# **CHAPTER 1: INTRODUCTION**

## **1.1. Introduction**

“Museum Nepal” ticket booking system is a web application created with the objectives of booking tickets of museum and heritage places of Nepal. “Museum Nepal” provide all details about historical places, it is specially designed to make your booking experience better. Customers can view the list of museums and their history, importance etc. and book any museum ticket as needed. This system also contains notices which show museum timing, off days etc. This system also contains museum details where visitors can learn museum importance and history. The system automatically calculates the subtotal and grand total. When a visitors book the ticket, the order information including the visitor’s name, address and billing instruction is stored in database securely and payment has been made. User should register when they first visit the site then for future it will be stored in system database permanently and they can book ticket anytime using their username and password. Admin can keep records of tourists visiting daily, weekly, monthly and yearly. Talking more about the project, the system displays details of total sold tickets, total student visited, Nepali as well as foreign citizen, staff and total earning of the day in a proper manner.

## **1.2. Problem Statement**

The traditional way of booking the ticket for the museum is the customer need to go to the desired museum and need to stand in queue and buy the ticket for the museum this will become more difficult for a person in order to overcome this problem The project gives real life understanding of online museum ticket booking system and the importance, history of museums. Here we provide website for museum ticket booking system. Online museum ticket booking system project provides enhanced techniques for maintaining the required information up to date, which results in efficiency. In offline ticket booking system things get worse when travel season came. Adapting online system makes life simple by saving time and energy. Though offline system exists if any inconvenience situation occurs.

Offline ticket booking makes life complicated when you instantly need it, in ticket counter you have to wait for long for the person who sells ticket not reach, not give any response on telephone which kills energy and time. We can even not get notice weather the museum is closed or opened.

## **1.3. Objectives**

The museum ticket booking is currently maintaining a booking process manually which is very time-consuming process. It deals with all paper work, so it becomes a very tedious job for the employee to look after it. The objectives are given below: -

* To develop web-based application for booking museum tickets.
* To develop a web application that supports museum management committee to solve all tedious tasks related to ticket booking.

## **1.4. Scope and Limitations**

Every website has its own features and its limitation. Museum ticket booking system offers following scope and limitation:

### **1.4.1. Scope**

Museum ticketing site focuses on providing quality services for the visitors. Museum ticket booking system meets following scopes in early stages:

* This system assists visitors in locating the greatest museum in the city.
* A single website can be used to purchase multiple museum tickets.

### **1.4.2. Limitations**

This system meets the limitations mentioned below in earlier stages, but no system is perfect. It is extremely difficult to design a system that is both accurate and efficient.

* There are no basic features like reviews or comments available.
* Only museums located within Kathmandu Valley are available.

## **1.5. Report Organization**

The report has been prepared following the guidelines provided by the Faculty of Humanities and Social Science, Tribhuvan University. The report is separated into different chapter Each chapter consists of various sub chapters with its content. The preliminary section of the report consists of Title Page, Acknowledgement, Abstract, Table of Contents, List of abbreviations, List of Figures and List of Tables. The main report is divided into 5 chapter, which include:

### **Chapter 1: Introduction**

It includes the general overview of the system and the project as a whole. It includes the Introduction Problem Statement, Objective, Scope and Limitations of the project etc.

### **Chapter 2: Background Study and Literature Review**

It includes the study of the current scenario the system will be deployed into. It includes the Review of the similar projects, theories done by other researchers.

### **Chapter 3: System Analysis and Design**

It includes the requirement and feasibility analysis of the system that can be generated through the studies of previous two chapters. It also includes Data Modelling (ER-Diagram), Process Modelling Diagram (DFD) and Architectural Design, Database Schema Design, Interface Design. in design phase.

### **Chapter 4 Implementation and Testing**

It includes the details of the different design and development tools used. The implementation details of the modules presented in the form of code snippets of functions, classes, it also includes the testing of the system with different test cases as per the requirement.

### **Chapter 5: Conclusion and Future Recommendations**

It includes the summary of the system and the project as a whole It also includes the possibilities system can implement in the future.

The final part of the report consists of References and Appendices. The references are listed in accordance to the IEEE referencing standards and the Appendices includes the screenshots of the system.

# **CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW**

## **2.1. Background Study**

The current ticket booking system in the museum is slow because each task is performed by a human. There is no online system for purchasing museum tickets. Because of the increased use of the internet in today's world, everyone desires an online system. As a result, we created this system to facilitate effective booking. The following are some general concepts and terminologies:

* Create account: A user can create account through registration process.
* Login: After creating account user can login through their details and can access to this site.
* Surfing: After login user can surf different museum and can book tickets.
* Payment: After adding booking details, user can proceed for final payment.

## **2.2. Literature Review**

Literature review is a self-contained unit in a study which analysis critically a segment of published body of knowledge through summary, classification and comparison of research studies and theoretical articles. It is an act of reviewing and examining the articles of other on the related topic. There are numbers of studies that is carried on the online booking system by different authorities, authors and professors[1]. Few of the literature of different persons around the globe are considered here:

Gojek is one of the transportation applications most popular online in Indonesia. This app is based on cloud technology (Abdillah et al., 2017) which offers a wide selection of services for its users. Gojek as a company “ride-hailing” is the “top number one Unicorns” in Indonesia is followed by the "Tokopedia" market place, travel site “Traveloka”, market place “Bukalapak”, and what has just emerged is financial technology (FinTech) “OVO”.[2]

Among the smartphone users, millennial users are dominant. Almost every country, the millennial generation is much more may be an internet and smartphone user compared to those aged 35 and older (Poushter, 2016).

According to the U.S. Survey Department, manufacturing sector is the largest supplier to e-commerce sales which has 47.4% of their total shipments, followed by vendors which is having 28.6% of their total sales. These two sectors make the business-to-business groups. Electronic commerce is generally considered to be the sales feature of e-business. It also consists of the exchange of data to facilitate the financing and payment aspects of business transactions. This is an active and resourceful way of communicating within an organization and one of the most operative and useful ways of leading business.[3]

For educational level, higher educated consumers are more likely to use the internet for their shopping medium because they are more computer literate (Burke,2002)[4]

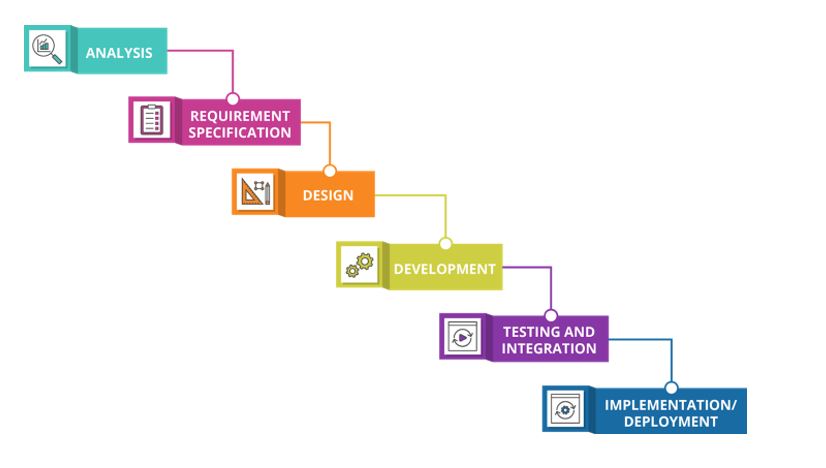
Sorooshian, Onn and Yeen (2013) further define e-ticketing as “a procedure of keeping record of sales, usage tracking and accounting for a passenger’s transport with no requirement for a paper ‘value document’” (p. 63). This definition clearly indicates that the e-ticket includes more than just a paperless document for the passenger: rather the e-ticket represents an extensive architecture within the organization that provides a wealth of information about the consumer.[5]

The definitions of e-tickets and e-ticketing provided in the literature clearly suggest that e-ticketing has a host of implications. **Lubeck, Wittmann and Battistella (2012)** able to examine these issues by tracing the evolution of e-tickets and efforts by the organization to improve efficiency in ticketing operations. According to these authors, e-tickets have evolved to address concerns associated with “inefficiency in information management and control of operations” (p. 18). E-tickets, as noted by Lubeck and co-workers, require the creation of a comprehensive technological platform that controls almost every aspect of the customer relationship within the organization. As such, the roots of e-ticketing go much further than the interface with the customer.[6]

# **CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

## **3.1. System Analysis**

The process flow for this museum ticket booking system includes analysis of the requirements, design, development, testing, and maintenance. All of the functional and non-functional needs are examined throughout the requirement analysis process, and the system is then designed according the requirements. In testing phase; if the testing is successful, the system is installed otherwise, some maintenance is required before the system can be used. As a software development model, the software employs the waterfall approach.



***Fig 3.1: Waterfall methodology for Museum ticket booking site.***

### **3.1.1. Requirement Analysis**

Requirement analysis is performed during the development of a system, and it is necessary to analyze the entire system requirement before implementing it.

### **Functional requirements:**

A functional requirement defines a system or component's function, where a function is defined as a specification of behavior between inputs and outputs.

### **Admin:**

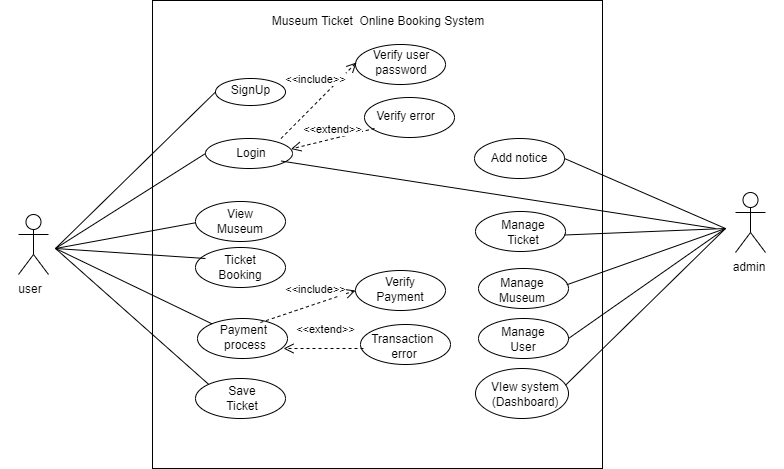
* Admin can sign up for the account.
* The software shall enable the admin to create and manage records.
* The software shall allow admin to change the price of tickets.
* The software shall enable admin to add new museums.
* The system shall be able to save visitor information.

