HETAUDA CITY COLLEGE

**Tribhuvan University**

**Institute of Science and Technology**

****

**Quizy**

**A PROJECT REPORT**

**Submitted To**

**Department of Computer Science and Information Technology**

**Hetauda City College**

**Under the Supervision of**

**Sulav Poudel**

**In Partial fulfillment of the requirements for the Bachelor’s Degree in Computer Science and Information Technology**

**Submitted by**

Sanjay Khatri

Jenesh Bhagat Pradhanang

Pawan Waiba

**September, 2019**

**Hetauda, Nepal**

HETAUDA CITY COLLEGE

**Tribhuvan University**

**Institute of Science and Technology**

Date: 2076/09/08,

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by Sanjay Khatri, Jenesh Bhagat Pradhanang and Pawan waiba entitled “**Quizy**” in partial fullfillment of the requirements for the degree of B.Sc in Computer Science and Information Technology be processed for the evaluation.

……………………………...

Sulav Paudel

Hetauda City College

Chisapani, Heatauda-4

HETAUDA CITY COLLEGE

**Tribhuvan University**

**Institute of Science and Technology**

Date: 2076/09/08

**Letter of Approval**

This is to certify that this project prepared by Sishir Ghimire and Shambhav Lama entitled “**Quizy**” in partial fulfillment of the requirements for the degree of BSc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

………………………

Mr. Saroj Poudel

(Program Coordinator)

Hetauda City College

………………………

Mr. Sulav Paudel

(Supervisor)

Hetauda City College

…………………………

Mr. Somebody

(External Examiner)

# Abstract

The **Quizy** is a web application for to take online quiz in an efficient manner and no time wasting for checking the paper. The main objective of **Quizy** is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. It also helps to prepare the quiz. And main thing is it take mind in the flow of the questions by increasing the curiosity. Mainly, this web application is suitable for school level students to make them intelligent in the lots of various type of questions.

# Acknowledgement

The success of this project "**Quizy**" would not have been possible without the kind support and assistant of many individuals. We are immensely blessed to get this all along the duration of our project. We would like to extend our profound gratitude to each and every one of them.

We are highly thankful to **Hetauda City College** for providing us this opportunity, as well as guidance, supervision and friendly environment for the successful completion of the project. We also admire the effort of BSc.CSIT coordinator **Mr. Saroj Poudel**, without his supporting role, the project would have been nowhere near to completion.

We would like to express our appreciation to our project supervisor **Sulav Paudel** who took interest in our project and guided us throughout the project by providing all necessary ideas, information and advice for the development of the functional web application.

Our thanks and appreciation also go to all the teaching staff of Hetauda City College for their constant support and encouragement which helped us successfully complete our project. We are fortune enough to get support and encouragement from each and every one of our colleagues in developing the project. We are also grateful to our family members who provide us constant support along the duration of this project.

Last but not least, we would like to thank each team member whose active involvement, hard work and collaborative nature made this project come out of the idea to the real world.

**Table of Contents**

[Abstract i](#_Toc62019799)

[Acknowledgement ii](#_Toc62019800)

[List of Abbreviation vi](#_Toc62019801)

[List of Figures vii](#_Toc62019802)

[List of Tables viii](#_Toc62019803)

[1. Introduction 1](#_Toc62019804)

[1.1. Relevance 1](#_Toc62019805)

[1.2. Problem Statement 1](#_Toc62019806)

[1.3. Objectives 1](#_Toc62019807)

[1.4. Scope and Limitation 1](#_Toc62019808)

[1.4.1. Scope 1](#_Toc62019809)

[1.4.2. Limitation 2](#_Toc62019810)

[2. Basic Concepts and Tools 2](#_Toc62019811)

[3. Requirement Analysis 3](#_Toc62019812)

[3.1. Study of Existing System 3](#_Toc62019813)

[3.2. Data Collection Methods 3](#_Toc62019814)

[3.2.1. Questionnaires 3](#_Toc62019815)

[3.2.2. Interaction 4](#_Toc62019816)

[3.2.3. Internet Research and observation 4](#_Toc62019817)

[3.3. Requirement Specifications 4](#_Toc62019818)

[3.3.1. Functional Requirements 4](#_Toc62019819)

[3.3.2. Non-Functional Requirements 5](#_Toc62019820)

[3.4. Server/Client Requirement 5](#_Toc62019821)

[3.4.1. Server Requirements 6](#_Toc62019822)

[3.4.2. Client Requirements 6](#_Toc62019823)

[3.5. Feasibility Analysis 7](#_Toc62019824)

[3.5.1. Technical Feasibility 7](#_Toc62019825)

[3.5.2. Economic Feasibility 7](#_Toc62019826)

[3.5.3. Operational Feasibility 8](#_Toc62019827)

[3.5.4. Schedule Feasibility 8](#_Toc62019828)

[4. System Design 9](#_Toc62019829)

[4.1. System Flowchart 9](#_Toc62019830)

[4.2. Use Case Diagram 10](#_Toc62019831)

[4.3. Data Flow Diagram 11](#_Toc62019832)

[4.3.1. Level 0 Data Flow Diagram 12](#_Toc62019833)

[4.3.2. Level 1 Data Flow Diagram 13](#_Toc62019834)

[4.4. Entity Relationship Schema 15](#_Toc62019835)

[4.5. Data Dictionary 15](#_Toc62019836)

[4.5.1. admins 16](#_Toc62019837)

[4.5.2. migrations 17](#_Toc62019838)

[4.5.3. password\_resets 17](#_Toc62019839)

[4.5.4. tbl\_products\_details 17](#_Toc62019840)

[4.5.5. tbl\_suppliers\_details 18](#_Toc62019841)

[4.5.6. tbl\_users\_details 19](#_Toc62019842)

[4.5.7. Gantt Chart 20](#_Toc62019843)

[5. System Development 22](#_Toc62019844)

[5.1. Software Development Model 22](#_Toc62019845)

[5.1.1. Agile Development Model 22](#_Toc62019846)

[5.2. Front End Tools 24](#_Toc62019847)

[5.2.1. HTML5/CSS3 24](#_Toc62019848)

[5.2.2. BootsrapVue 24](#_Toc62019849)

[5.2.3. Vue 24](#_Toc62019850)

[5.3. Back End Tools 25](#_Toc62019851)

[5.3.1. Laravel Framework 25](#_Toc62019852)

[5.3.2. MySQL 28](#_Toc62019853)

[5.3.3. Pusher 29](#_Toc62019854)

[5.3.4. Apache Server 2.4.35 29](#_Toc62019855)

[5.4. Others Tools 29](#_Toc62019856)

[5.4.1. PHPStorm JetBrains 29](#_Toc62019857)

[5.4.2. Windows Visio 2016 29](#_Toc62019858)

[5.5. Graphical User Interface 30](#_Toc62019859)

[5.5.1. Home Page 30](#_Toc62019860)

[5.5.2. User Login Page 30](#_Toc62019861)

[5.5.3. Register Page 30](#_Toc62019862)

[5.5.4. Admin Page 30](#_Toc62019863)

[5.5.5. Host Page 30](#_Toc62019864)

[5.5.6. Game Page 30](#_Toc62019865)

[6. Testing 31](#_Toc62019866)

[6.1. Unit Testing 31](#_Toc62019867)

[6.1.1. Test case for user login 31](#_Toc62019868)

[6.1.2. Test case for admin login 31](#_Toc62019869)

[6.1.3. Test case for user registration 32](#_Toc62019870)

[6.2. Integration Testing 33](#_Toc62019871)

[6.3. System Testing 33](#_Toc62019872)

[6.3.1. Black Box Testing 33](#_Toc62019873)

[6.3.2. White Box Testing 33](#_Toc62019874)

[7. Implementation and Result Analysis 34](#_Toc62019875)

[8. Conclusion and Future Enhancement 36](#_Toc62019876)

[9. Bibliography 38](#_Toc62019877)

[Appendix 39](#_Toc62019878)

# List of Abbreviation

AJAX Asynchronous JavaScript and XML

API Application Programming Interface

COCOMO Constructive Cost Model

CSS Cascading Style Sheets

CRUD Create Read Update and Delete

DFD Data Flow Diagram

ERD Entity Relationship Diagram

GPS Global Positioning System

HTML Hypertext Markup Language

IDE Integrated Development Environment

JSON JavaScript Object Notation

MIS Management Information System

MIT Massachusetts Institute of Technology

MVC Model View Controller

NTA Nepal Telecommunications Authority

PDO PHP Data Objects

PHP PHP Hypertext Preprocessor

RDBMS Relational Database Management System

SQL Structured Query Language

UML Unified Modeling Language

XML Extensible Markup Language

XP Extreme Programming

# List of Figures

[Figure 4.1‑a: System flow chart 10](#_Toc16607141)

[Figure 4.2‑a: Use Case Diagram for main feature 10](#_Toc16607142)

[Figure 4.3‑a: Level 0 Data Flow Diagram 11](#_Toc16607143)

[Figure 4.3‑b: Level 1 Data Flow Diagram for User 12](#_Toc16607144)

[Figure 4.3‑c: Level 1 Data Flow Diagram for Admin 13](#_Toc16607145)

[Figure 4.4‑a: Entity Relationship Schema 14](#_Toc16607146)

[Figure 5.1‑a: Agile Development 21](#_Toc16607147)

[Figure 5.3‑a: Model-View-Control 26](#_Toc16607148)

[Figure 5.3‑b: Laravel application structure 27](#_Toc16607149)

# List of Tables

[Table 3.4‑a: Minimum Hardware Requirements for Server 6](#_Toc16606537)

[Table 3.4‑b: Minimum Software Requirements for Server 6](#_Toc16606538)

[Table 3.4‑c: Minimum Hardware Requirements for Clients 6](#_Toc16606539)

[Table 3.4‑d: Minimum Software Requirements for Clients 7](#_Toc16606540)

[Table 3.5‑a: List of major technologies used in Quizy 7](#_Toc16606541)

[Table 4.3‑a: Data Flow Diagram Notation 10](#_Toc16606542)

[Table 4.5‑a: Description of database tables 15](#_Toc16606543)

[Table 4.5‑b: Details of admins table 15](#_Toc16606544)

[Table 4.5‑c: Details of migrations table 16](#_Toc16606545)

[Table 4.5‑d: Details of password\_resets table 16](#_Toc16606546)

[Table 4.5‑e: Details of tbl\_admin\_products\_suppliers\_details table 16](#_Toc16606547)

[Table 4.5‑f: Details of tbl\_suppliers\_details table 17](#_Toc16606548)

[Table 4.5‑g: Details of tbl\_users\_details table 18](#_Toc16606549)

[Table 5.3‑a: List of artisan command and description 24](#_Toc16606550)

[Table 6.1‑a: Test case for user login 30](#_Toc16606551)

[Table 6.1‑b: Test case for admin login 30](#_Toc16606552)

[Table 6.1‑c: Test case for user registration 31](#_Toc16606553)

[Table 6.1‑d: Test case for forget password 31](#_Toc16606554)

# Introduction

## Relevance

‘**Quizy**’ is a web application that has general questions related to current affairs and computer. It has multiple choice questions with time limit and it also calculate score of each correct answer. It is good for students of every age group it helps in increasing general knowledge about world, sport and computers etc. User need to register first then login into the ‘**Quizy**’. User can organize the quiz contest.

## Problem Statement

**‘Quizy’** is a web application developed to conduct or organize quiz based on time constraints. **‘Quizy’** system is accessed by entering the username and password which is added to the database. Only the host has access to credential, so they can create and manage quizzes. The player can join the quiz with the pin provided by the host. Quiz is started by the host by displaying on question with four options. The players then have to choose the options. If the answer is correct, score is incremented by one and no negative marks for wrong answer. Quiz system will finally direct to the score page and final score will be displayed along with the names of the player.

