of authentication.

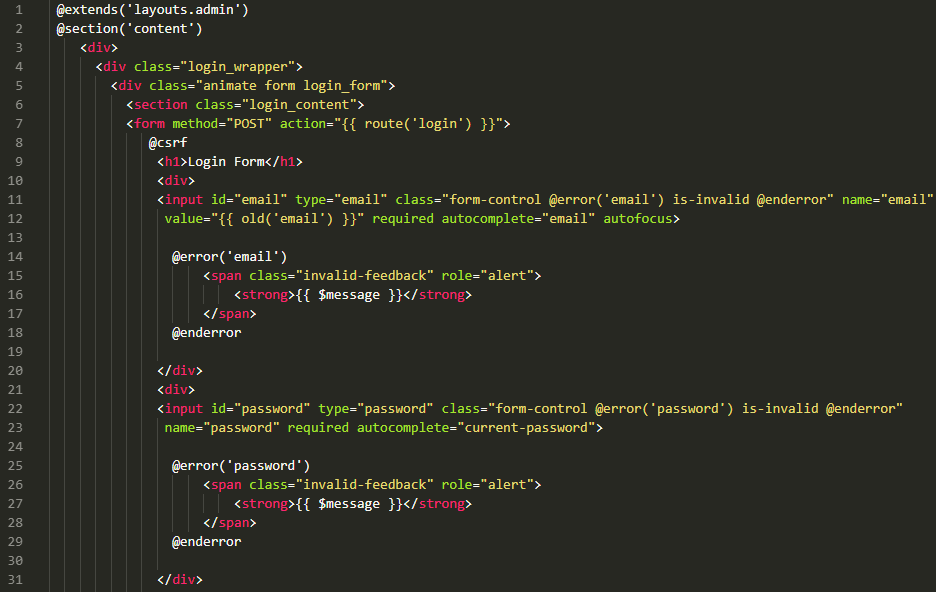


Figure 80: view page of authentication.



Figure 81: Model page of Blog.



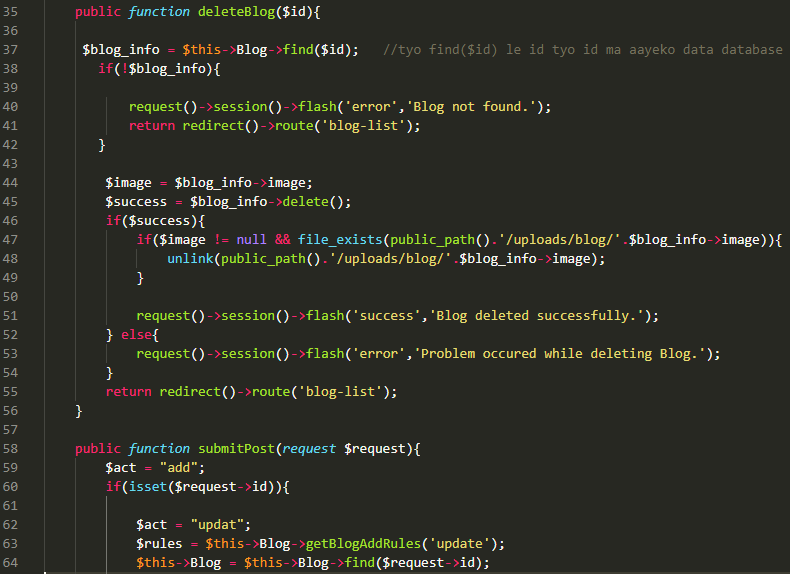
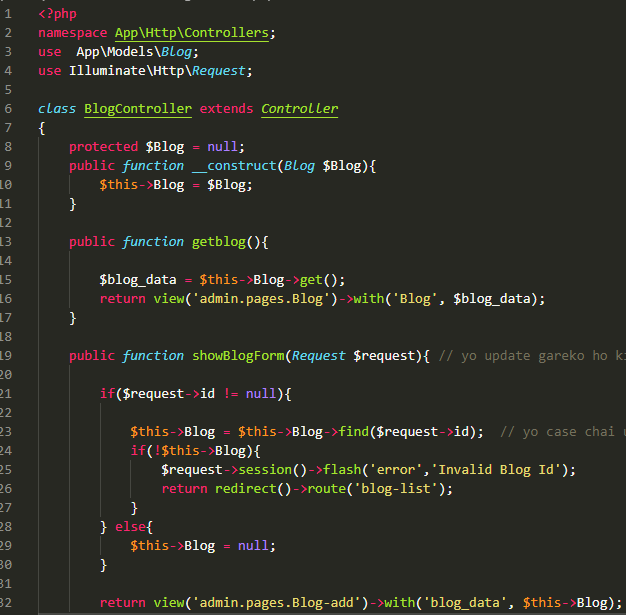
Figure 82: view page of blog.



Figure 83: View of Blog listing page.



Figure 84: View page of blog add, update, delete page.



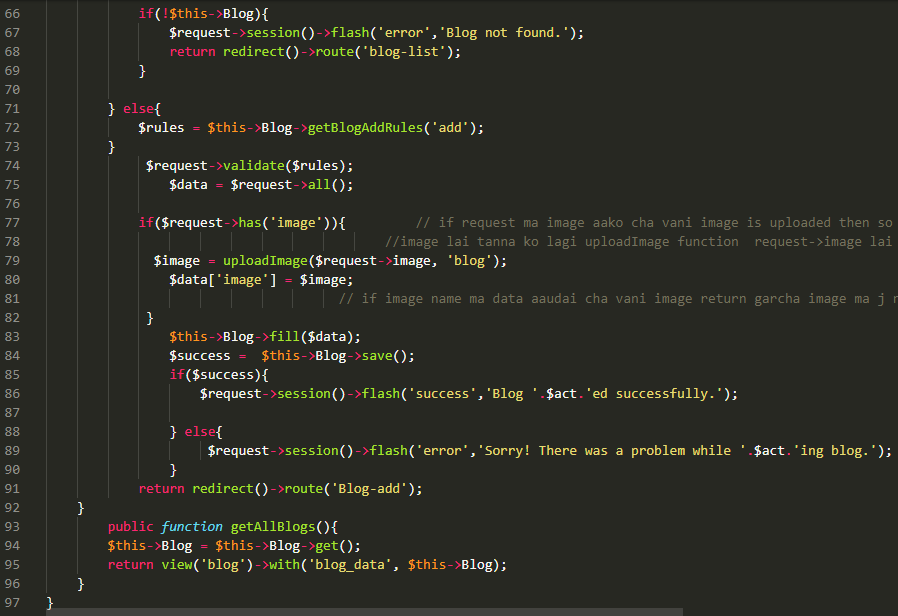


Figure 85: Controller page of Blog.