### **User:**

* The user has the ability to sign up for and login to the system.
* Only users with a valid id and password should be permitted to access the system.
* The user should be able to search for the museum of their choice.
* Users should be able to book their own tickets.
* Users can make payments online.

## **Use Case Diagram:**

Our system allows two user privileges’ i.e., user and admin. In the above point, we discussed the various activities that users and administrators can perform in our system. The Use Case diagram below illustrates this in a graphical format.

****

***Fig3.2: Use case diagram for Museum ticket booking system.***

### **Non-functional Requirements:**

Non-functional requirements are evaluated using following requirements:

### **- Security Requirements:**

Only administrators and other members of the management committee have access to visitor system details and records in this system. This system is extremely secure because it can only be operated by authorized committee members.

### **- Reliability Requirement**

This system efficiently handles booking tasks. User registration, login, and payment were all completed correctly, making this system dependable.

### **- Accuracy:**

This system handles booking tasks effectively. This system is reliable because user registration, login, and payment were all completed correctly.

### **- Availability Requirement**

The system is completely available to the user and is used 24 hours a day. The system must be operational 24 hours a day, seven days a week.

### **- Usability Requirement**

As an interface, the system makes use of a web browser. Because all users are familiar with the general use of browsers, no special training is required. The system is easy to use, which makes it even easier.

### **3.1.2. Feasibility Analysis**

The findings of feasibility analysis are used to determine whether or not to proceed with the project. In short, a feasibility analysis evaluates the project's potential for success. Prior to working on the project, the following feasibility analysis was performed:

### **I. Technical Feasibility:**

The Web application is technically feasible and meets current technology standards. The museum ticket booking system is compatible with almost all modern web browsers and, more importantly, can run on any modern operating system. As a result, the proposed system is technically feasible.

### **II.** **Operational feasibility:**

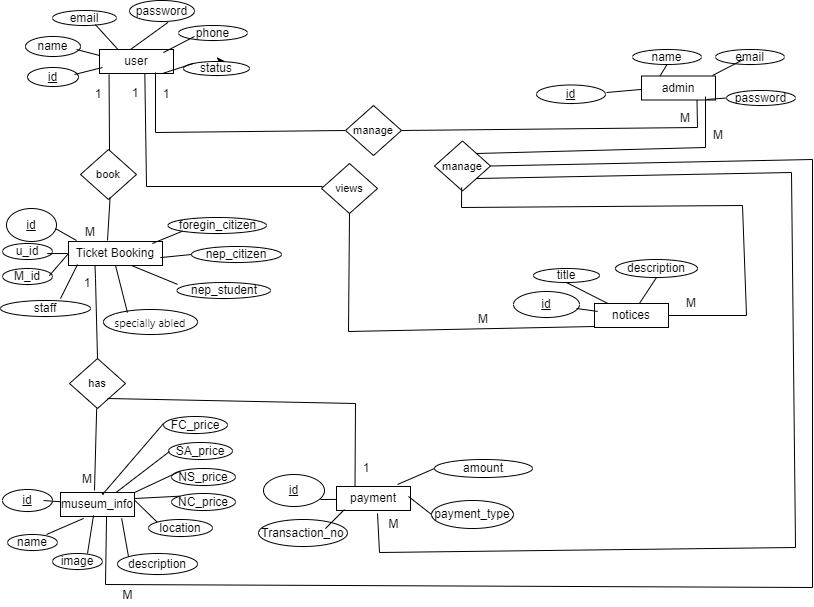
This system meets all of the requirements for a ticket booking system. It is fully operational and can be successfully implemented, and is easy to use due to its user-friendliness. It does not require any special skills to operate; even inexperienced computer users can use it.

### **III. Economic feasibility:**

The system is economically feasible as well as cost effective. Because all of the necessary tools and resources are open source or free. The organization did not need to deploy any new hardware or software after the system is in use.

### **3.1.3. Data Modeling (ER-Diagram)**

An entity relationship diagram depicts the relationships between entities in a database, such as people, things, or concepts. An ERD will also depict the characteristics of these entities. The ER diagram below includes six entities, three relationships, and thirty five attributes.

****

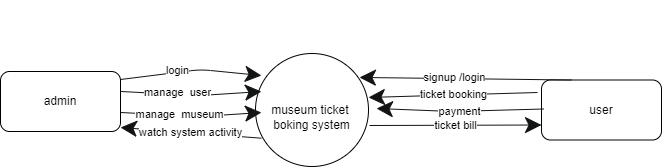
***Fig3.3: ER-diagram of museum ticket booking system***

### **3.1.4. Process Modeling(DFD)**

A data-flow diagram depicts the flow of data through a process or system. The DFD also contains information about each entity's and the process's outputs and inputs. This system is demonstrated using 0 and 1 level DFD.[7]

**0 level DFD:**

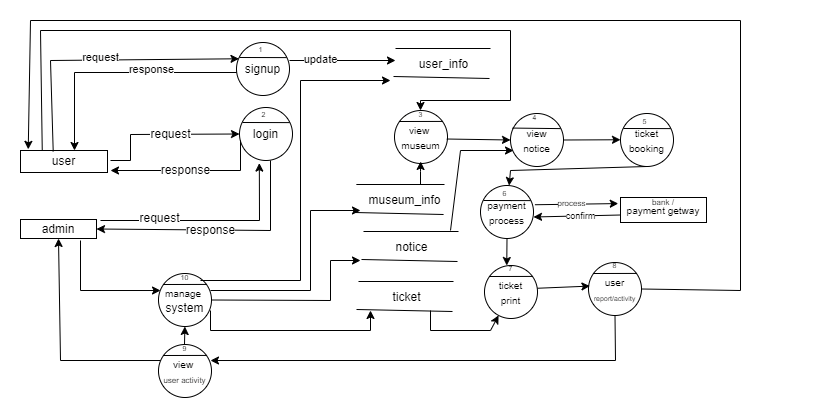
DFD Level 0 of the Museum Ticket Booking System is a basic overview of the entire system or process being analyzed or modeled. DFD 0 displays the main (primary function) process and the activities of users and administrators in the system. It helps the developer identify the primary function (process) in the Museum booking System.

****

***Fig 3.4: 0 level DFD of museum ticket booking system***

**1 level DFD:**

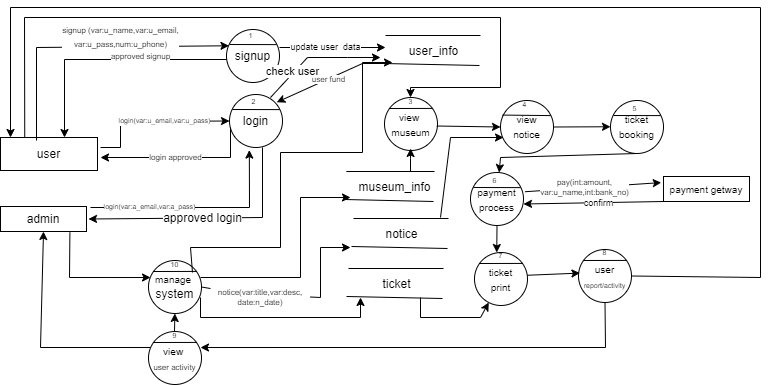
The DFD level 1 of the Museum Ticket Booking System is a detailed breakdown of pieces of the Context Level Diagram. The diagram in DFD level 1 also shows more information about how processing works and the database used. Every process has a process number that indicates how the system works.

****

***Fig 3.5: 1 level DFD of museum ticket booking system***

***p***

**Physical DFD 1**

****

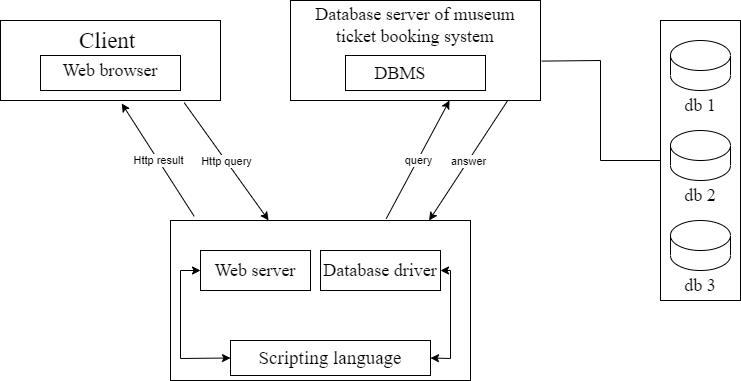
***Fig 3.5: Physical DFD of museum ticket booking system***

## **3.2: System Design:**

Different design diagrams of the museum ticket booking system have been created in order to graphically represent the various functional requirements of the system, and they are as follows:

### **3.2.1. Architectural Design:**

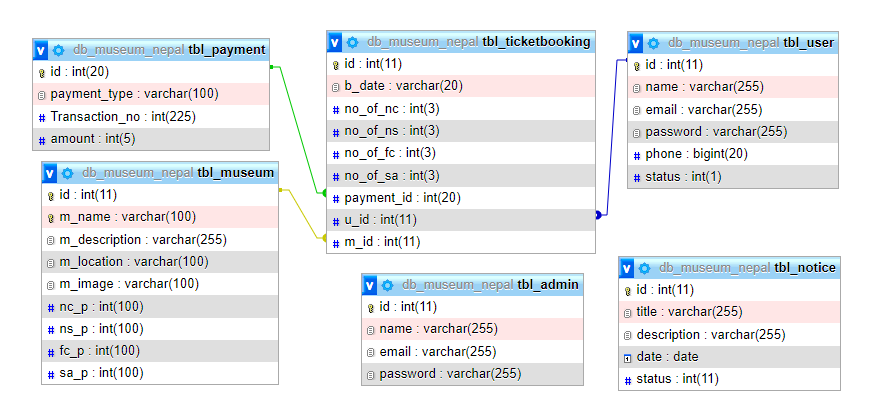
Users interact with the system via a simple user interface on the museum ticket booking website. The architecture of the museum ticket booking websites is three tiers.



***Fig 3.6: 3-tier architecture of museum ticket system.***

### **3.2.2. Database Schema Design**

The database schema of Museum Ticket Booking System is the skeleton structure that represents the logical view of the complete database. It specifies how data is arranged and how relationships between them are defined. It defines all of the constraints that will be applied to the data. The database schema is a structure of a database described in a formal language supported by the database management system (DBMS).