## Objectives

The main objective of **‘Quizy’** is to digitalize the system and facilitate a user-friendly environment for all users and reduces the manual effort. In the past days quiz is conducted manually but in further resolution of the technology we are able to generate the score and pose the queries automatically. The functional requirements include to create users that are going to participate in the quiz, automatic score and administrative tasks like add, delete, update for admin privilege users. In this application, all the permissions lie with the administrator i.e., specifying the details of the quiz, customizing the questions and hosting the quiz.

## Scope and Limitation

### Scope

The use of technology is increasing day by day, we all depends on technology, and we use various technologies to accomplish specific tasks in our lives. Todays, various type of games are played online and various organizations organize online games and **‘Quizy’** is one of those online game with lots of knowledge.

### Limitation

**Technology**, the one obvious limitation is that not every ESL classroom has a computer, let alone access to a computer lab with one computer for each student.

**Quality**, the astounding variety of materials available in the internet – not all of them are good quality. Some may have mistaken; others may not be challenging enough and in others there are distracting ads or banners with content that is inappropriate.

# Basic Concepts and Tools

**2.1 Introduction to Vue.js**

Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hands, Vue is also perfectly capable of powering sophisticated single- page application when used in combination with modern tooling and supporting libraries.

**2.2 Introduction to Laravel**

Laravel is a PHP based web framework for building high- end web applications using its significant and graceful syntaxes. It comes with a robust collection of tools and provides application architecture. Moreover, it includes various characteristics of technologies like ASP.NET MVC, CodeIgniter, Ruby on Rails, and many more. This framework is an open-source framework. It facilitates developers by saving huge time and helps reduce thinking and panning to develop the entire website from scratch. Along with that, the security of the application is also taken care of by Laravel. So, all its features can increase the speed of web development for you. If you are familiar with intermediate PHP scripting and PHP basics, Laravel can prepare your work more efficiently.

# Requirement Analysis

Before making any new system, we have to take consideration of different requirements. Since our project is based on the digitalization of online quiz contest. Hence, we use the following techniques to analysis the requirement.

## Study of Existing System

There is different web application of social organizations that are available in Nepal which helps to donate the funds for the users. Web sites like Manav sewa Nepal, Maita Nepal, etc. are some examples of such system.

When we observed closely, we found most of these web sites are not fully functional. They still do not have waste food and clothes donation and collection feature. Hence, we intent to create a system that will overcome this problem and provide a full digitized way of donating the waste going foods and clothes.

## Data Collection Methods

Data collection is the systematic approach of gathering and measuring information from a verity of source to get accurate and complete picture of an area of interest. We have done the data collection for our Quizy with various techniques.

### Questionnaires

A questionnaire is a very popular tool for data collection, where a number of respondents are asked to answer a set of question either written or verbal about a pertinent topic. Questionnaire is done by creating a set of questions for the targeted respondents specifying the information needed. To collect the information for Quizy we have prepared a set of questions for the respondent.

* Does the existing online donation system give you the assurance of your donation?
* Do you prefer donating from the internet using website? If yes, why? If no, why not?
* Do you think using online donation is convenient?

### Interaction

Interaction was done with various normal users which help us to gain vital information and views on the aspect of using online donating system over manual system. It helps us in gathering information, combined perspectives and opinion. Interaction allowed us to get directly through their ideas, demands and expectations that significantly enhance the design and development of our web application.

### Internet Research and observation

We search different websites as a part of our research. We observe and analysis how they offer services. This research helps us to understand about the existing system and features they couldn’t address.

## Requirement Specifications

Requirement specification can be broken down into functional requirements and non-functional requirements.

### Functional Requirements

Functional requirements define the function and components of the system. It also describes the technical details, accepting inputs, processing and what a system is going to achieve.

#### Login module

This module is provided by administrator for user to donate their things who have registered in the system.

* **Input-** User id and password.
* **Process-** After entering user id and password by user, administration checks whether the data provided by user is in the database or not.
* **Output-** Registered user can access the system and use the services.

#### Administrator module

Admin module also provide user id and password with which admin can access and control the system.

* **Input-** Login id and password.
* **Process-** Manage bus user’s data as well as view the collected items.
* **Output-** Administrator will maintain the database, customer’s details and will give the details of reserved system.

#### Surf module

Here, users can surf the web application for the donation.

* **Input-** Donate as per the wish.
* **Output-** Shows the donation history.

#### User module

This module is mainly for the users, where user can surf the web app and donate their things according to their wish. Here unregistered users can only donate funds.

* **Input-** User id and password
* **Process-** Validation process will occur
* **Output-** Only registered user can get the service provided by admin.

### Non-Functional Requirements

Non-functional requirements describe how the system works. Following are the non-functional requirements of our system:

#### Performance

This is the part where depends on the speed of internet. This is about time expectation. Mouse click is the priority for the users for donating their things.

#### Security

This is the important part of it. We give the high security to our user to protect their accounts from an unauthorized user or illegal hacking.

## Server/Client Requirement

Quizy system is based on client server architecture. Server/Client requirement helps us to outline the minimum software and hardware needed to deploy our project.

### Server Requirements

The values below refer to the minimum available hardware and software required to run Quizy. The Laravel framework has a few system requirements. (Server Requirements, 2017).

Table ‑: Minimum Hardware Requirements for Server

|  |  |
| --- | --- |
| Processor | 2.4 GHz or above |
| RAM | Greater than or equal to 2GB |
| Hard Disk | 80GB or above |
| Cache Memory | 512MB or above |

Table ‑: Minimum Software Requirements for Server

|  |  |
| --- | --- |
| Basic | PHP Version greater than or equal to 5.6.4, OpenSSL PHP Extension, PDO, PHP Extension, Mbstring PHP Extension, Tokenizer PHP Extension, XML PHP Extension |
| Back End | MySQL Server 2008 |
| Operating System | Linux or any compatible operating system |
| Web Server | Apache, Nginx or any compatible web server |

### Client Requirements

The values below refer to the minimum hardware and software requirements need to client.

Table ‑: Minimum Hardware Requirements for Clients

|  |  |
| --- | --- |
| Processor | 1.8GHz or any compatible processor |
| RAM | 256 or above |

Table ‑: Minimum Software Requirements for Clients

|  |  |
| --- | --- |
| Operating System | Any compatible operating system (Windows, Android, Linux, Blackberry etc.) |
| Browser | Internet Explorer 8 or any compatible browser |

## Feasibility Analysis

Before the system is developed, it must be investigated to find out whether the system is feasible or not. Feasibility study helps to decide if the project is worth the investment of time and money. It determines the potential success of the project. We performed following feasibility test for our project:

### Technical Feasibility

The main objective of a technical feasibility study is to determine whether a certain plan of action is feasible or not. A technical feasibility assessment should be applied to our project; it helps us to know whether the project can be done technically or not. The technical feasibility concerns with equipment and software that will successfully satisfy the requirements. Our system is being developed using following technologies:

Table ‑: List of major technologies used in Quizy

|  |  |
| --- | --- |
| Front-End | HTML 5, CSS3, BootstrapVue |
| Backend | Laravel, SQL |
| Client-Side Script | Vue JS |
| For data interchange | AJAX, JSON |

The project will be less risky because our group members are familiar with above technologies and the systems.

### Economic Feasibility

Economic feasibility is considering as the bottom line for most system. We have to choose the technology that easily fits in the budget. Basic Constructive Cost Model (Basic COCOMO) is used for cost estimation of our project.

Quizy is going to build using PHP language. We are trying to deal with developing well understood application program, the size of development team is small and the team member are unexperienced in developing similar type of projects.

The web application is available and accessible via the Internet which makes it economically feasible for user. Users only require an Internet connection to access contents included in the web application. Moreover, there are no charges of any sorts associated with contents in the web application.

|  |  |
| --- | --- |
| Webpage Name | Number of Lines of Code |
| Login | 145 LOC |
| Sign Up | 210 LOC |
| Verification | 146 LOC |
| Manage Product | 213 LOC |
| Fund Donation | 180 LOC |
| Create Donation Page | 190 LOC |
| Organization Details | 110 LOC |
| Forget Password | 50 LOC |
| Contact us | 40 LOC |
| Home Page | 130 LOC |
| Admin Dashboard | 145 LOC |
| Total Line of Code = 2127 | |

Total Kilo Line of Code (KLOC) = 2767/1000 = 2.127

Effort = 3.0 x (KLOC)1.12 PM = 3.0 x 2.1271.12 = 7.35 Person Months

Time to develop the software (Tdev) = ­ 2.5 x (Effort)0.35 Months = 2.5 x (7.35)0.35 = 4.9 months

Cost required to develop the software = 4.9 x 8000 = Rs.39200

Basic COCOMO shows that the project will be completed in 6 months at the budget of Rs.40000.

### Operational Feasibility

Operational feasibility is mainly related to the human involvement. The system solves the problem and fulfills the aspects of operational feasibility study.

### Schedule Feasibility

Evaluation of time is the most important consideration in the development of project. Schedule feasibility helps to identified how long will each activity take and how long will the entire project take? Project schedule split project into task and estimate time and resources required to complete each task. The main purpose of scheduling is to identify the task dependencies and is to avoid delays caused by one task waiting for another to complete. Typically, Gantt chart is used for monitoring and tracking the project progress.

# System Design

System Design describes the components, modules, system activities, and system architecture of Quizy. Various system design aspect has been followed and these are explained below:

## System Flowchart

Below flowchart shows the overall system flow of Quizy.

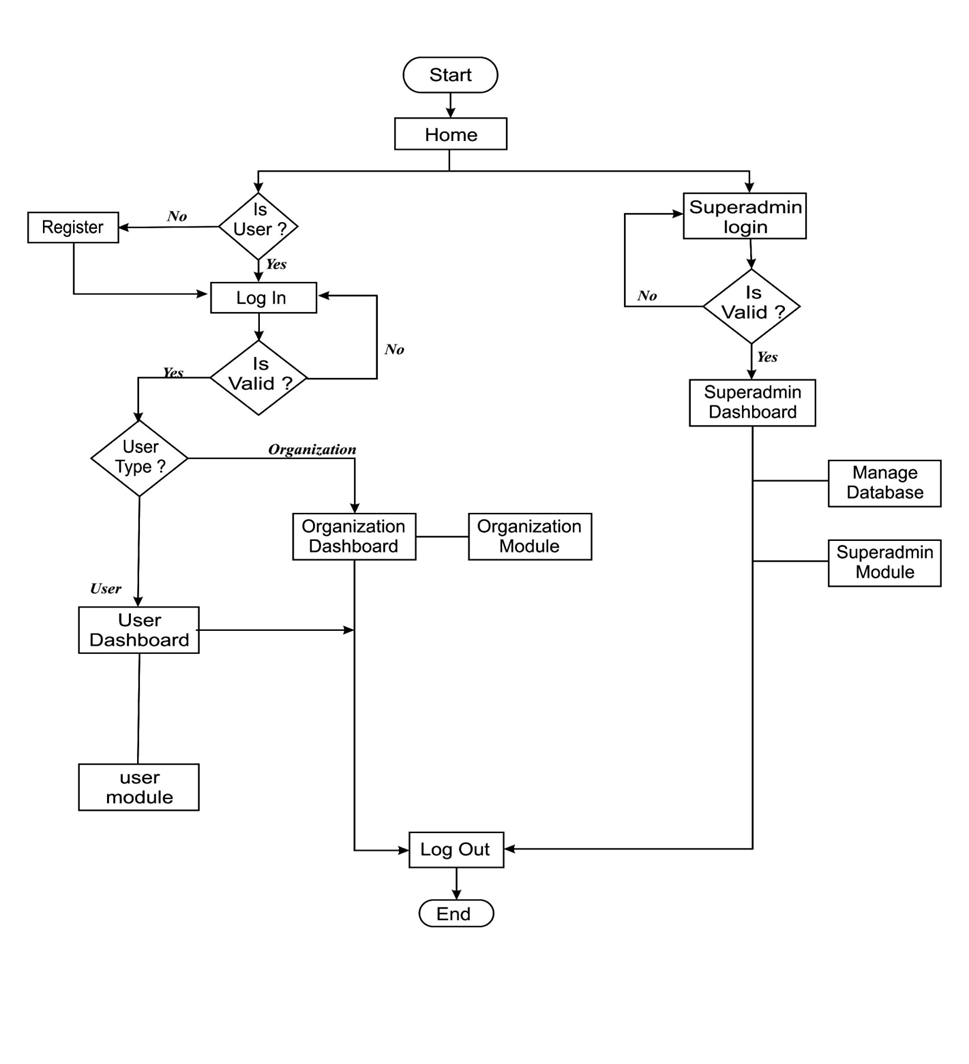


Figure ‑: System flow chart

## Use Case Diagram

The purpose of use case diagram is to capture the dynamic aspect of system. Use case diagram consists of use cases, actors, and their relationships. Use case diagram helps to get the outside view of our system, it also helps to identify the external and internal factors influencing the system. Following use case shows the dynamic aspect of proposed system.

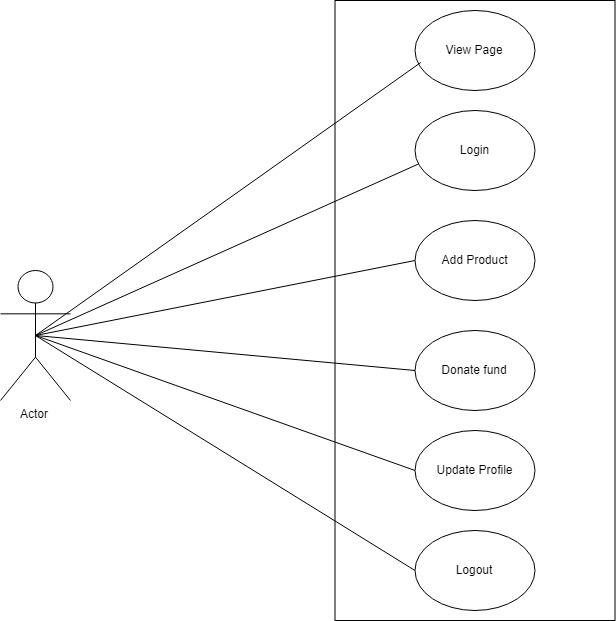


Figure ‑: Use Case Diagram for main feature

## Data Flow Diagram

Data Flow Diagram shows the flow of data in a system. Data Flow Diagram (DFD) consists of entities, processes, data store and data flow (Data Flow Diagram Notation, 2009).

Table ‑: Data Flow Diagram Notation

|  |  |  |
| --- | --- | --- |
| Notations | Represent | Description |
|  | Entity | Source and destination of information data |
|  | Data Process | Activities and action taken on the data |
|  | Data Store | Data storage |
|  | Data Flow | Data movement |

### Level 0 Data Flow Diagram

Level 0 DFD also known as context diagram. This diagram contains only one process node.

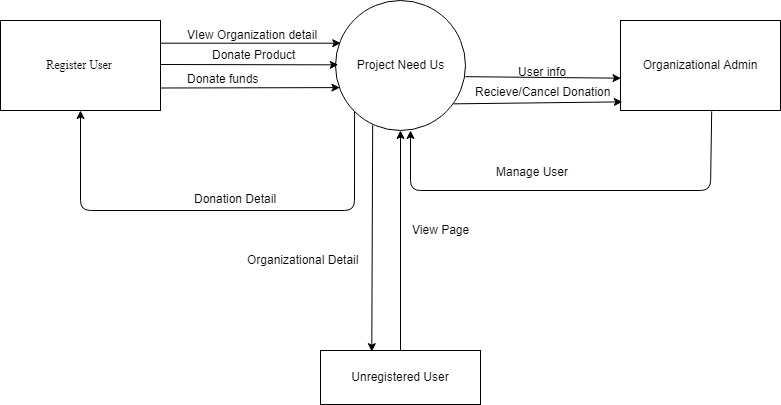


Figure ‑: Level 0 Data Flow Diagram

### Level 1 Data Flow Diagram

Level 1 Data flow diagram provides a more details than context level diagram. Level 1 DFD method is used to specify the functional requirement for the system design. It describes the business process, user perceived information, a workflow, and a decision rule for system.

The level 1DFD shows only the major high-level processes in the system interact. Each process on the level 0 DFD can be decomposed into more explicit DFD, called Level 1 diagram, which shows how it operate in getter details. For simplicity, we have decomposed our Quizy Data Flow Diagram into two parts; DFD for user and DFD for admin.

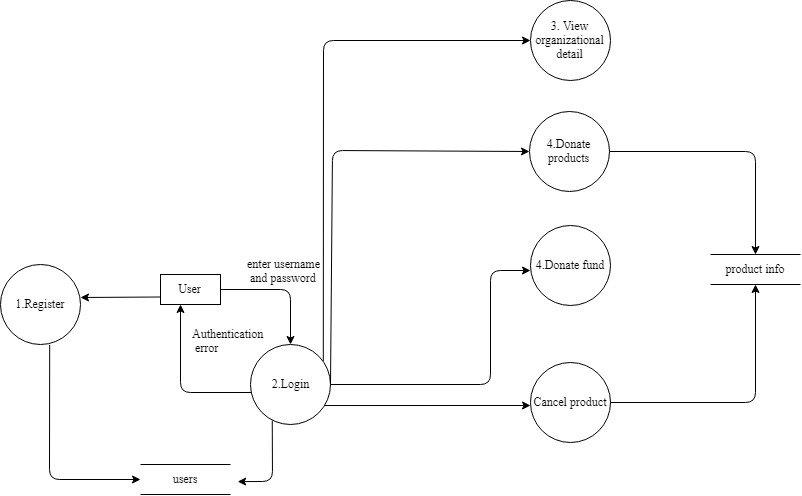


Figure ‑: Level 1 Data Flow Diagram for User

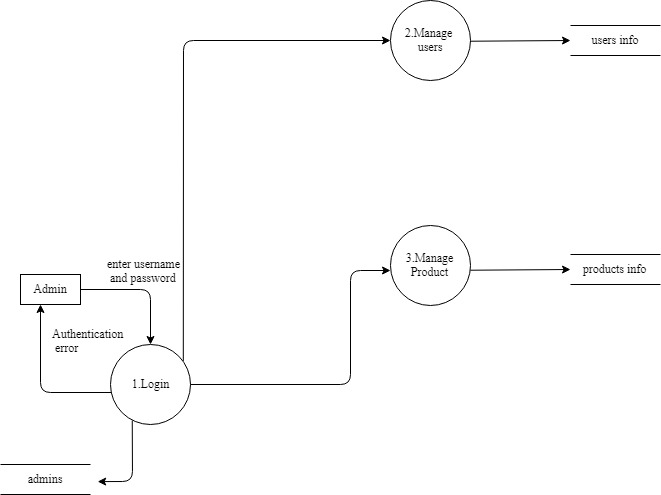


Figure ‑: Level 1 Data Flow Diagram for Admin

## Entity Relationship Schema

Entity Relationship Schema is tabular reorientation that describes how entities are related to each other.

ER Schema of Quizy is given below:

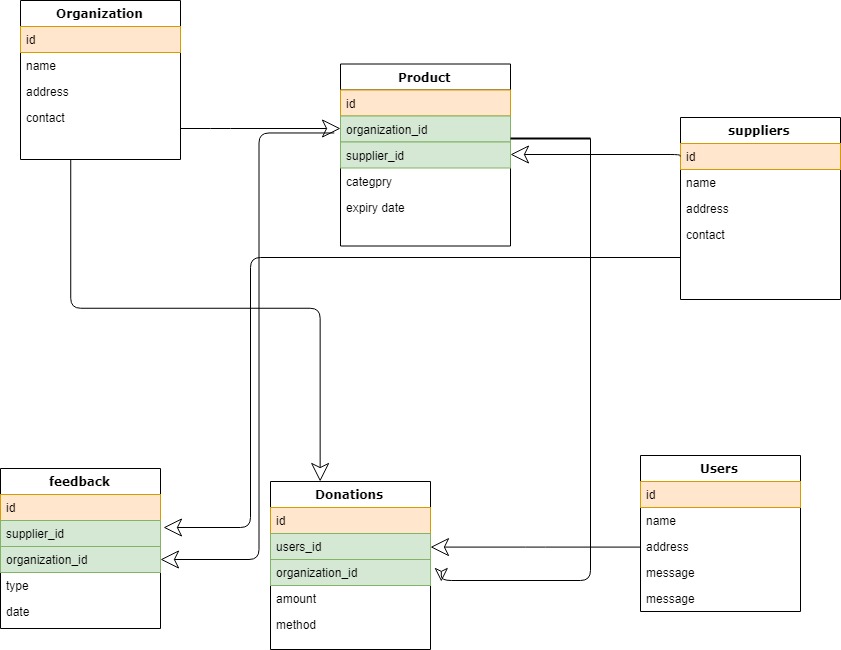


Figure ‑: Entity Relationship Schema

## Data Dictionary

A data dictionary provides a detailed description for each element or variable in your dataset and data model. Data dictionaries are used to document important and useful information such as a descriptive name, the data type, allowed values, units, and text description. A data dictionary provides a concise guide to understanding and using the data. The parameters reported in the data set need to have names that clearly describe the contents. Ideally, the names should be standardized across files, data sets, and projects, in order that others can readily use the information (Lind & Aulenbach, 2014).

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. The documentation should contain a full description of the parameter, including the parameter name, how it was measured, the units, and the abbreviation used in the data file. A missing value code should also be defined. Use the same notation for each missing value in the data set. Use an extreme value (-9999) and do not use character codes in a numeric field. Supply a flag or a tag in a separate field to define briefly the reason for the missing data. Within the data file use commonly accepted abbreviations for parameter names.

Table ‑: Description of database tables

|  |  |  |
| --- | --- | --- |
| SN | Table Name | Description |
| 1. | admins | Store the details of the user of the system with the administrative access |
| 2. | migrations | Store the records of all the tables that have been created in the Database for ease of use |
| 3. | password\_resets | Store the email and token for password details |
| 4. | tbl\_products\_details | Store the products donated by suppliers |
| 5. | tbl\_suppliers\_details | Store the supplier details for the donation |
| 6. | tbl\_users\_details | Store the records of the user that have registered for using the system |

### admins

This table stores the details of the system user who will have admin privilege. This table is used to verify credentials whenever anyone tries to access the admin dashboard. The email attribute is unique in this table. The data for the admins can’t be inserted from the Quizy application. Credentials for admins are seeded using Laravel artisan Command.

Table ‑: Details of admins table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Type | Attributes | Null | Default | Links to |
| id | int(10) | UNSIGNED | No | - | - |
| fname | varchar(191) | - | No | - | - |
| lname | varchar(20) | - | No | - | - |
| email | varchar(191) | - | No | - | - |
| password | varchar(191) | - | No | - | - |
| remember\_token | varchar(100) | - | Yes | NULL | - |
| created\_at | timestamp | - | Yes | NULL | - |
| updated\_at | timestamp | - | Yes | NULL | - |

### migrations

This table is created by the Laravel framework to manage different tables that are created using Laravel artisan command. This table is used to check if the table being created already exist in the system or not. This table is used for several purpose like migration roll back to desired state as well as refreshing the migration.