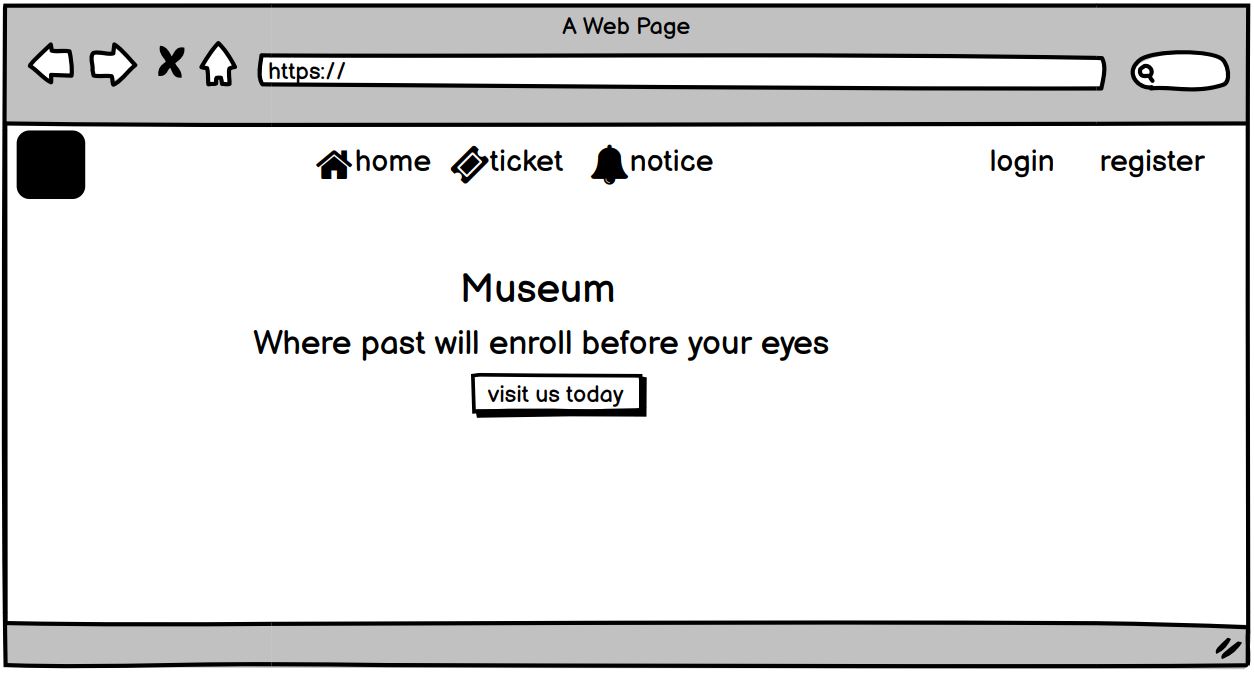


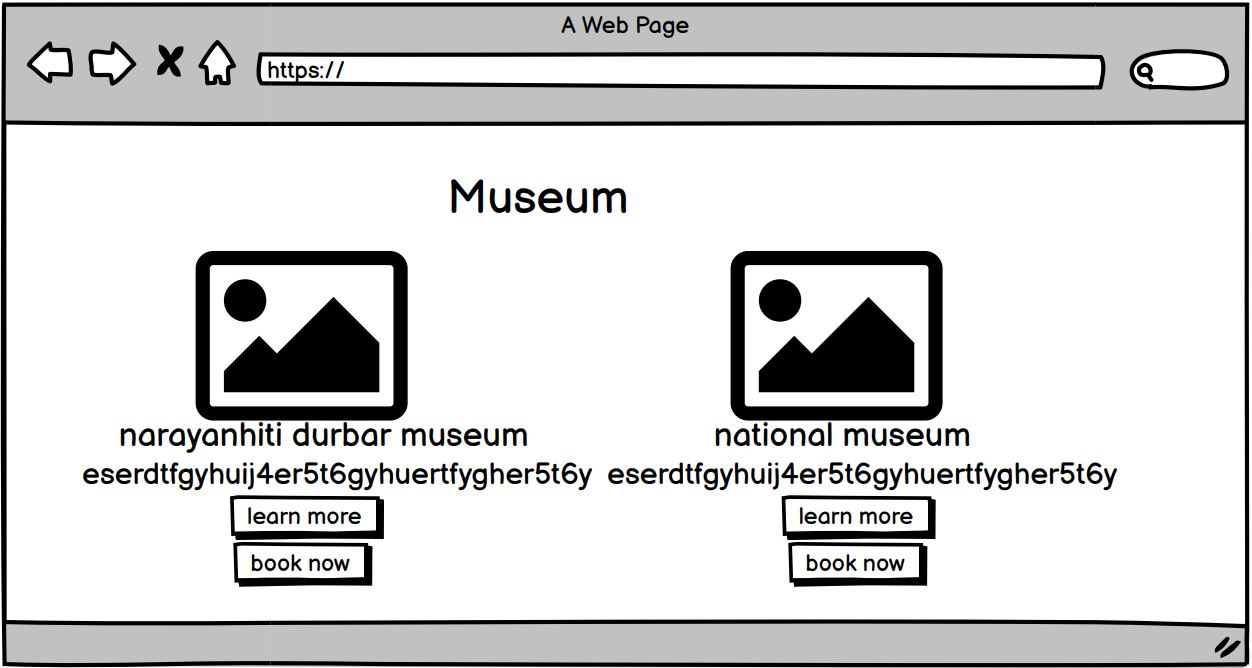
***Fig 3.7: Database schema of museum ticket booking system.***

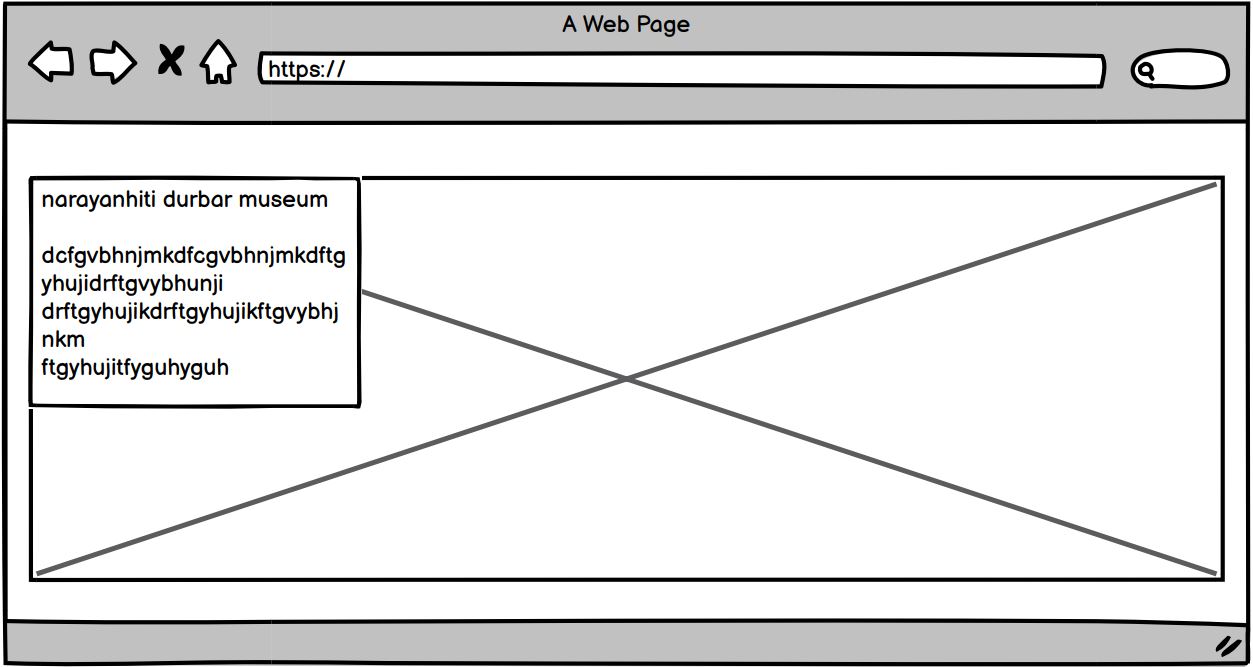
### **3.2.3. Interface Design**

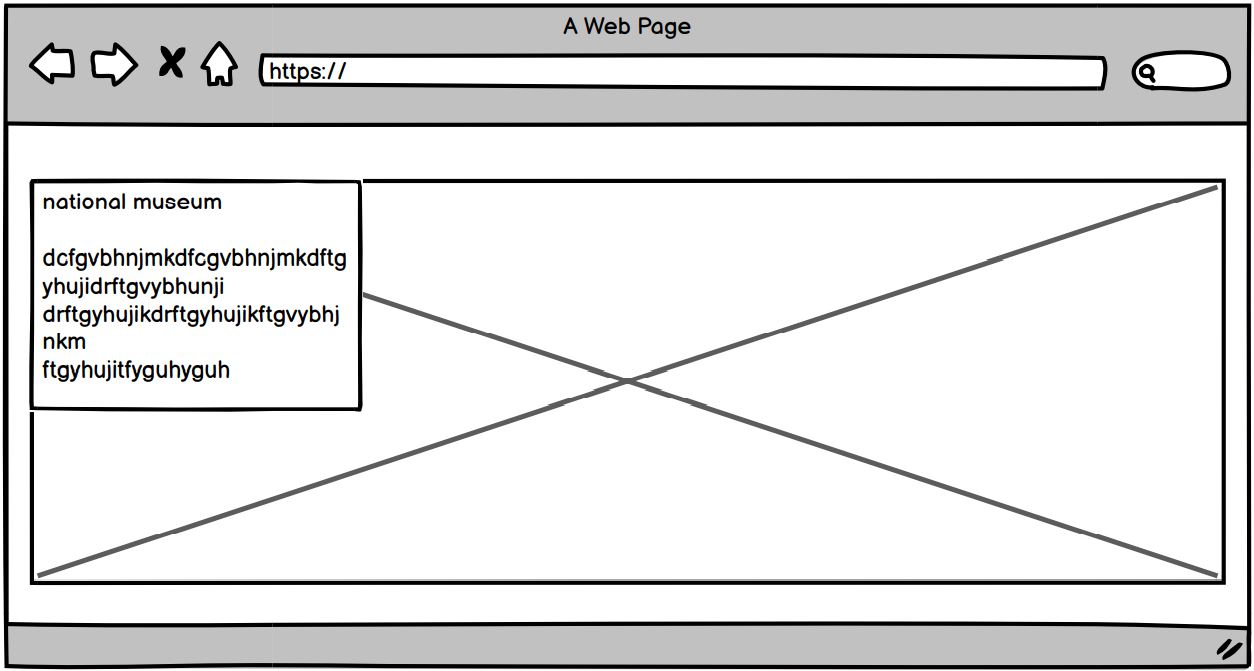
Different tool is used to create layout of museum ticket booking system. In this balsamiq is used to design the system wireframe/layout.