Table ‑: Details of migrations table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Type | Attributes | Null | Default | Links to |
| id | int(10) | UNSIGNED | No | - | - |
| migration | varchar(191) | - | No | - | - |
| batch | int(11) | - | No | - | - |

### password\_resets

This table is used to store the token generated for the user when the user tries to reset their password if they have forgotten. User’s email and their respective tokens are stored. The token is matched when the user resets the password. The token is sent to the user via a system-generated mail to their respective email.

Table ‑: Details of password\_resets table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Type | Attributes | Null | Default | Links to |
| email | varchar(191) | - | No | - | - |
| token | varchar(191) | - | No | - | - |
| created\_at | timestamp | - | Yes | NULL | - |

### tbl\_products\_details

This table stores the products details donated by the supplier or users. This table stores the products list in string format. This table is used to present the products view when the users wants to select the category of their products they desire to donate. In this table the attribute tbl\_products\_details\_id is used as foreign key to relate with the tbl\_suppliers\_details which contains all information about the particular supplier.

Table ‑: Details of tbl\_admin\_products\_suppliers\_details table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column | Type | Attributes | Null | Default | | Links to |
| tbl\_admin\_products\_ suppliers\_details\_id | int(10) | UNSIGNED | No | - | - | |
| tbl\_producsts\_details\_ id | int(10) | UNSIGNED | No | - | -> tbl\_products\_details.tbl\_products\_details\_id  ON UPDATE RESTRICT  ON DELETE CASCADE | |
| total\_products | int(11) | - | No | - | - | |
| created\_at | timestamp | - | Yes | NULL | - | |
| updated\_at | timestamp | - | Yes | NULL | - | |

### tbl\_suppliers\_details

The table stores the different supplier details.

Table ‑: Details of tbl\_suppliers\_details table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Type | Attributes | Null | Default | Links to |
| tbl\_suppliers\_details\_id | int(10) | UNSIGNED | No | - | ->  tbl\_users\_details.tbl\_user\_details\_id  ON UPDATE RESTRICT  ON DELETE CASCADE |
| street\_name | varchar(191) | - | No | - | - |
| ward\_name | varchar(191) | - | No | - | - |
| contact\_num | int(15) | - | No | - | - |
| created\_at | timestamp | - | Yes | NULL | - |
| updated\_at | timestamp | - | Yes | NULL | - |

### tbl\_users\_details

This table is crucial to the system. It keeps the record about the details of each users. This table stores user details like its name, email, etc. The attribute tbl\_users\_details\_id is auto generated by the system and is used as primary key and also for identifying the users.

Table ‑: Details of tbl\_users\_details table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Type | Attributes | Null | Default | Links to |
| id | int(10) | UNSIGNED | No |  |  |
| name | varchar(191) |  | No |  |  |
| email | varchar(191) |  | No |  |  |
| password | varchar(191) |  | No |  |  |
| remember\_token | varchar(100) |  | Yes | NULL |  |
| created\_at | Timestamp |  | Yes | NULL |  |
| updated\_at | Timestamp |  | Yes | NULL |  |
| verify\_email\_token | varchar(191) |  | Yes | NULL |  |

### Gantt Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Id** | **Task Name** | **Start** | **Duration** | **End** |
| 1 | Team formulation | 14-Feb | 1 | 15-Feb |
| 2 | Idea generation and Title section | 16-feb | 10 | 26-Feb |
| 3 | Proposal Writing | 27-Feb | 5 | 04-March |
| 4 | Data Collection and Research | 05-March | 7 | 12-March |
| 5 | System Design | 13-March | 11 | 24-March |
| 5.1 | Flowchart | 13-March | 3 | 16-March |
| 5.2 | ER- diagram | 13-March | 3 | 16-March |
| 5.3 | DFD | 17-March | 3 | 20-March |
| 5.4 | Use Case Diagram | 21-March | 3 | 24-March |
| 6 | Coding and Testing | 25-March | 70 | 03-June |
| 6.1 | Main Layout and UI Design | 25-March | 7 | 01-April |
| 6.2 | Login and Registration Module | 02-April | 5 | 07-April |
| 6.3 | Admin and User Panel CRUD | 08-April | 8 | 16-April |
| 6.4 | Feedback module | 17-April | 10 | 27-April |
| 6.5 | Google Map Module | 28-April | 5 | 03-May |
| 6.6 | Unit Testing and verification | 04-May | 4 | 08-May |
| 6.7 | Integration of Different Module | 09-May | 9 | 18-May |
| 6.8 | System Testing and Validating | 19-May | 7 | 26-May |
| 6.9 | Implanting and Alpha Testing | 27-May | 5 | 01-June |
| 7 | Mid Term Evaluation | 02-June | 2 | 04-June |
| 8 | Supervisor Meeting | 05-June | 10 | 15-June |
| 9 | Project On Hold | 16-June | 20 | 06-July |
| 10 | Project Report Documentation | 07-July | 30 | 06-August |

# System Development

## Software Development Model

### Agile Development Model

Agile Method is an umbrella term that refers a group of development process, and not any single model of software development. Agile method is the combination of iterative and incremental development model. It breaks the product into small increment builds and these builds are developed in iterative manner (Hughes, Cotterell, & Mall, Agile Development, 2015).



Figure ‑: Agile Development

Agile development eliminates heavy weight implementation methodologies. In Agile development process featured requirements are decomposed into several parts that can be incrementally developed. Each incremental part is developed over an iteration. We use extreme programming method of agile development as the process model for our project.

#### Extreme Programming

Extreme Programming is one of the agile methodology for system development. Here changes in requirements can be easily handled. It improves the quality and responsiveness of system application. Fundamental principle of extreme programming is rapid feedback, simplicity, incremental change and quality work (Hughes, Cotterell, & Mall, Extreme Progrmming, 2015).

Extreme Programming (XP) has been declared to be a new way of software development: a lightweight methodology, which is efficient, low-risk, flexible, predictable, scientific, and distinguishable from any other methodology. In the core of XP practices are programming activities, with strong emphasis on oral communications, automated tests, pair programming, storytelling culture and collective code-ownership at any time in the XP project. The paper gives an overview of XP practices and raises some serious concerns regarding their role in conceptual modelling and code generation; which directly affects software architecture solutions (Extreme programming and its development practices, 2000). There are major five phases in this agile methodology:

#### Planning

The first step we are going to take is planning. We will convert the requirements into the iteration which is usually a small parts of the system functionalities. We will focus on deadline of the iteration so that we will be able to complete the requirements on time.

#### Designing

After the planning we will start designing the overall system. Here we will include object oriented design models (UML) to illustrate the overall functionalities of the system. This design models will show the active users of the system and their involvement in every process. Entity Relationship graph and data dictionary are developed at design part.

#### Coding

This is the most important phases that we should take into account while developing the system. Here we will do pair coding for developing the system using one machine so that at the end we will be able to produce high quality code and at the same time reduce the overall cost. It will also help us to reduce the conflict and optimize the functionalities of the system.

#### Testing

The testing is done to verify and validate the Quizy system. The objective of testing is to find as many bugs as possible and fix them so that it is free from any programming and logical error and the results would obtain as expected from the system.

#### Feedback

Feedback is the important phase where the developers directly get feedback from the customer or the project manager regarding the project. We took the feedback from each other as well as our supervisor after completing each module. This is the reason we can enhance our system make it more applicable as per the demand of the market.

## Front End Tools

A "front-end" application is one that application users interact with directly. The front end is an interface between the user and the back end. In simple words we can understand the concept of front and back end with the help of below given definition. In client/server applications, the client part of the program is often called the front end and the server part is called the back end. While creating a front end, different components relating to the software development were used. These components are explained below.

### HTML5/CSS3

HTML5 is a markup language used to design the webpages. It uses predefined tags to create the webpages. It is run using a web browser. We have used HTML5 to create the front-end of our system. We specially choose HTML5 for designing front-end of Quizy since it is the latest version and it contain number our new elements that helped to make our application more interactive.

CSS is a style sheet language that is used to describe the appearance and formatting of a web document, which is written in a markup language. It enhances the presentation of the web document. We used CSS3 to make Quizy front-end more attractive and to give the viewer better experience with its all new feature.

### BootsrapVue

BootstrapVue, based on the world's most popular framework - Bootstrap v4, for building responsive, mobile-first sites using Vue.js. BootstrapVue is used in our application as it is easy to use and understand. It makes the Quizzy responsive and adds more attractive features.

### Vue

Vue is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

## Back End Tools

Back-end refers to everything the user can’t see in the browser, like database and servers. Back-end developers or programmer is required to create a dynamic website. A dynamic web application is constantly changing and updated in real time. Quizy is developed by using following back-end technologies:

### Laravel Framework

Laravel framework is a clean and well-structured PHP framework for web application development. It is a free, open-source framework and distributed under the MIT License. Laravel was created by Taylor Otwell and initial version was released in June 2011. Laravel provides powerful tools needed for large and small, robust application (Laravel Framework Introduction, 2018). The main features of Laravel are explained below:

#### Artisan Command

Artisan commands help developer to quickly create skeleton code for doing some administrative tasks, there are serval common task that we have to do respectively during the development process such as the creation of database schema and build their migrations, creation of controller, middleware, views etc. We can create, delete, update database, controller class, middleware class etc. by entering few command-line codes (Artisan Development, 2016). Some of them are listed in Table 5.3-a.

Table ‑: List of artisan command and description

|  |  |
| --- | --- |
| Artisan Command | Description |
| php artisan list | To view a list of all available Artisan commands |
| php artisan cache:clear | To flush the application cache |
| php artisan key:generate | To set the application key |
| php artisan help migrate | To view a help screen, precede the name of the command |
| php artisan tinker | To enter the Tinker environment |
| php artisan –version | To view the current version of your Laravel |
| php artisan make:controller | To create controller class |
| php artisan make:middleware | To create middleware class |
| php artisan make:provider | To create provider class |
| php artisan make:test | To create test class |
| php artisan serve | To Start the development server with default options |
| php artisan migrate:install | Creates the migration repository |
| php artisan migrate:make | To create the migration |
| php artisan migrate:refresh | Resets and reruns all the migrations |
| php artisan migrate:reset | Rollback all the database migrations |
| php artisan migrate:rollback | Rollback the last database migration |
| php artisan migrate:status | To show the status of each migration |
| php artisan view:clear | Clear all compiled view files |
| php artisan route:cache | Create a route cache file for faster route registration |
| php artisan route:list | List all registered routes |

#### Laravel MVC Pattern

Basically, Laravel is a fully Model-View-Controller (MVC) compliant framework. MVC pattern of Laravel ensures clarity of logic and presentation, and also helps in improving the performance, security, and allowing the better documentation. The MVC architecture pattern let the developer write a code that can be divided on the basis of the following three things:

##### Model

A Model is the layer by which developer can manipulate data, it lies between the data and application. The data itself can be stored in various types of database systems such as Maria DB, MS Access, MySQL, SQL Lite or Excel files etc.

##### View

Views are the visual or presentation layer of our web application, view is responsible for presenting the data that the Controller received from the model. In Laravel, Blade Templates are used as view.

##### Control

The primary function of controller is to handle requests and return appropriate responses to the correct Views. Controller acts as the middle man between Model and View. Controller is the logical layer of the application. Controller can receive data from view then process data and write the data to database, and redirecting users to appropriate routes.

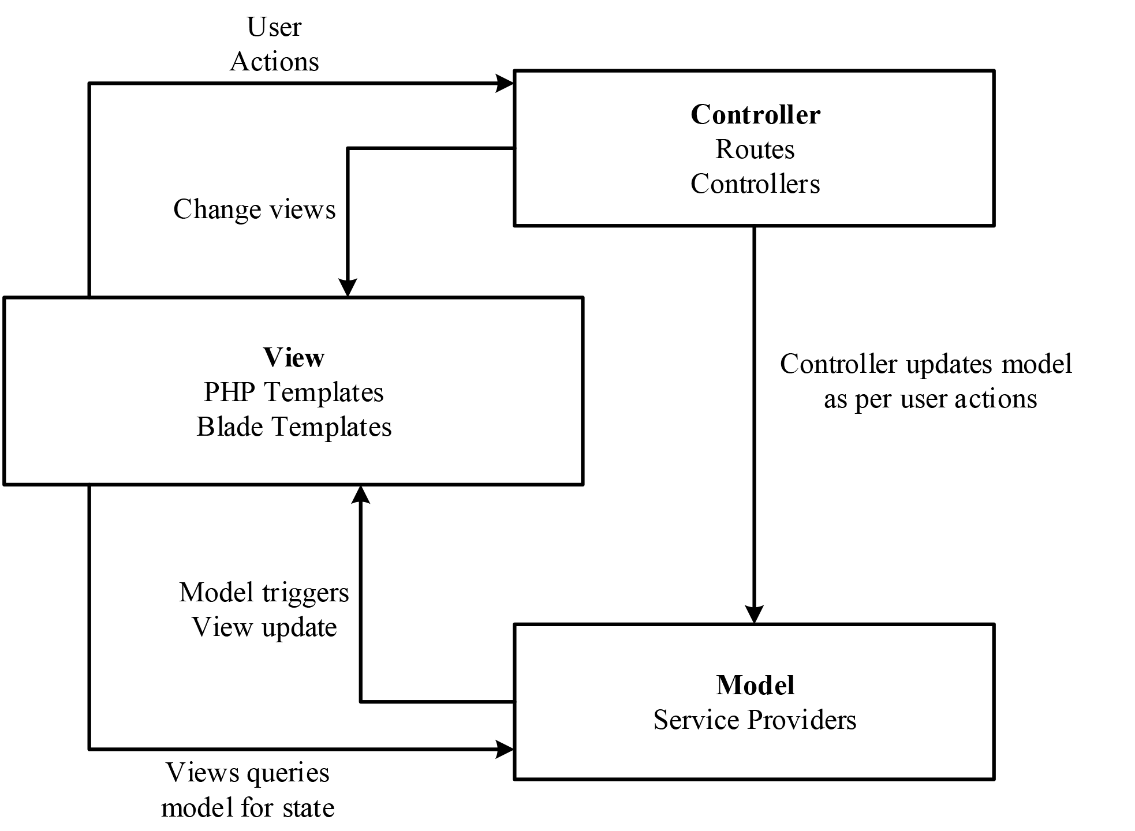


Figure ‑: Model-View-Control

#### Structure of Laravel application

The default Laravel application structure is intended to provide a great starting point for both large and small applications. Of course, we are free to organize our application whatever we like. Laravel imposes almost no restrictions on where any given class is located - as long as Composer can auto load the class. The below Figure 6.2-B (Directory Structure, 2017) shows the structure of Laravel application:

|  |
| --- |
| ├── app # Your Laravel application  │ ├── Commands  │ ├── Console  │ ├── Events  │ ├── Exceptions  │ ├── Handlers  │ ├── Http  │ ├── Providers  ├── artisan # Artisan command-line utility  ├── bootstrap  │ ├── app.php  │ ├── autoload.php  │ └── cache  ├── composer. json  ├── config  ├── database  │ ├── migrations # Database migration classes  │ └── seeds # Database seeder classes  ├── package.json  ├── phpspec.yml  ├── phpunit.xml # PHPUnit configuration file  ├── public  │ ├── assets  │ ├── favicon.ico  │ ├── index.php # Entry point to run the application  │ ├── packages  │ ├── robots.txt  ├── resources  │ ├── assets  │ ├── lang  │ └── views # Contain blade template  ├── server.php # A lightweight local development server  ├── storage  │ ├── app # App storage, like file uploads etc  │ ├── framework # Framework storage (cache)  │ └── logs # Application-generated logs  ├── tests # Contain all test cases  └── vendor # All third-party libraries installed by Composer |

Figure ‑: Laravel application structure

### MySQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structural Query Language (SQL). It is the most popular open source database. It is highly noted for its quick processing, proven reliability, ease and flexibility of use. It also complies on a number of platforms. (Back up and Recovery, n.d.)

All the database of Quizy are created using MySQL. It gives the flexibility of open source, high performance and data security. It also supports large databases. Comprehensive transactional support, complete workflow control and compatible with multiple platforms are some other features of MySQL that attracted us to use it for Quizy.

### Pusher

Pusher empowers developers with APIs to create collaboration & communication features in their web and mobile apps. Pusher Channels provides real-time communication between servers, apps and devices. Channels is used for [real time charts](https://pusher.com/docs/channels/getting_started/javascript-realtime-chart), real-time, real-time maps, multiplayer gaming, and many other kinds of UI updates. We used pusher in our project for real time updating of the scores of players during the game session.

### Apache Server 2.4.35

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 20th birthday as a project in February 2015 (Apache Server, 2017).

Apache Web Server is open-source Web server creation, deployment and management software. Initially developed by a group of software programmers, it is now maintained by the Apache Software Foundation. Apache Web Server is designed to create Web servers that have the ability to host one or more HTTP-based websites. Notable features include the ability to support multiple programming languages, server-side scripting, an authentication mechanism and database support. Apache Web Server can be enhanced by manipulating the code base or adding multiple extensions.

## Others Tools

### PHPStorm JetBrains

PhpStorm is a powerful editor for PHP, HTML and JavaScript with on-the-fly code analysis, error prevention and automated refactoring for PHP and JavaScript code. It is built on JetBrains' IntelliJ IDEA platform. We are found this IDE very user friendly and using throughout the development of the project. We are using the Cracked version of PhpStorm.

### Windows Visio 2016

Microsoft Visio (formerly Microsoft Office Visio) is a diagramming and vector graphics application and is part of the Microsoft Office family. The product was first introduced in 1992, made by the Shapeware Corporation. It was acquired by Microsoft in 2000. We are using this tool to draw different graphical representation (Flowchart, ER-Diagram, Data Dictionary, UML Diagrams, Use Case Diagrams) for our project.

## Graphical User Interface

### Home Page

*This is the page in Appendix A.1* that is rendered when the application is loaded first. Here user can join the quiz of which they should have pin of. The visitor can join the game with their name and the pin associated with the quiz they want to join. After entering the valid pin, the user is redirected to the game page, where they can play the quiz.

### User Login Page

This is the page which is rendered when the host wants to login. Only registered users can login through this page. With success correct credentials, the user is redirected to the dashboard of Host. *We can see the user login page in Appendix A.3.*

### Register Page

This page is used to register a new host. Only host are allowed to host quiz, thus one should register to host quiz. Once the user registers with the required credentials, they will have to wait for their account approval by the admin. *The password reset page is in Appendix A.4*

### Admin Page

*This is the module that we can see in Appendix A.5* is used by admin to manage the host in the application. The admins can also manage the categories used in the quiz. Only the user with authorized credentials can have access to this page.

### Host Page

The Host page is where the quiz is made by authenticated user. They can manage the questions with multiple categories, can add them in quiz and host the quiz. Hosts can create multiple questions and attach them to quizzes. *We can see the host layout page in the Appendix A.7.*

### Game Page

The Game page is where the quiz is played. The player with authentic pin joins the quiz. After the host starts the quiz, the player is presented with a question with four options. The player has to select their option and submit. Based on their answer, the score is reflected on the screen. *We can see the game layout page in the Appendix A.7.*

# Testing

## Unit Testing

Unit testing is the testing of individual software component or module. These tests are usually written by the developers of the module or programmer themselves. We performed the testing of test case by providing the input to the system then analyze the result.

### Test case for user login

Table ‑: Test case for user login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Description | Input test data | Expected Result | Actual Result | Pass / Fail |
| 1. | Enter valid data for username and password | Username:  user@user.com  Password: user123 | Should redirect to the home page | Display the home page | Passed |
| 2. | Enter invalid username and password | Username: user@usr.com  Password: rtwedfw | Error message display as the username and password you entered is incorrect | **These credentials do not match our records.** | Passed |
| 3. | Username or password field empty | Username: test@test.com  Password: | An error message should be displayed | Error message display as the input field must not be empty | Passed |

### Test case for admin login

Table ‑: Test case for admin login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Description | Input test data | Expected Result | Actual Result | Pass / Fail |
| 1. | Enter valid data for username and password | Username:  admin@admin.com  Password: admin123 | Should redirect to admin dashboard. | Display you are logged in as Admin! | Passed |
| 2. | Enter invalid username and password | Username: igsdj  Password: jdsf | Error message display as the username and password you entered is incorrect | **These credentials do not match our records.** | Passed |
| 3. | Username or password field empty | Username: example@ab.com  Password: | An error message should be displayed | Please fill out this field. | Passed |

### Test case for user registration

Table ‑: Test case for user registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Description | Input test data | Expected Result | Actual Result | Pass / Fail |
| 1. | Enter valid first name, last name, email and password and re-password field is same | Name: Sishir Email:sishir@test.com  Password: user123  Re-password: user123 | Show Email is successfully send verify your account | Email is successfully send, please your email. | pass |
| 2. | Enter valid first name, last name, email and password and re-password field is different. | Name: Ghimire  Email:ghimire@test.com  Password: user123  Re-password: usr123 | Password do not match. | Display password is not matching. | pass |
| 3. | Any one field is empty. | Name:  Email:subash@test.com  Password: user123  Re-password: usr123 | Show This field is required. | Displaying \* this field is required. | pass |

## Integration Testing

The Integration testing part of a testing methodology is the testing of the different modules/components that have been successfully unit tested when integrated together to perform specific tasks and activities. The test is often done on both the interfaces between the components and the larger structure being constructed, if its quality property cannot be assessed from its components. After integrating the requirements, we tested it, it was fine and satisfactory.

## System Testing

The system testing part of a testing methodology involves testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system, and then testing it as a whole. This testing is listed under the black-box testing method, where the software is checked for user expected working conditions as well as potential exception and edge conditions.

### Black Box Testing

Black box testing is software testing where functionality of the software is tested without looking the internal code structure, knowledge of internal paths and the implementation details. It typically involves running through every input to verify that the result is as expected. We have decided to perform Equivalence Class Testing and Boundary Value Testing for our system. Our system has successfully passed these vary test and ready for the implementation on the real world.

### White Box Testing

White box testing is software testing where functionality of the system is ignored and only focuses on the code and the structure of the code. This testing is done to check whether all the code implements correctly. To ensure this to happen in our system we performed Branch Coverage technique and our system passed this testing as every code is working as expected.

# Implementation and Result Analysis

This chapter covers the implementation stage of the Quizy System. The implementation phase involves system implementation in the Real-World Environment and to ensure its proper functionality. Before the system can be implemented, certain requirements must be meet. The Quizy System requires different hardware requirements for the server specified in the Table 2.4-a and software requirements for the server specified in the Table 2.4-b. As for the client requirements, the client must have an access to browser supported computing device and working internet connection.