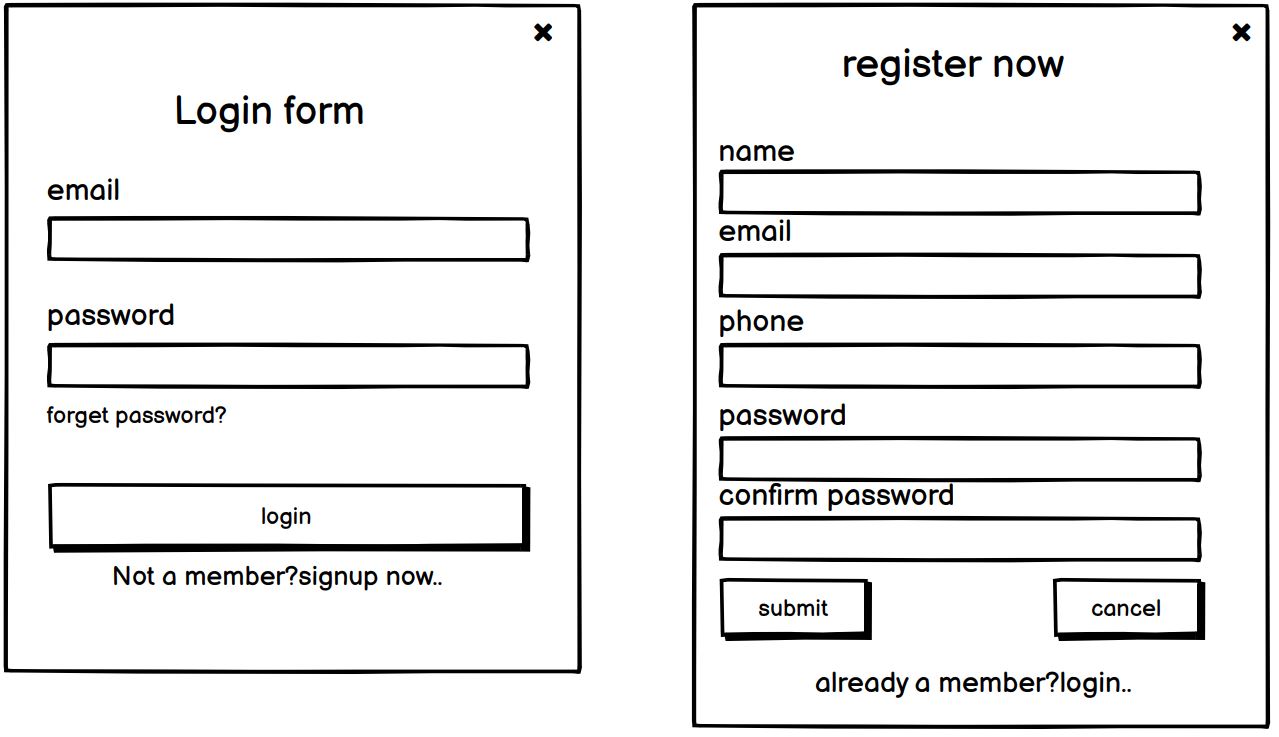
### **User Pannel**

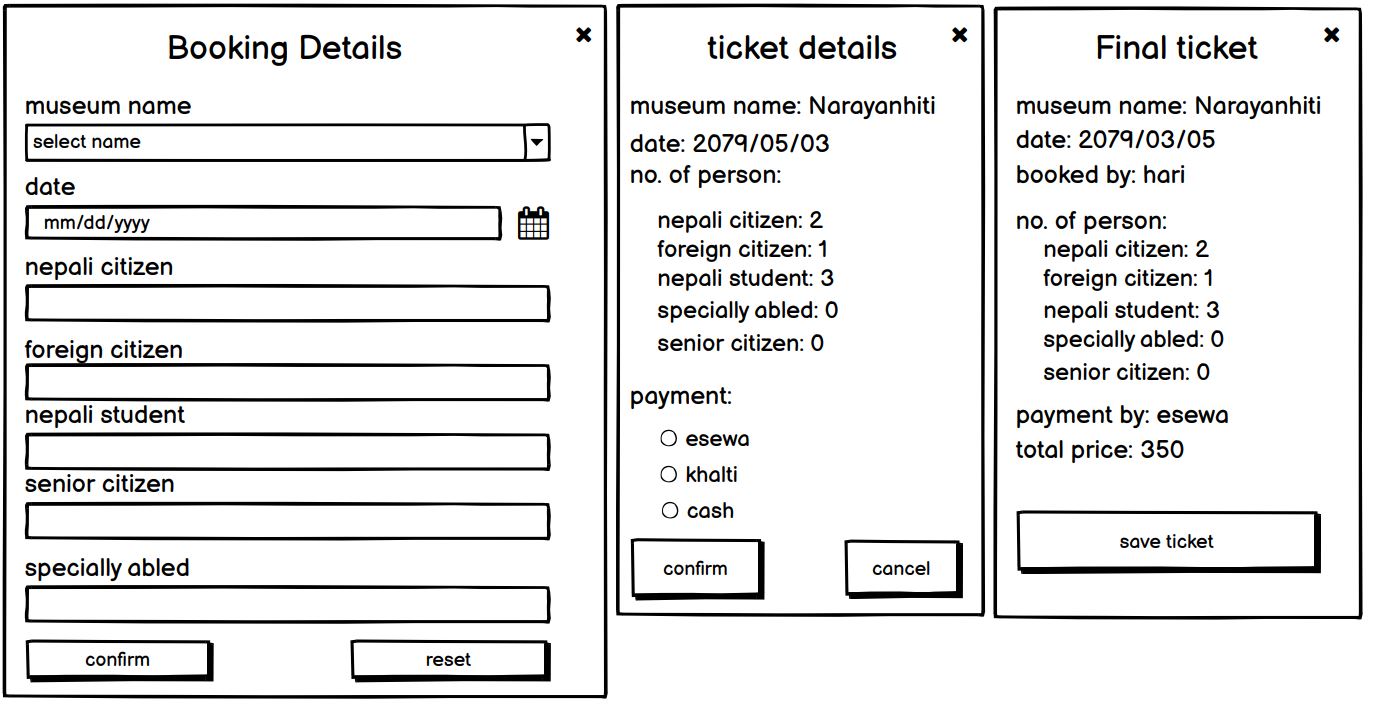


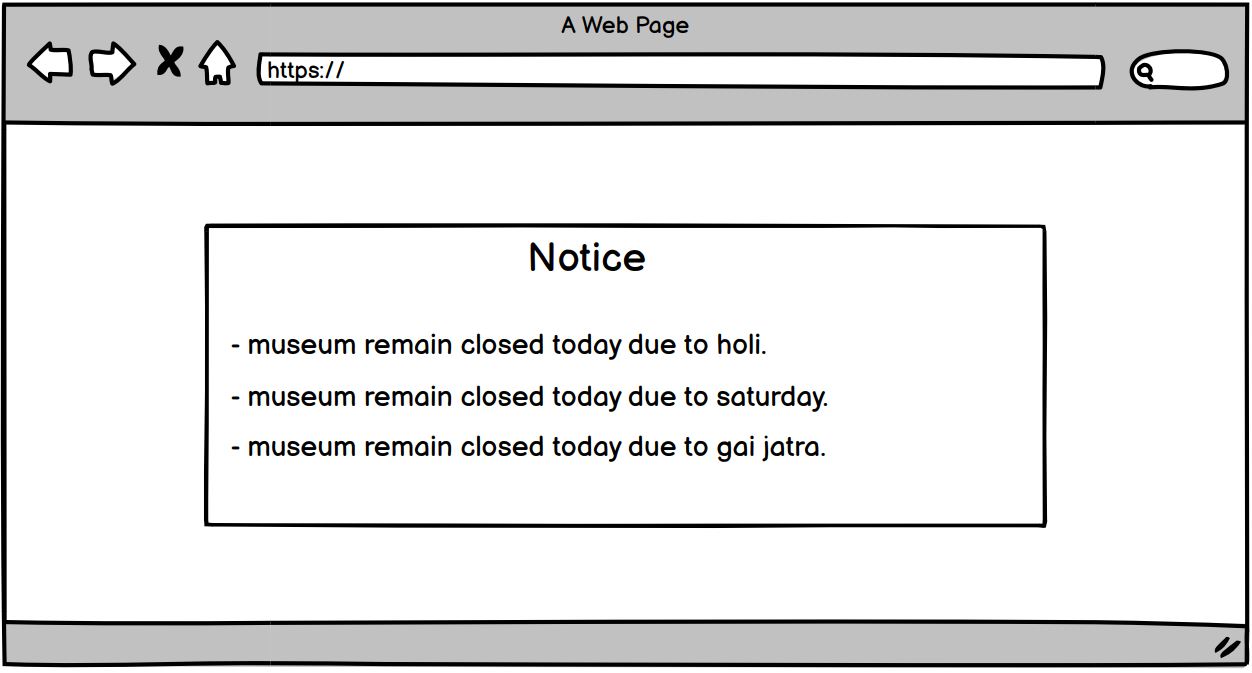




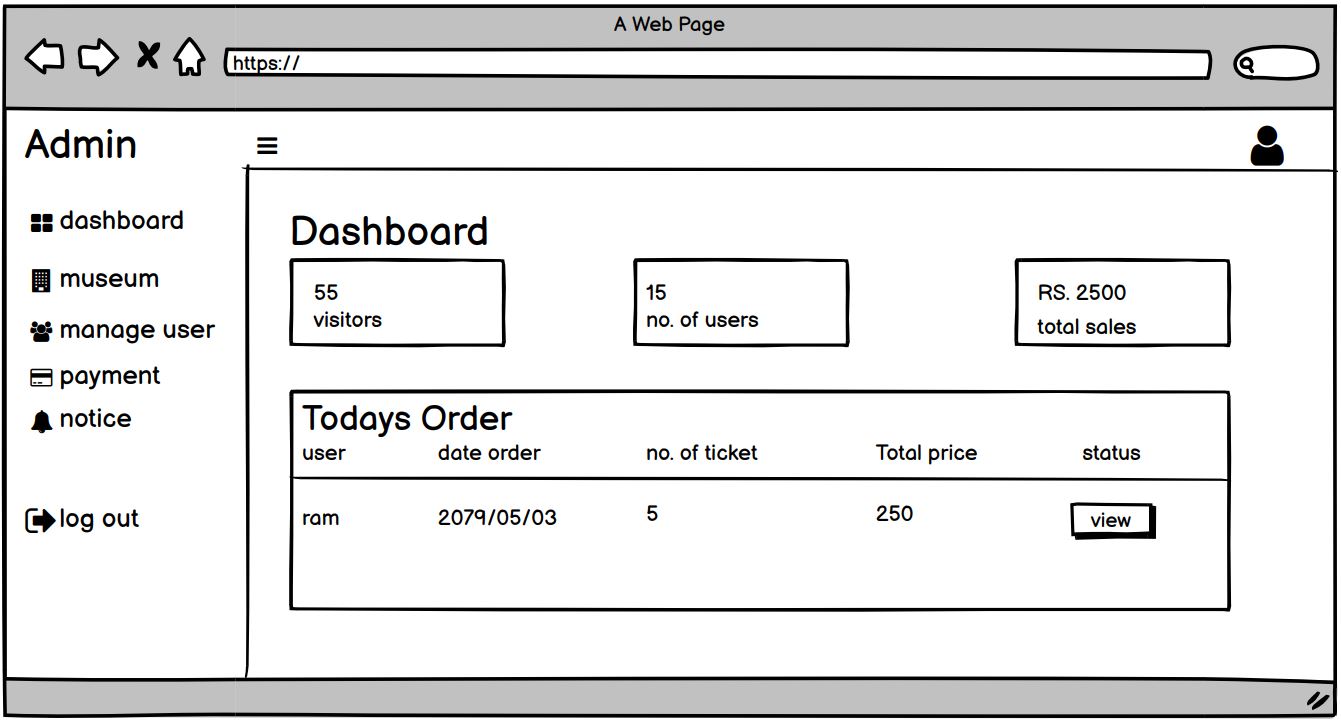


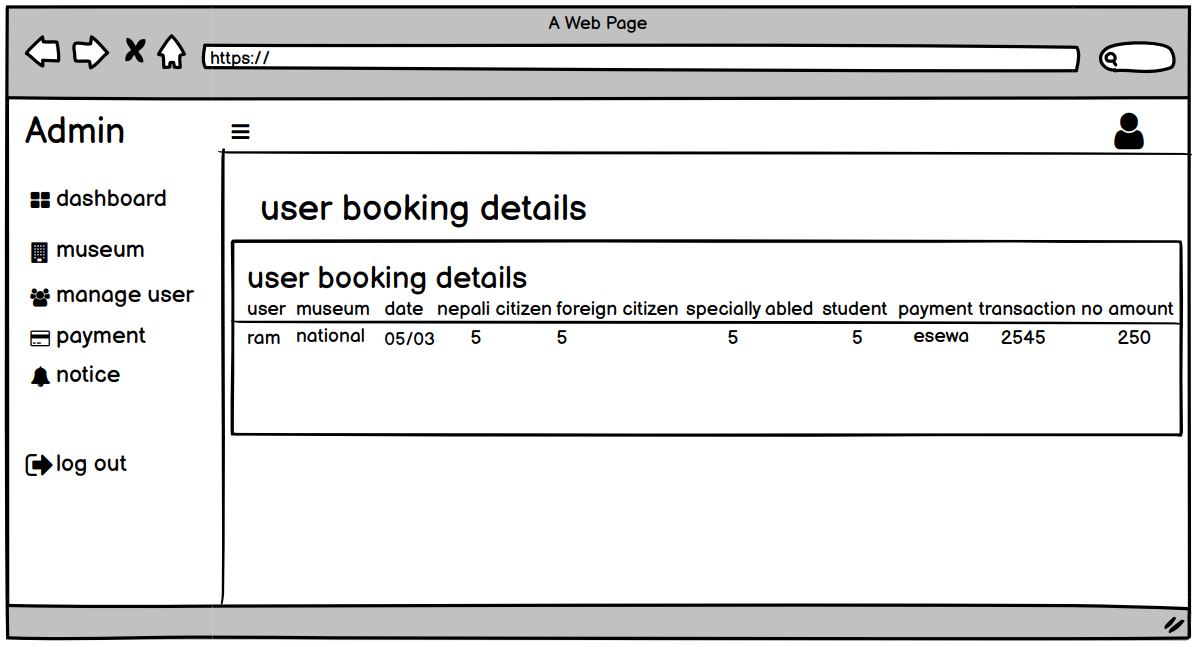


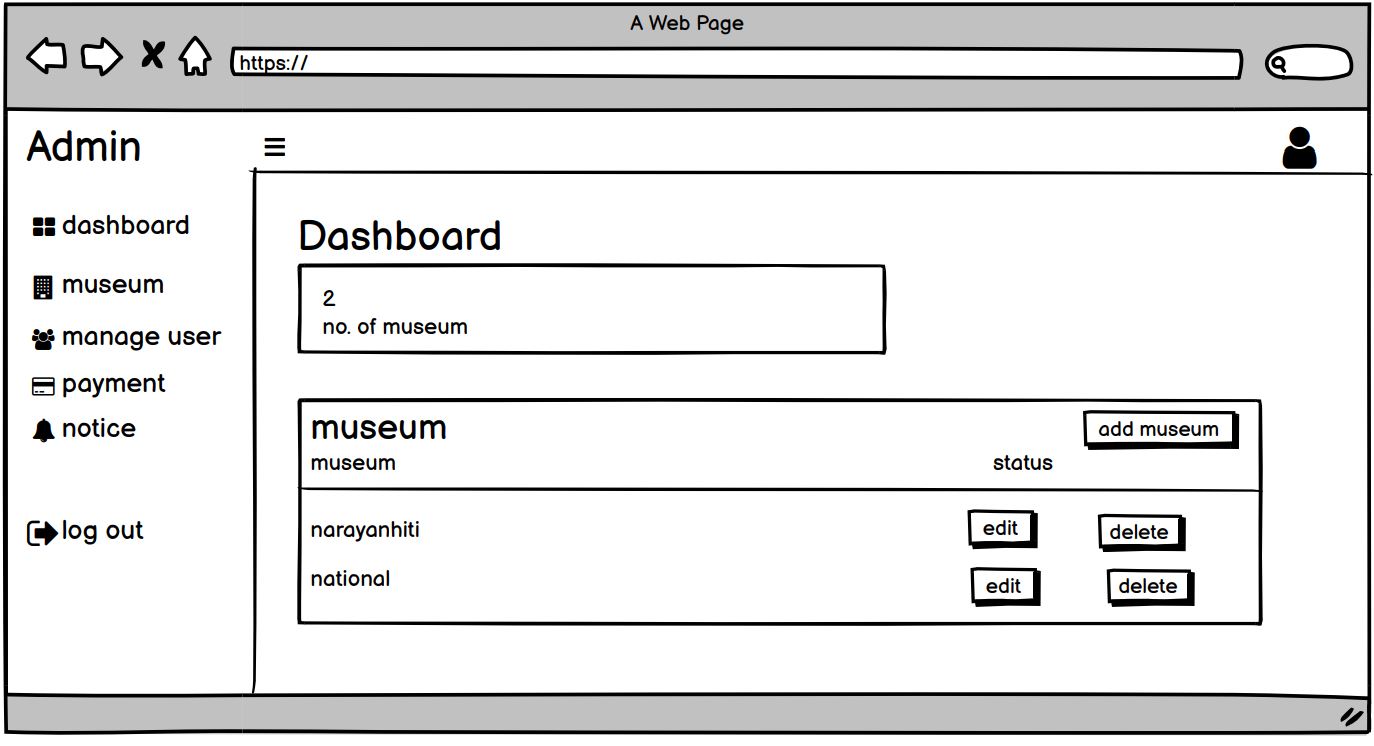


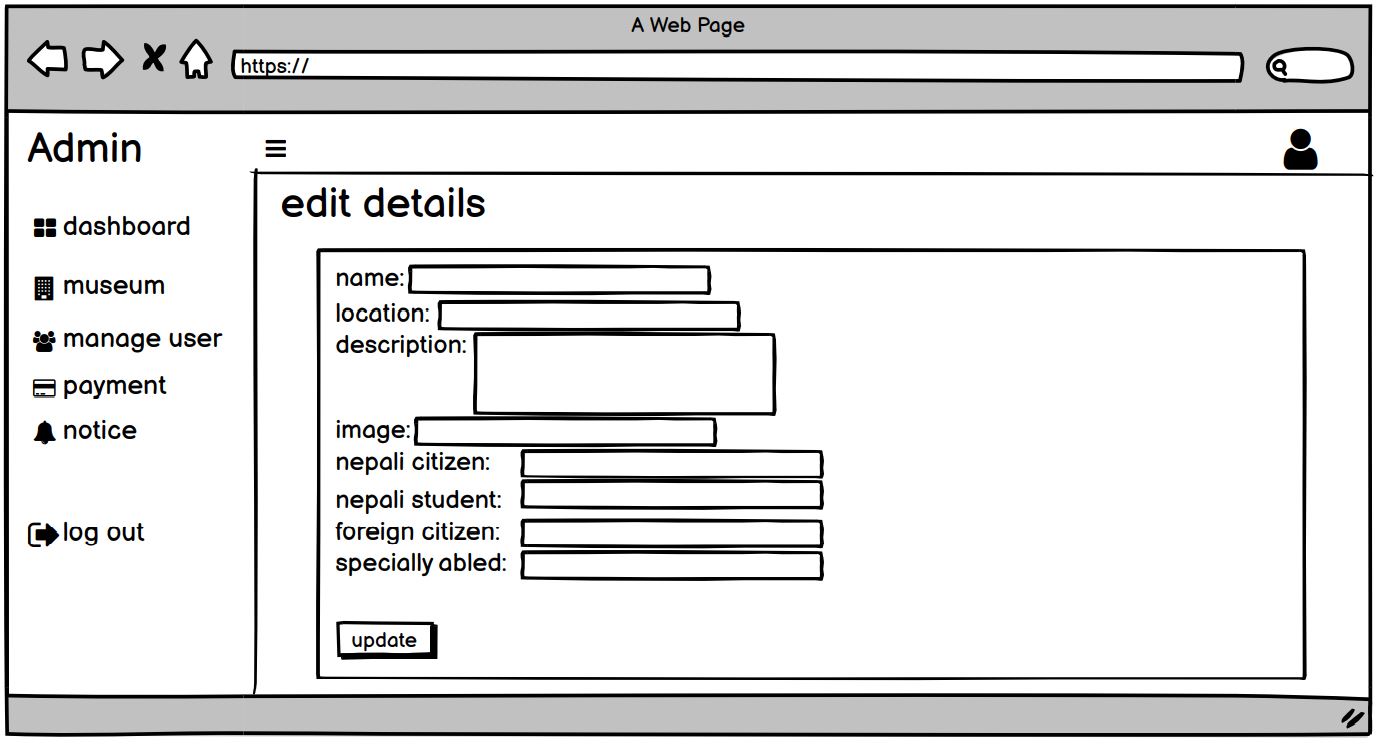