After the requirements are meet, the system can be implemented and can be used to carry out day-to-day transaction. Beside these, the system must have PHP installed and MySQL server for the system to work. PHP is required as the system is developed using PHP language and the MYSQL server is used to handle the database operation. The system is developed in MVC pattern using Laravel framework. The system needs to be hosted on the WWW so that anyone can have access to the system from anywhere on the globe. ( Armel, 2014)

Quizy is designed and developed in two components: - admin and user components. The admin components are used only accessible to the administrator with authorized credentials. The administrator can manage all the details of the users, their products, confirmations and cancellations as well. The administrator is also responsible for managing the users who are willing to perform transaction using the Quizy system. The User component is accessible to all the clients, users via internet using a web browser. Users of the system can donate their products online by providing their contact number and house address. The user can also donate their funds through different payment gateways.

There are several payment services that can be integrated with Quizy like online banking, online payment like e-sewa. But currently, Quizy can only provide donation services through Khalti payment merchant. In order to use Khalti payment, the user must have an authorized account of Khalti with valid mobile number. The user must have enabled third party transaction pin to use the payment service. Initially the user enters their valid Khalti mobile number to which a verification code is sent. This verification code along with third party transaction pin is used to verify the transaction from the client side. Whenever your customer pays using the Khalti widget, the client side makes a request to the Khalti server to initiate and confirm the payment. Once they've confirmed the payment, the client will receive a response containing unique token and amount for that particular transaction. Upon receiving the transaction token, the client will make a request to your server with the token and the payment amount. (Khalti)

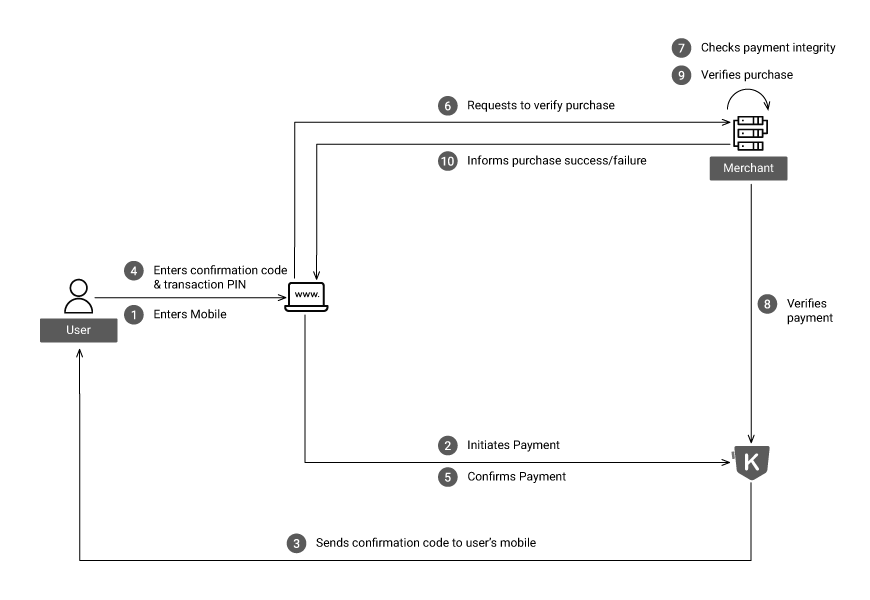


Figure 7-a: Khalti Payment Process

Quizy is a web-based product and funds donation system that aims to provide the user with real time donating their products. In order to provide online payment service, different payment gateways can be integrated as per the client’s requirements. After the system is implemented, the different module of the system is executing well and producing the desired output as expected. The system reduces the overall time involved in travelling from places to place. Ones do not need to visit the organization or donation camp to donate their products and funds. They can donate from their home.

# 

# 

# Conclusion and Future Enhancement

Quizy is a web based online product donating system that aims to provide the user with real time donating service. Quizy is developed for the ease of the user as it saves the time and is accessible from anywhere with just the help of web browser and internet. To build a system a variety of requirements are needed. To gather the requirements for our system we used the basic methods like questionnaires, interaction, internet research and observation. We get to know about the views of the people, their needs and expectation from this survey. We even learnt about the system that is currently running in the real-world scenario. It helped us to find what services they provide and how. The study of the current system motives us to build Quizy and provide those services that the current system lacks.

Quizy implements agile development model to develop. In the agile model, the feature requirements are decomposed into several small parts that can be incrementally developed. Agile development model adopts iterative approach. Each incremental part is developed over iteration. Agile focuses on the iterative process with small team of developers, deploying pair programming and having a customer representative in the team. Under agile development model we are using extreme programming approach. XP has been declared to be a new way of software development: a lightweight methodology, which is efficient, low-risk, flexible, predictable, scientific and distinguishable from any other methodology. We have used the XP model due to its features like programming activities, with strong emphasis on oral communications, automated tests, pair programming, storytelling culture and collective code-ownership at any time.

The development of Quizy needed a number of technologies. We used HTML, CSS, Bootstrap, Semantic UI and jQuery to design our frontend and Laravel framework, MySQL, Apache server to design our backend. During the design and development of our system we learnt to use these development tools. We got the knowledge about the use and implement of these tools and also learn the flow of these tools. The Laravel frame work was the new workspace for us but as we used it we get the idea of what MVC pattern is and how to implement it in a system design. As every system need to validate and verified, we also, put Quizy through different testing methods and the outcomes were obtained as expected from the system. So, this system is flawlessly validated and verified.

Quizy is a web-based application to provide online product donation with the help of only web browser and the internet. This system is only limited to the web application now but it can further extend as mobile application as well as desktop application. We can also further enhance it in Multilanguage. This system has a great scope in the future as Nepal is a developing country. Every process of our country is converting from manual to digital so this system can provide a step in the digitization process of Nepal. As time changes people are getting more and busier, so in that busy schedule people can use this application to save their time and effort.

As we all know, every system is not perfect in this world. We are also aware that every system needs to make some compromise or limit its feature to fully develop.  We have to admit that the current Quizy lacks some of the objectives due limitation off technologies we can use.

# Bibliography

Armel, J. (2014). *Web application development with Laravel PHP.* Helsinki: Metropolia University of Applied Sciences. Retrieved from https://www.theseus.fi/bitstream/handle/10024/74052/Author.pdf

*Apache Server*. (2017, October 23). Retrieved from apache.org: https://httpd.apache.org/

*Artisan Development*. (2016, August 23). Retrieved from Laravel: https://laravel.com/docs/5.0/commands

*Back up and Recovery*. (n.d.). Retrieved from MySQL: https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html/

Data Flow Diagram Notation. (2009). In S. Tayal, B. B. Agarwal, & M. Gupta, *Software Enginnering & Testing* (pp. 62-63). Jones and Bartlett Publisher.

*Directory Structure*. (2017, January 24). Retrieved from Laravel: https://laravel.com/docs/5.4/structure

Hughes, B., Cotterell, M., & Mall, R. (2015). Agile Development. In B. Hughes, M. Cotterell, & R. Mall, *Software Project Management* (pp. 88-89). Delhi: McGraw Hill Education (India) Private Limited.

Hughes, B., Cotterell, M., & Mall, R. (2015). Extreme Progrmming. In B. Hughes, M. Cotterell, & R. Mall, *Software Project Management* (5th Edition ed., pp. 89-92). Delhi: McGraw Hill Education (India) Private Limited.

Juric, R. (2000). Extreme programming and its development practices. *22nd International Conference on Information Technology Interfaces* (pp. 97-104). Pula, Croatia: IEEE. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=915842&isnumber=19790

*Laravel Framework Introduction*. (2018, February 7). Retrieved from laravel.com: https://laravel.com/docs/7/introduction

Lind, E., & Aulenbach, S. (2014). *Create a data dictionary*. Retrieved from DataOne.org: https://www.dataone.org/best-practices/create-data-dictionary

Vue Js. (2019, July 7) Retrieved from vuejs.org: https://vuejs.org/v2/guide/

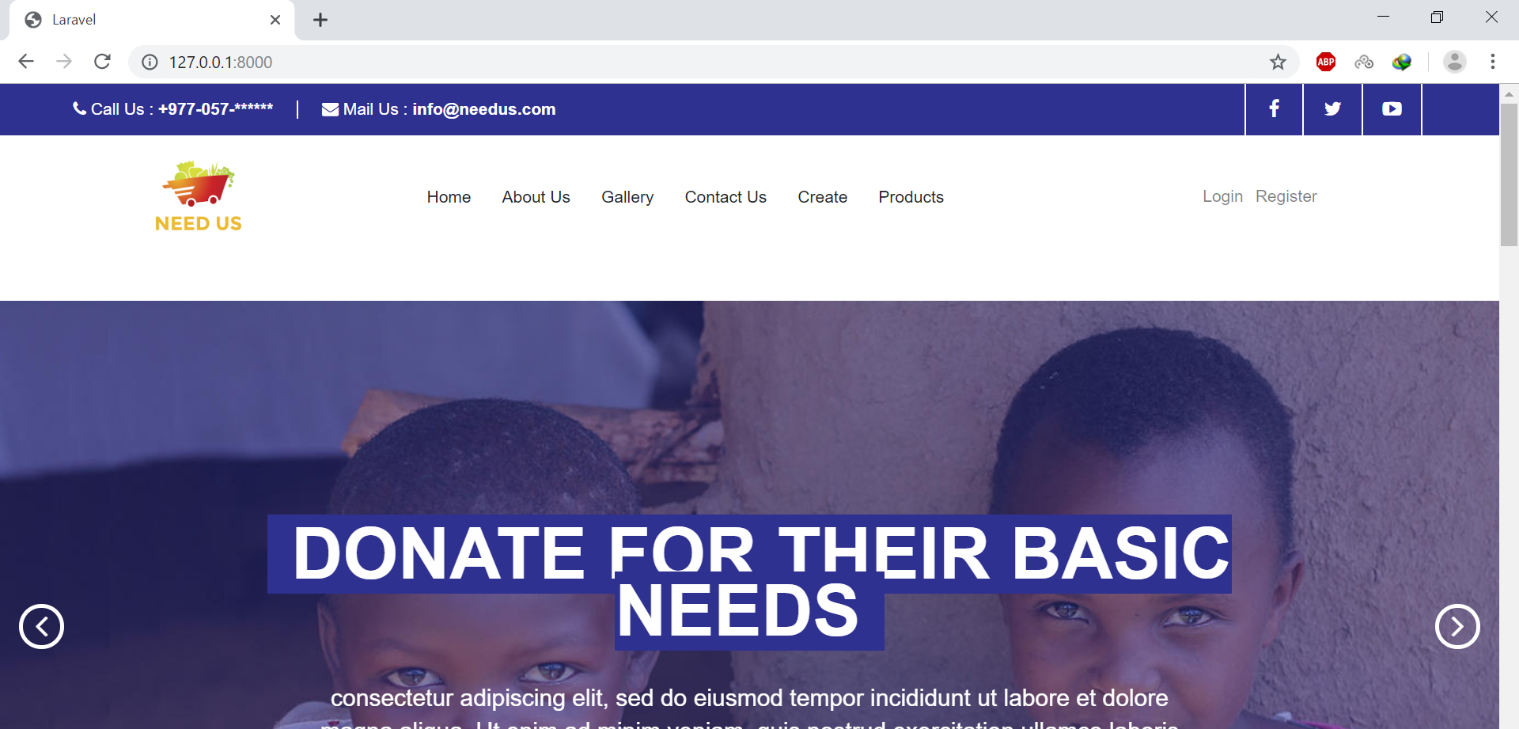
NTA. (2019, June 15). *MIS Reports*. Retrieved from Nepal Telecommunications Authority: https://nta.gov.np/wp-content/uploads/MIS-2076.pdf

Resig, J. (2015, May). John Resig: Building JQuery. (S. Charles, Interviewer) www.computer.org. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7111910&isnumber=7111853

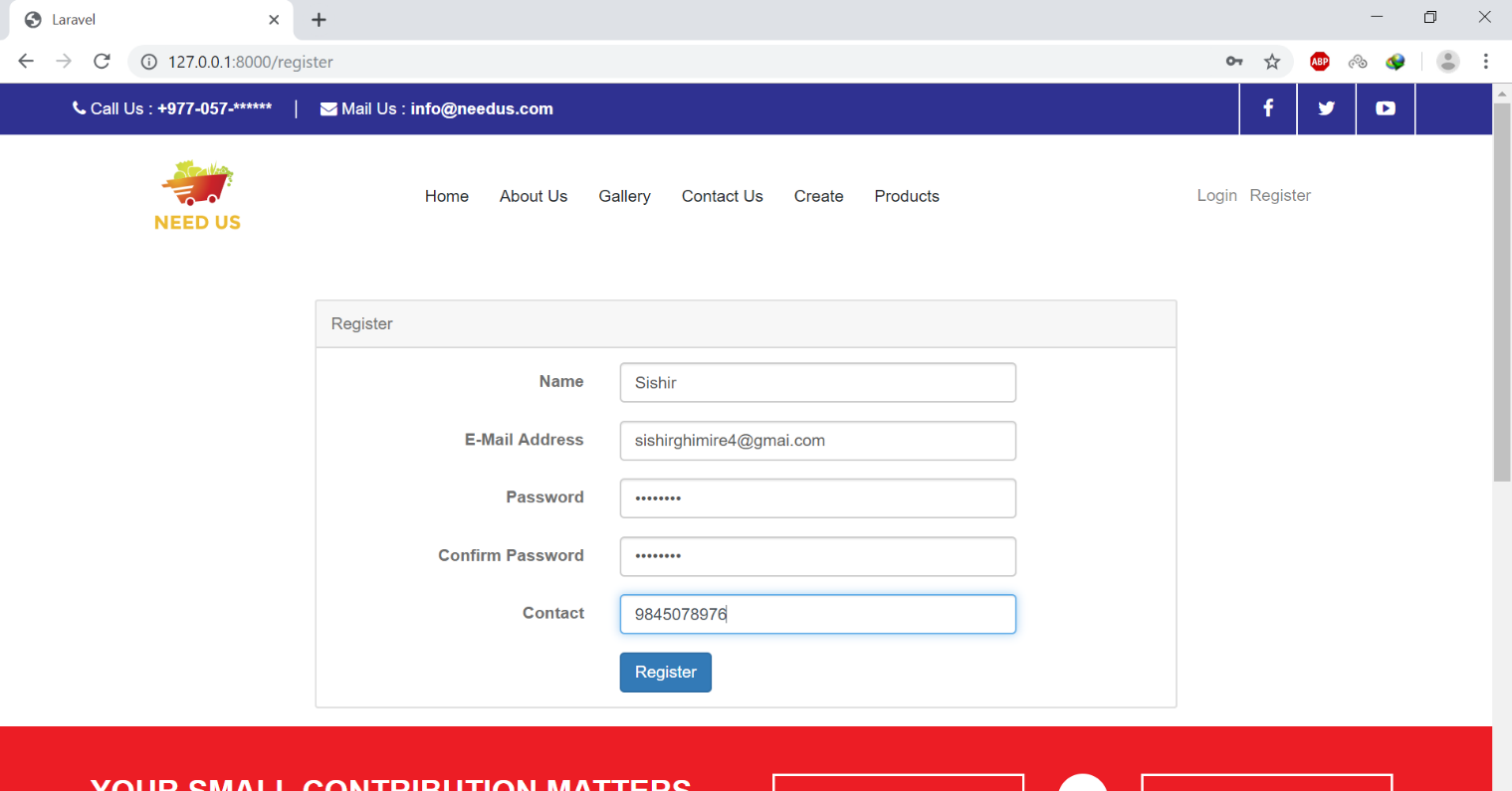
*Server Requirements*. (2017, January 24). Retrieved from Laravel.com: https://laravel.com/docs/7/installation#server-requirements