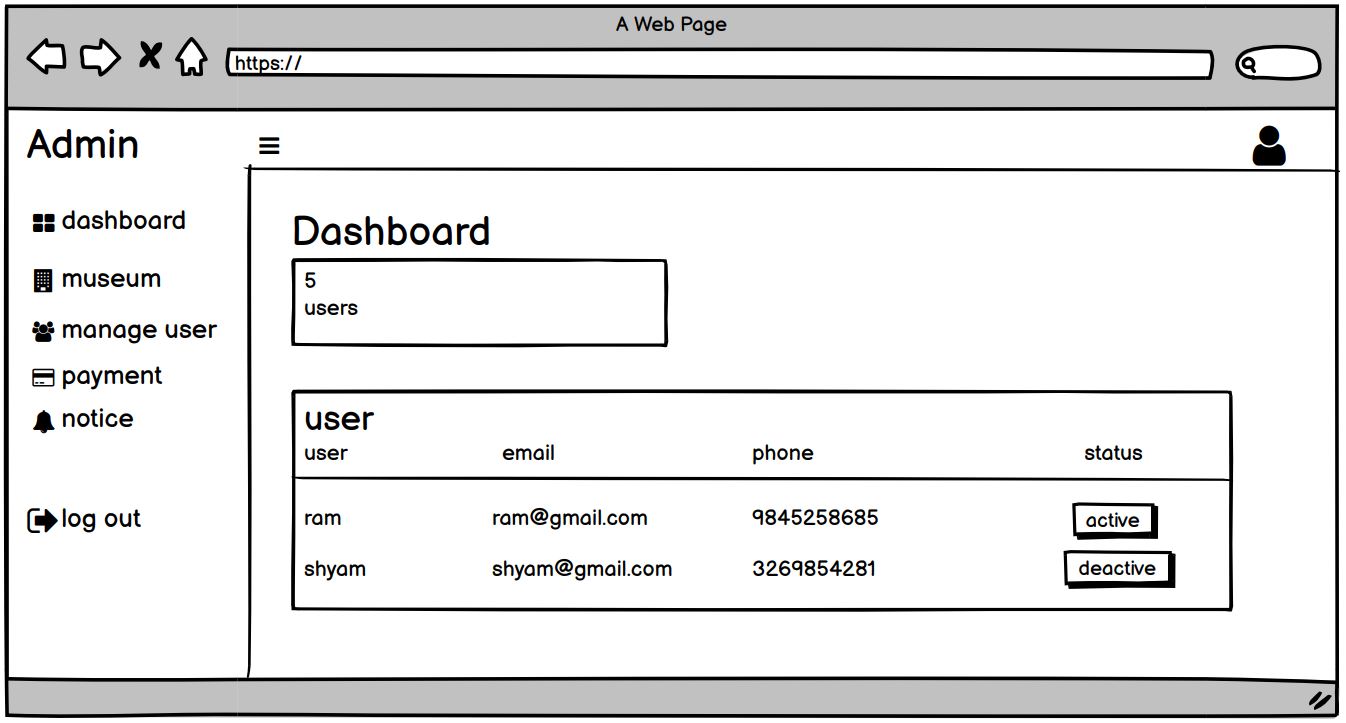
### Admin Pannel

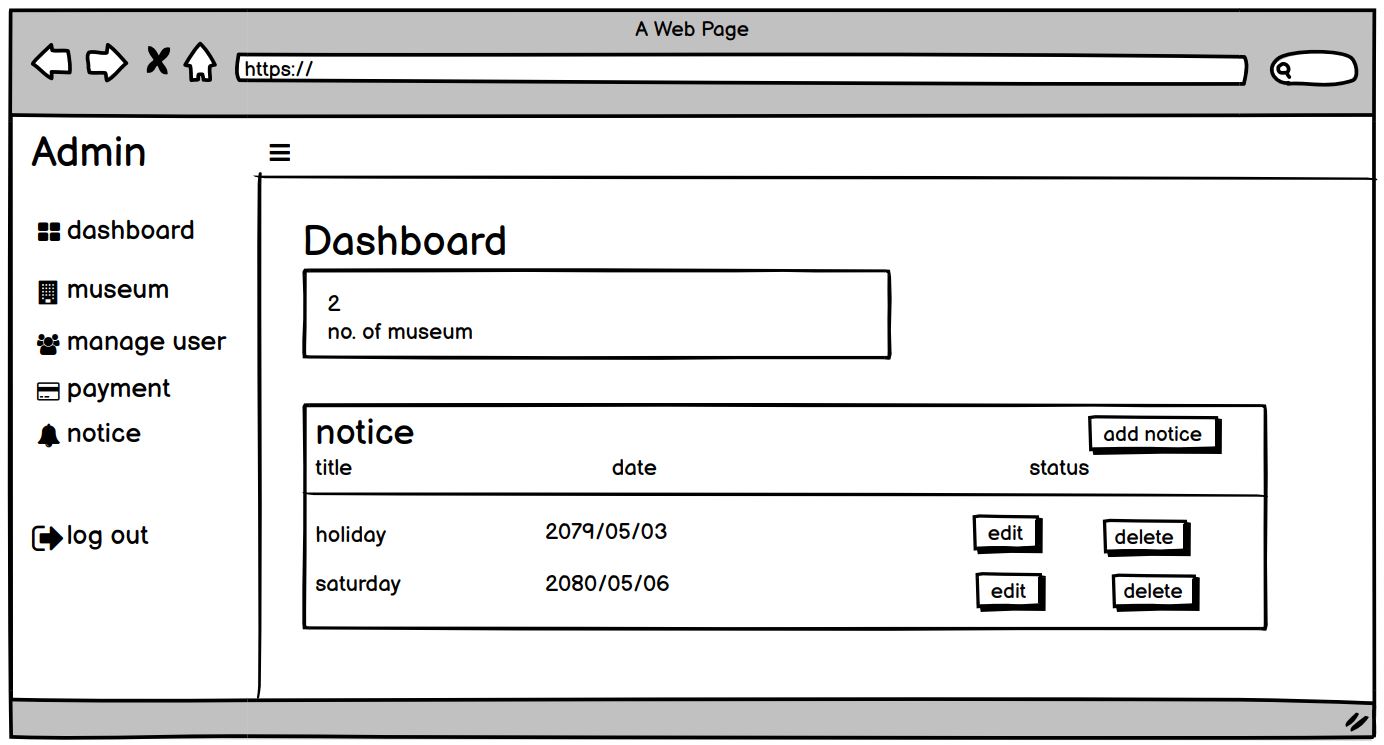


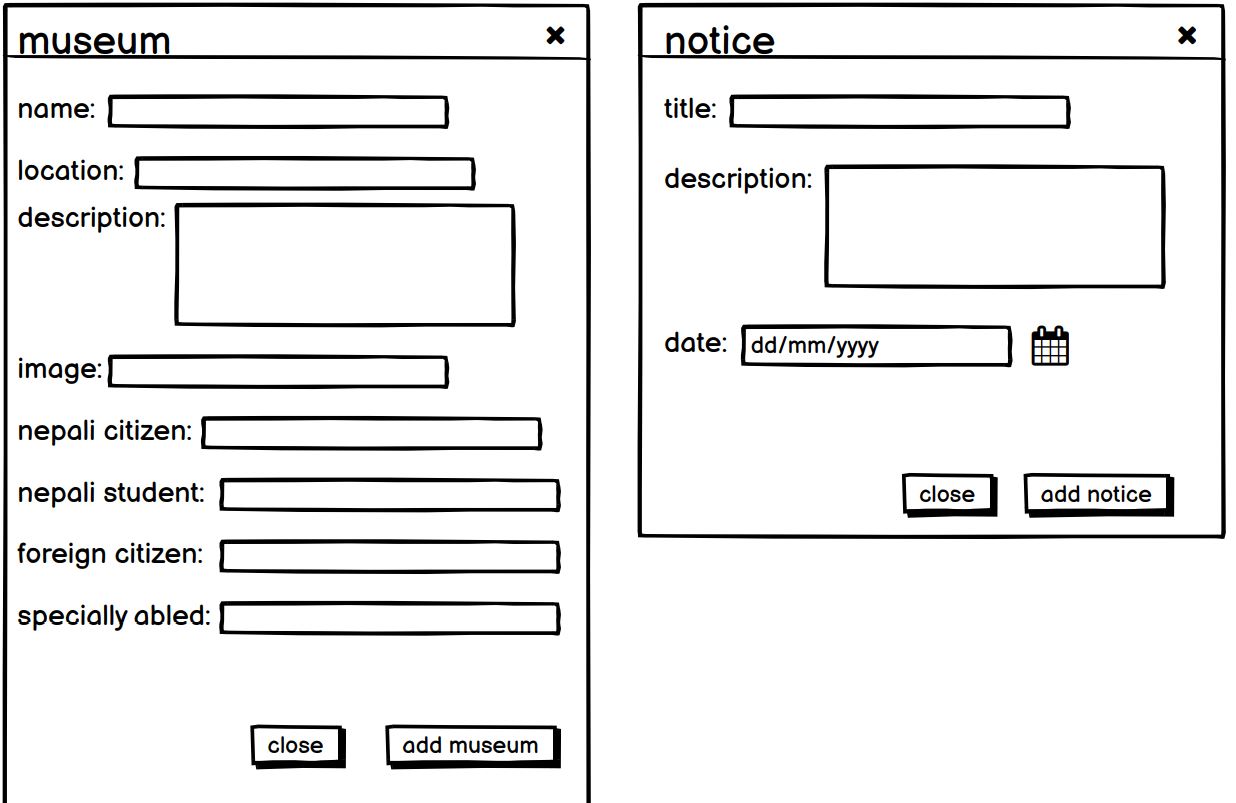












# **CHAPTER 4: IMPLEMENTATION AND TESTING**

## **4.1: Implementation**

Implementation is the phase in which the system is actually built. We studied and analyzed all of the information we gathered before putting a system in place for users. Coding, testing, installation, and documentation are all part of the implementation process. Several tools are used in the development of this system, which is discussed further below.