# Appendix

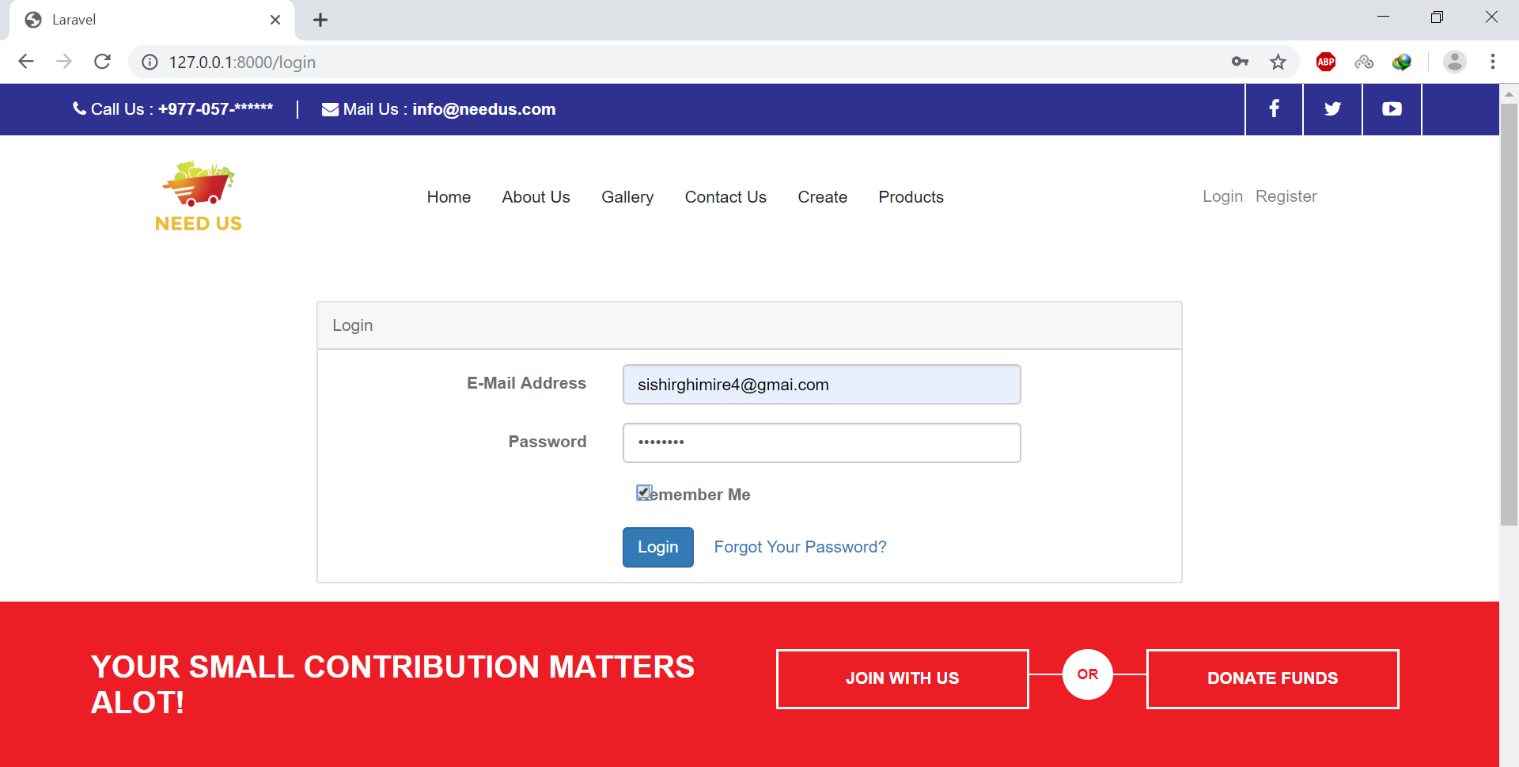
1. **Home Page for Quizy**



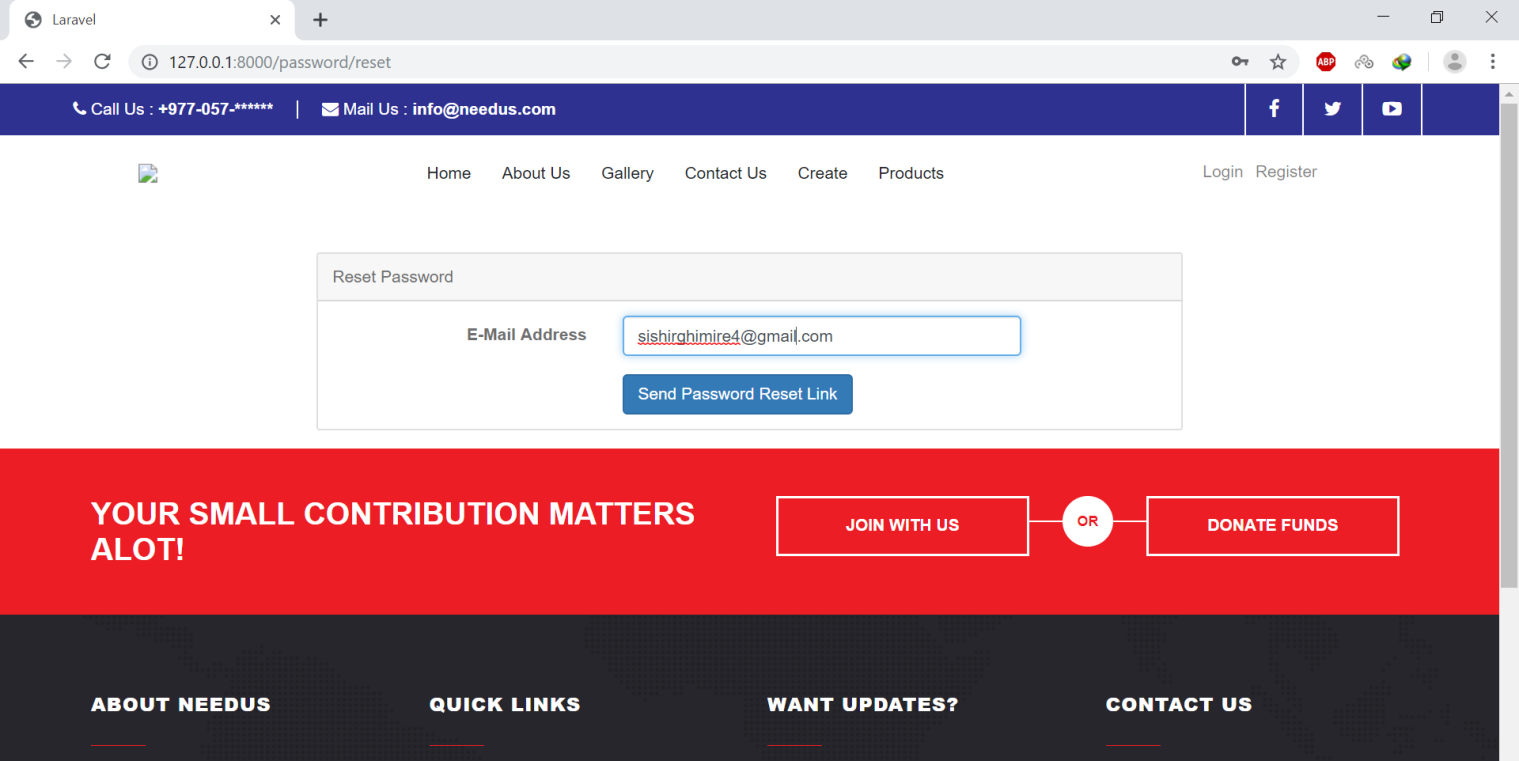
**A.2 User Registration**

****

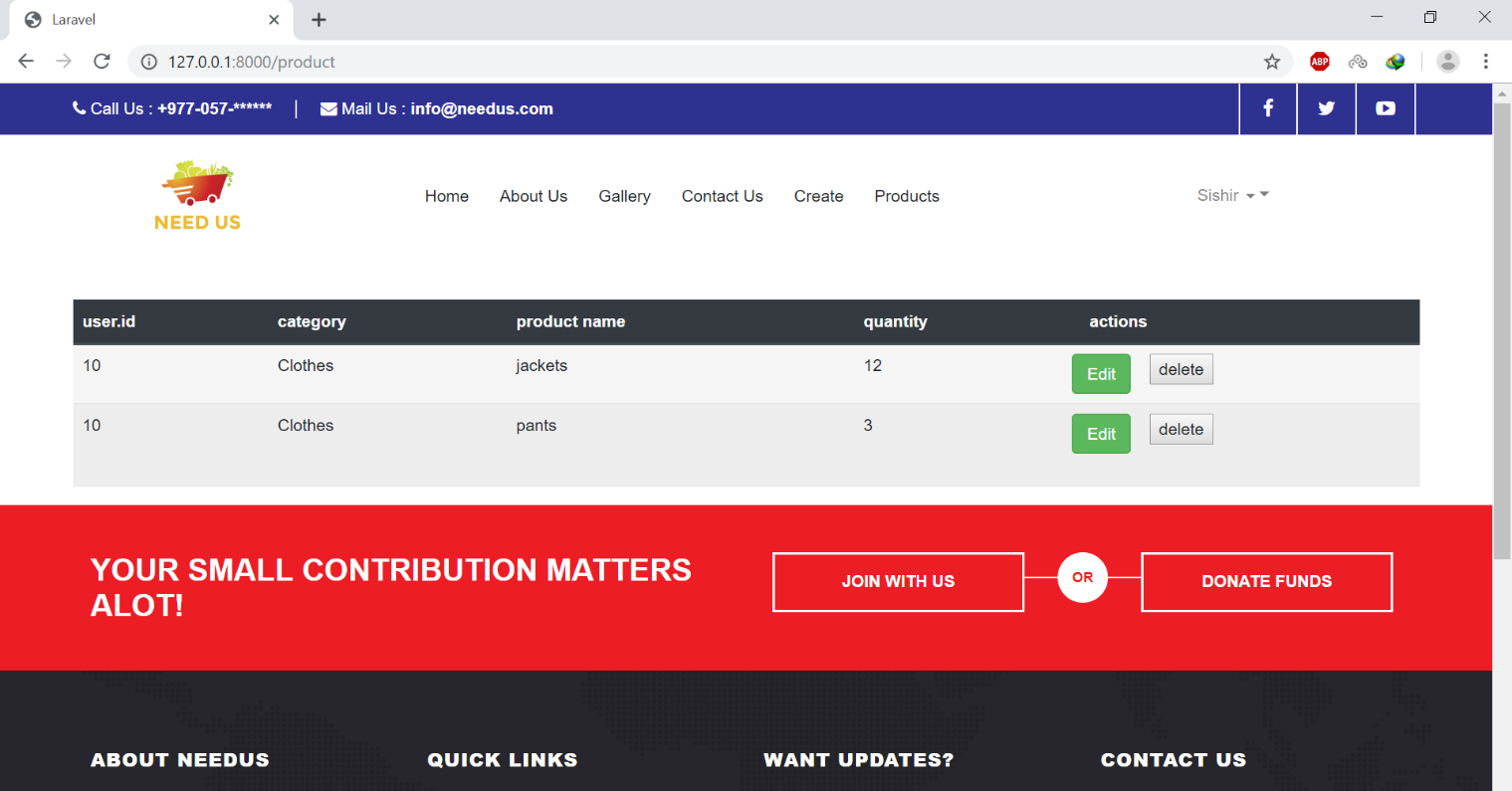
**A.3 User Login Page**

****

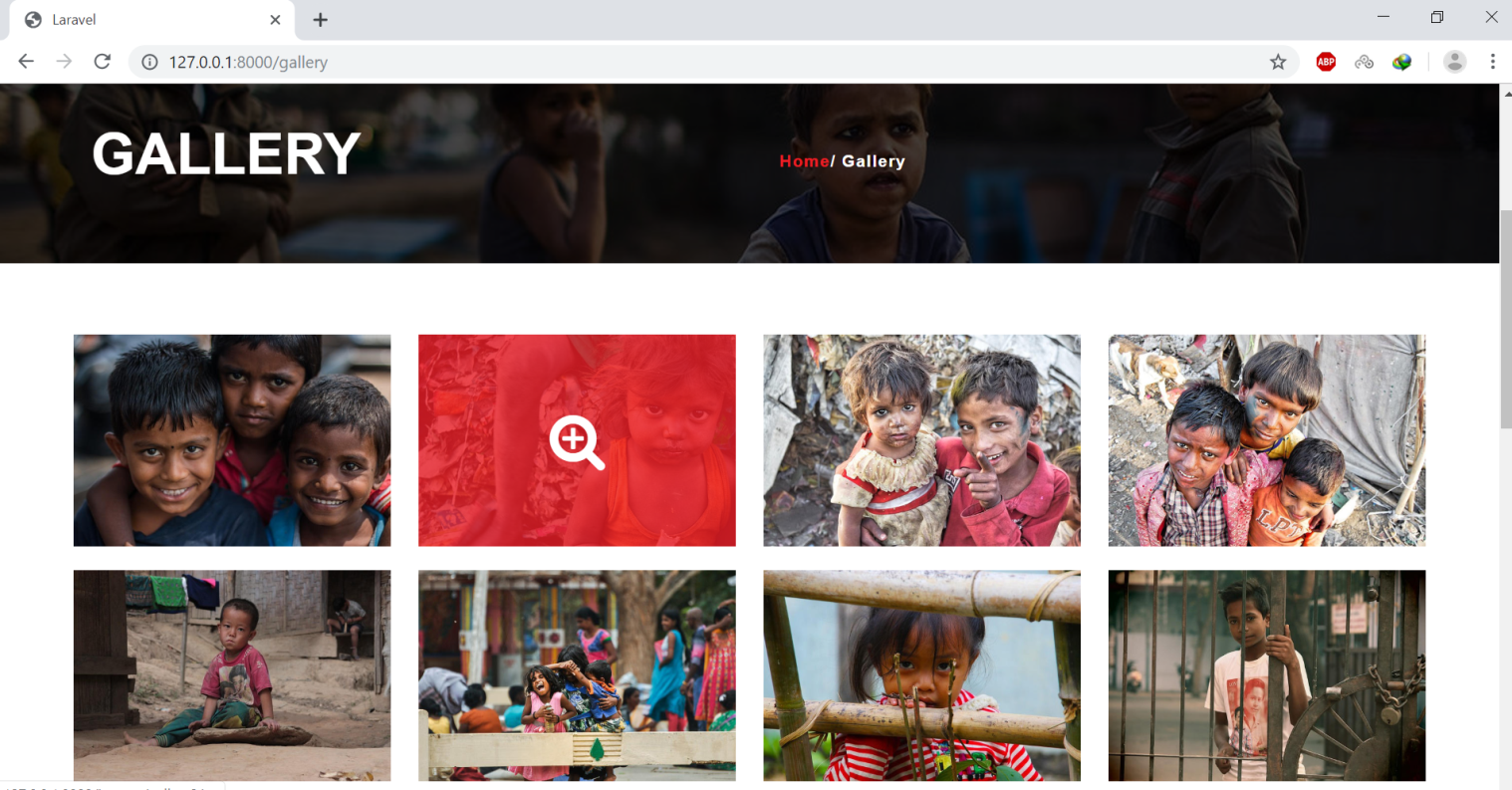
**A.4 Password Reset Page**



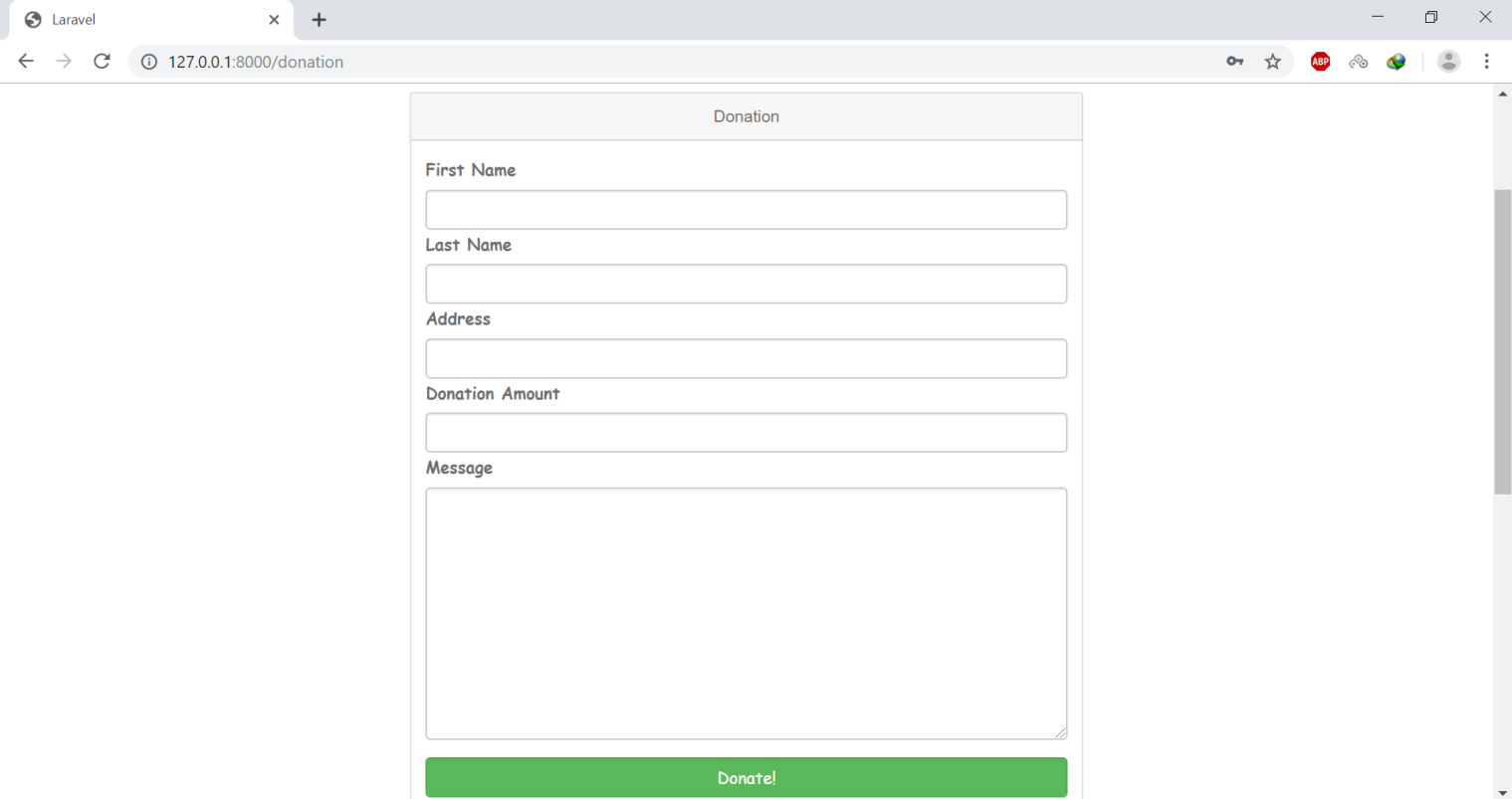
**A.5 Products CRUD Page**



**A.6 Gallery Page**



**A.7 Fund Donation Layout Page**

****