### **4.1.1: Tools used:**

Different front-end and backend tools are used in development of the system.

### **Visual Studio Code:**

In the project, Visual Studio Code was used as the main code editor to write the codes for the platform. We chose Visual Studio Code as the primary code editor for the project as it is an open-source software with a user-friendly interface and support for multiple programing languages and added functionality like debugging, syntax, highlighting, intelligent code completion, snippets, code refactoring embedded GIT etc.[8]

### **Balsamiq:**

Balsamiq is the open source designing tools. It is used to create wireframe design of museum ticket booking system.

### **Front End:**

The frontend section is built with HTML, CSS, Bootstrap and JavaScript.

### **HTML**:

HTML is a markup language used to format text documents on the web. It is used to design the layout of a web page.

### **CSS:**

CSS3 is a style sheet language that is used to style the basic layout created by HTML5. It aids in the creation of more appealing and beautiful layouts.

### **Bootstrap:**

Bootstrap is used to style and make responsive websites.

### **JavaScript:**

JavaScript is a client-side programming language that enables us to add dynamic behavior to our webpages. It enables client-side scripts to interact with users and control browsers. It is also used to validate various forms.

### **Back End:**

The backend is implemented using PHP.

### **PHP:**

PHP is a server-side scripting language that is used for web development but can also be used for general-purpose programming. PHP scripts are executed on the server and aid in database interaction. It is appropriate for web development because it is easily embedded in HTML.

### **Database:**

My SQL is used as database tool.

### **My SQL:**

My SQL is used to store data in database.

### **4.1.2. Implementation Details of Modules**

A module is a high-level description of a functional area that consists of a group of processes that describe the module's functionality. The proposed system consists of several modules, including user modules, admin modules, login modules, and booking modules.

### **User Module:**

After entering to the website, user have to login or signup to the system. To create a user account, the user must fill out a form that includes fields such as Username and Password. If the details entered while filling out the form are identical to those already in the database, the form will be submitted and if the details entered by user doesn’t match in database then it throws error. When the form is successfully submitted, all of the information is saved in the database. After that, the user can access the system whenever they want.

### **Login Module:**

After successfully creating an account, the user can log in to the system via the login module. The login module has two fields: username and password. When the username and password entered by the user match the database username and password, the user is logged in. To access the system, the user must enter the correct username and password.

**Code:** if (count($err) == 0) {

        // check login info  from database:

        try {

            $db\_name = 'db\_museum\_nepal';

            $db\_host = 'localhost';

            $db\_username = 'root';

            $db\_password = '';

            $con = mysqli\_connect($db\_host, $db\_username, $db\_password, $db\_name);

            $login\_sql = "select \* from tbl\_user where  email='" . $email . "' and password='" . $password . "'";

            $login\_check =  mysqli\_query($con, $login\_sql);

            if (mysqli\_num\_rows($login\_check) == 1) {

                $users = mysqli\_fetch\_array($login\_check);

                @session\_start();

                $\_SESSION["login"] = 1;

                $\_SESSION['id'] = $users['id'];

                $\_SESSION['name'] = $users['name'];

                $\_SESSION['email'] = $users['email'];

                header('location:homepage.php');

            } else {

                $err['login'] = 'No users found';}}

catch (Exception $e) {

            die('Database connection error' . '<br>' . $e->getMessage());

        }

    }

### **Signup Module:**

User can signup/register to the system through signup module. The signup module has 5 fields i.e. Name, Email, Phone, Password, Confirm Password. User should fill this all required fields to register to the system. After entering all these data users can now login to the system and use the system.

**Code:**  if (count($err) == 0) {

        try {

            $db\_name = 'db\_museum\_nepal';

            $db\_host = 'localhost';

            $db\_username = 'root';

            $db\_password = '';

            $con = mysqli\_connect($db\_host, $db\_username, $db\_password, $db\_name);

            $registration\_sql =

                "insert into tbl\_user(name,email,password,phone,status)values( '" . $name . "','" . $email . "','" . $password . "','" . $phone . "',1)";

            $registration\_query = mysqli\_query($con, $registration\_sql);

            // print\_r($registration\_sql);

            // print\_r($registration\_query);

            header("location:index.php");

        } catch (Exception $e) {

            die('database connection error' . '<br>' . $e->getmessage());

        }

    }

### **Booking Module:**

The main function of this system is booking. To book, the user must be logged in to the system and book the museum. To book a ticket to a desired museum, they must enter the requested information and click the confirm button.

**Code:**  if (count($err) == 0) {

        try {

            $b\_date = date("y/m/d");

            $NS = intval($\_POST["NS"]);

            $NC = intval($\_POST["NC"]);

            $FC = intval($\_POST["FC"]);

            $SA = intval($\_POST["SA"]);

            //  fetch museum price list

            $sql\_museum\_price = " select nc\_p,ns\_p,fc\_p, sa\_p from tbl\_museum where m\_name= '$MN';";

            $query\_mueum\_price = mysqli\_query($con, $sql\_museum\_price);

            $row\_museum\_price = mysqli\_fetch\_array($query\_mueum\_price);

            $P\_NS = $NS \* $row\_museum\_price['ns\_p'];

            $P\_NC = $NC \*  $row\_museum\_price['nc\_p'];

            $P\_FC = $FC \* $row\_museum\_price['fc\_p'];

            $P\_SA = $SA \* $row\_museum\_price['sa\_p'];

            $TP = $P\_NC + $P\_NS + $P\_FC;

            @session\_start();

            $\_SESSION["MN"] = $MN;

            $\_SESSION["date"] = $date;

            $\_SESSION["NC"] = $NC;

            $\_SESSION["NS"] = $NS;

            $\_SESSION["FC"] = $FC;

            $\_SESSION["SA"] = $SA;

            $\_SESSION["TP"] = $TP;

        } catch (Exception $e) {

            die("$e");

        }

    }

### **Payment Module:**

When booking tickets, after entering the requested information, a payment option will be displayed. For online payment, the user can select any of the desired payment options.

**Code:**  if (count($err) == 0) {

        try {

            $\_SESSION["payment\_type"] = $payment\_type;

            $\_SESSION["payment\_no"] = $payment\_id;

            $tbl\_booking\_sql = 'INSERT INTO `tbl\_ticketbooking` (`id`, `u\_name`, `m\_name`, `b\_date`, `no\_of\_nc`, `no\_of\_ns`, `no\_of\_fc`, `no\_of\_sa`, `payment\_type`, `transction\_no`, `amount`) VALUES (NULL, "' . $\_SESSION["name"] . '","' . $\_SESSION["MN"] . '","' . $\_SESSION["date"] . '","' . $\_SESSION["NC"] . '", "' . $\_SESSION["NS"] . '","' . $\_SESSION["FC"] . '","' . $\_SESSION["SA"] . '","' . $\_SESSION["payment\_type"] . '","' . $\_SESSION["payment\_no"] . '","' . $\_SESSION["TP"] . '")';

            $tbl\_booking\_query = mysqli\_query($con, $tbl\_booking\_sql);

            if ($tbl\_booking\_query) {

                echo "inserted";

            } else {

                echo "false";

            }

        } catch (Exception $e) {

            die("Connection Error" . $e->getMessage());

        };

    }

### **Admin Module:**

Authentication in the admin module is done using the admin's email and password. If the admin enters the correct email and password, he will be able to access his dashboard. The system is managed by the administrator, who has access to a dashboard from which he or she can manage users, view booking details, edit notices, add and delete museums, and so on.

**Admin login:**  $sql = "select \*from tbl\_admin where email='$username' and password='$password';";

        $res = mysqli\_query($conn, $sql);

        if ($res) {

            echo "true";

            $row = mysqli\_fetch\_assoc($res);

            $id = $row["id"];

            $name = $row["name"];

            echo $id;

            echo $name;

            session\_start();

            $\_SESSION["id"] = $id;

            $\_SESSION["name"] = $name;

            header("location:admin.php");

        }

**Manage user:**  $res=mysqli\_query($conn, "SELECT \* FROM `tbl\_user`");

                    $check\_user=mysqli\_num\_rows($res)>0;

                    if($check\_user)

                    {

                        while($row=mysqli\_fetch\_array($res)){

                            $id = $row['id'];

                    ?>

                    <tr>

                        <td>

                            <?php echo $row['name']; ?>

                        </td>

                        <td>

                            <?php echo $row['email']; ?>

                        </td>

                        <td>

                            <?php echo $row['phone']; ?>

                        </td>

                        <td>

                            <?php

                            if($row['status']==1){

                                echo '<p><a type="button" class="button" href="status.php?id='.$row['id'].'&status=0" >Active</a></p>';

                            }

                            else{

                                echo '<p><a  type="button" href="status.php?id='.$row['id'].'&status=1" <span style="background:red;" class="button">Deactive</a></p>';

                            }

                            ?>

                        </td>

                    </tr>

                    <?PHP

                    }

                    } else {

                        echo "no user found";

                    }

**Edit museum:** if(isset($\_POST['submit']))

{

    $id = $\_GET['id'];

    $name=$\_POST['name'];

    $location =$\_POST["location"];

    $desc=$\_POST['description'];

    $filename = $\_FILES["image"]["name"];

    $tempname = $\_FILES["image"]["tmp\_name"];

    $folder = "muesumImages/".$filename;

    $nc=$\_POST["nc"];

    $ns=$\_POST["ns"];

    $fc=$\_POST["fc"];

    $sa=$\_POST["sa"];

    if(move\_uploaded\_file($tempname,$folder))

    {

        echo "file upload successfully";

    }else{

        ?>

                <script>

                    alert('Your Images hasnot selected')

                </script>

        <?php

    }

    $sql="UPDATE `tbl\_museum` SET `m\_name` = '$name', `m\_description` = ' $desc', `m\_location` = '$location', `m\_image` = '$filename ', `nc\_p` = ' $nc', `ns\_p` = '$ns', `fc\_p` = '$fc', `sa\_p` = '$sa' WHERE `tbl\_museum`.`id` = $id;";

    $query\_run = mysqli\_query($conn,$sql);

    if($query\_run){

        header("location:museum.php");

    }

}

**Add museum:** if (isset($\_POST["add"])) {

    if (isset($\_POST['mname'])) {

        $mname = $\_POST["mname"];

        echo $mname;

        echo "<br>";

    } else {

        $err['mname'] = "Enter museum name";

    }

    if (isset($\_POST['mdesc'])) {

        $mdesc = $\_POST["mdesc"];

        echo $mdesc;

        echo "<br>";

    } else {

        $err['mdesc'] = "Enter description";

    }

    if (isset($\_POST['mlocation'])) {

        $mlocation = $\_POST["mlocation"];

        echo $mlocation;

        echo "<br>";

    } else {

        $err['mlocation'] = "Enter  location";

    }

    if (isset($\_FILES['mimage'])) {

        // echo $fileName;

        // $targetFilePath = $targetDir . $fileName;

        // $fileType = pathinfo($targetFilePath, PATHINFO\_EXTENSION);

        $filename = $\_FILES["mimage"]["name"];

        $tempname = $\_FILES["mimage"]["tmp\_name"];

        $folder = "muesumImages/".$filename;

        if(move\_uploaded\_file($tempname,$folder)){

            echo "done";

        }

        echo "<br>";

    } else {

        $err['mimage'] = "Enter  image name";

    }

    if (isset($\_POST['mnc'])) {

        $mnc = $\_POST["mnc"];

        echo $mnc;

        echo "<br>";

    } else {

        $err['mnc'] = "Enter nepali citizen";

    }

    if (isset($\_POST['mns'])) {

        $mns = $\_POST["mns"];

        echo $mns;

        echo "<br>";

    } else {

        $err['ms'] = "Enter nepali student";

    }

    if (isset($\_POST['mfc'])) {

        $mfc = $\_POST["mfc"];

        echo $mnc;

        echo "<br>";

    } else {

        $err['mfc'] = "Enter foreign name";

    }

    if (isset($\_POST['msa'])) {

        $msa = $\_POST["msa"];

        echo $msa;

    } else {

        $err['msa'] = "Enter  sa";

    }

    if (count($err) == 0) {

        $sql="INSERT INTO `tbl\_museum` (`id`, `m\_name`, `m\_description`, `m\_location`, `m\_image`, `nc\_p`, `ns\_p`, `fc\_p`, `sa\_p`) VALUES (NULL, '$mname',' $mdesc', '$mlocation', ' $filename', '$mnc', '$mns', '$mfc', '$msa');";

        $res=mysqli\_query($conn,$sql);

        if($res){

            // echo "insert successfull ";

            header("location:museum.php?msg=1");

        }else{

            // echo " not success";

        }

    }

**Delete museum:** if (isset($\_GET['mid'])) {

    include "\_dbconnect.php";

    session\_start();

        $id= $\_GET['mid'];

        $\_SESSION['mid']=$id;

    $sql="DELETE FROM `tbl\_museum` WHERE `tbl\_museum`.`id` = $id;";

   $res= mysqli\_query($conn,$sql);

   if($res){

       header("location:museum.php");

   }else{

       ?>

            <script>

                alert("not deleted");

            </script>

       <?php

   }

}

## **4.2: Testing**

Testing is done to check the behavior of a complete and fully integrated software product based on the software requirement specification document. There are many types of tests to be carried out on a web application from performance, functionality, database loading time, response time, user’s action and many others. We will not carry out all types of tests for the application considering the time scale to present this project. We will focus the test cases on functionality, security and performance. Some of the types of testing we did are:

### **4.2.1. Test Cases for Unit Testing**

Unit testing is a software development process in which the small part of an application, called units are individually and independently examined for proper operation.

***Table 4.1: Test Table for admin access.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case  ID | Test Scenario | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1 | Check admin access | 1)Enter URL of admin panel  2)Enter username & password | 1)<http://localhost/>  admin/  admin.php  2)username: admin123  Password:1 | To Redirect to admin dashboard | Redirected  to admin dashboard | Pass |

*Table 4.2: Test Table for Register of visitor.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case  ID | Test Scenario | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
| 2 | Check Register | 1)Enter URL of Register  2)Enter Name email  Phone  Password  Confirm password | 1) <http://localhost/>  museum\_nepal\_project/  index.php  2)name: Ram  Email: [ram@gmail.com](mailto:ram@gmail.com)  Phone: 9856421896  password:  123456789 | To Register user. | Redirected  to index page | Pass |

*Table 4.3: Test Table for Login of visitor.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case  ID | Test Scenario | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
| 3 | Check login | 1)Enter URL of login  2)Enter username & password | 1) <http://localhost/>  museum\_nepal\_project/  homepage.php  2)username:  [ram@gmail.com](mailto:ram@gmail.com)  password:  123456789 | To Redirect to user Dashboard | Redirected  to user  panel | Pass |

*Table 4.4: Test Table for Ticket Booking.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case  ID | Test Scenario | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
| 4 | Check Ticket  Booking | 1)Enter URL of booking  2)Enter required field | 1) <http://localhost/>  museum\_nepal\_project/  homepage.php | Ticket booked. | Ticket booked. | Pass |

### **4.2.2 Test Cases for System Testing**

System testing process involved testing the application after development.

*Table 4.5: For adding Museum.*

|  |  |
| --- | --- |
| Test Case 1 | Add museum |
| Test Data | Museum name: Lumbini  Location: bhairahawa  Description: birth place of lord Buddha.  Img:  Entering ticket prices:  Nepali citizen: 200  Nepali student: 50  Foreign citizen: 500  Specially abled: 50 |
| Expected Result | A message should be displayed saying “**Success !** new museum has been added” |
| Test Result | Success ! new museum has been added. |

*Table 4.6: For Editing Museum details.*

|  |  |
| --- | --- |
| Test Case 2 | Edit museum details. |
| Test Data | Museum name: Kathmandu  Location: Kathmandu  Description: located in kathmandu.  Img:  Entering ticket prices:  Nepali citizen: 100  Nepali student: 50  Foreign citizen: 300  Specially abled: 50 |
| Expected Result | Should be redirected to <http://localhost/admin/museum.php> page. |
| Test Result | Redirected to <http://localhost/admin/museum.php> page. |

*Table 4.7: For museum delete.*

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Test Result |
| On click of Delete | Delete the museum | Successful |

*Table 4.8: For adding Notices.*

|  |  |
| --- | --- |
| Test Case 3 | Add Notice |
| Test Data | Title: Holiday  Description: holiday due to gaijatra.  Date: 2079/05/03 |
| Expected Result | A message should be displayed saying “**Success !** new notice has been added” |
| Test Result | Success ! new notice has been added. |

*Table 4.9: For Editing Notice details.*

|  |  |
| --- | --- |
| Test Case 2 | Edit Notice details. |
| Test Data | Title: Holiday  Description: holiday due to Ghodejatra.  Date: 2079/05/06 |
| Expected Result | Should be redirected to <http://localhost/admin/notice.php> page. |
| Test Result | Redirected to <http://localhost/admin/notice.php> page. |

*Table 4.10: For Notice delete.*

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Test Result |
| On click of Delete | Delete the Notice | Successful |

*Table 4.11: For Activating and Deactivating user account.*

|  |  |  |
| --- | --- | --- |
| Test Case | Expected Result | Test Result |
| On click of Active | User account deactive. | Successful |
| On click of deactive | User account active | Successful |

# **Chapter 5: Conclusion and Future Recommendations**

## **5.1 Lesson learnt/ Outcome**

With the completion of this project, it was possible to achieve project goal. After registering to the system visitors can see various museums and can book ticket online without any difficulties. Users can pay ticket fare online which is the main benefit to the visitors as, they should not have to wait in line for paying ticket fare. In this way visitors can save time and enjoy with family and friends.

## **5.2 Conclusion**

In present time ticket booking takes too much time in museum because each task is performed by a human. There is no online system for purchasing museum tickets. Because of the increased use of the internet in today's world, everyone desires an online system. This traditional system of ticket booking takes too much human effort as well as time loss. So, online system is must in today’s time.

Museum ticketing site is designed by mainly focusing on online ticket booking of museum. It removes all tedious job of booking ticket, keeping data secure, managing crowd etc. It removes all paper work in the museum.

The whole system is created using waterfall model. The first phase is analysis and after analysis of the requirement, design, development, testing is done and it is deployed. The system is developed using different tools i.e. frontend tool, backend tool and some design tools. HTML, CSS, JavaScript, Bootstrap and JQuery are some of the frontend tools used in development and PHP is used as a backend tool as well as MYSQL for database connectivity. Balsamiq is also used to design wireframe/layout of system.

After all this effort museum ticketing system is developed. Museum Ticket Booking System is a web site. The concept is to keep all record in digitalized form so that there is less chance of losing data. The system is simple and user friendly so, no special knowledge is required for using the system. The prescribed objective is fulfilled in the system which make this a better website for ticket booking.

## **5.2 Future Recommendations**

There are many things that can be added in future to improve this website. There is more to be done, thus this application can be seen more useful. The system can be updated based on the user’s requirements/recommendation for this system. Here are some changes that can be made in near future:

1. Adding review and feedback option so that users can give their genuine review.
2. Adding museums that are outside of Kathmandu valley.

# **REFERENCES**

[1] https://en.wikipedia.org/wiki/Literature\_review

[2]<https://www.researchgate.net/figure/Online-Transportation-Popularity-in-> indonesia\_fig3.

[3] https://www.lucidchart.com/pages/er-diagrams

[4]https://www.researchgate.net/publication/233630620\_What\_drives\_consumers\_to\_shop\_online\_A\_literature\_Review

[5] https://www.scielo.br/j/jistm/a/cnthfmWkjtxdCQby3YRp3zG/?lang=en

[6]https://www.researchgate.net/publication/262474866\_Electronic\_Ticketing\_System\_As\_a\_Process\_of\_Innovation

[7] https://www.lucidchart.com/pages/data-flow-diagram

[8] P. Priya, What is Visual Studio Code?