Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report

on

"Staff Personal Information Management System"

[Code No: COMP 232]

(For partial fulfillment of 2nd Year/2nd Semester in Computer Engineering)

Submitted by:

Sudip Prasad Neupane(32)

Divyaswor Makai Shrestha(48)

Submitted to:

Mr. Santosh Khanal

Department of Computer Science and Engineering

Submission Date: July 24, 2018

Project Synopsis

Objective:

1. To understand the concept and utilization of database management system.
2. To develop a database related to management of personal information of the staff.
3. To create a simple web application to perform and demonstrate the basic operations on the database including insert, delete and search.

System Design:

1. Programming Language: PHP, HTML
2. Database Management System: MySQL
3. Tools used: phpMyAdmin, Sublime Text, XAMPP

Scenario:

This project is a web archive of a news publication's news, staff and ad details. This helps maintain safe record of every news published as well as the staff and dealings with other companies.

This sub-system is one of the branches of Newspaper Company which includes the information about working staff. This database sub- system includes all the official information about the working staff of a Newspaper company. All the important information about person related to the company will be included in this system. It includes Name, Family info, Address (temporary /permanent), Photo, Citizenship No, Academic qualification, Contact information along with date in which the person started working for the company.

The subsystem will provide the list of all current staff of the company. The information of the staff can be edited. If any new staff is added to the company, their respective information can be added.

List of Entities with justification

staff_list: Our database should contain all the required personal information about staff working under the newspaper organization. This entity covers almost all data of staffs like name, address, date of birth, date enroll and may more.

staff_category: A newspaper company has different types of staffs working under the company. This entity determines the position/status of the respective staffs in newspaper company.

staff_roles: This entity helps in connecting the entities staff_list and staff_catagory to determine current position of the staff in the newspaper company. It shows 'is-a' relation between the entities which will be one-to-one.

qualification: This entity gives enlists the various qualification the staff of the company may have.

staff_qualification: This entity represents 'is-a' relationship between staff_list and qualification which will be one-to-one considering we take the maximum possible qualification of the staff.

contact_number: This entity is weak entity of the staff_list. Our database needed this weak entity because one staff can have more than one contact number, i.e. many-to-one relationship.

List of attributes with justification

Attributes of entity staff_list:

- staff_id

  It is numeric value automatically increment after registration of the staffs. It is required to uniquely identify each staff in the database.

- first_name

  It stores the first name of staffs.

- last_name

  It stores the last name of staffs.

- dob

  This attributes stores the date of birth of staff.

- citizenship_no

  It stores citizenship_no of respective staffs which will be unique as citizenship numbers of two individuals are not the same.

- photo

  It stores the photo of staff.

- father_first_name

  It stores first name of staff's father.

- father_last_name

  It stores the last name of staff's father.

- date_enrolled

  It stores the date value when the respective staff started working under the company.

- email

  It stores the email of the staff which will be a unique attribute.

- temp_address

  It stores the temporary address of the staff.

- permanent_address

  It stores the permanent address of the staff.

Attributes of staff_catagory

- <u>category_id</u>

  This attribute stores numeric value which uniquely determines the each position of staff in the company.

- category

  It stores the position of the staffs in the company.

Attributes of staff_roles

- staff_id

  This attribute is foreign key in reference to the staff_id under the staff_list entity to determine the staff. It will be unique in case that a staff holds a single position in the company.

- category_id

  It is foreign key in reference to category_id under the staff_category. It is to determine the category of the respective staff.

Attributes of qualification

- <u>qualification_id</u>

  This will be the primary key for the entity to determine the various possible qualifications of the staff

- qualification

  It will give the possible qualification values

Attributes of staff_qualification

- staff_id

  This attribute is foreign key in reference to the staff_id under the staff_list entity to determine the staff. It will be unique considering we store the maximum level of qualification of the staff

- qualification_id

  It is a foreign key in reference to qualification_id under the qualification entity. It is to determine the qualification of the staff.

Attributes of contact_number (Weak Entity)

- staff_id

  This attribute is foreign key in reference to staff_id under staff_list entity, which make contact_number as weak entity of staff_list entity help in determining phone numbers of respective staffs.

- contact_number

  This is discriminant key of contact_number which stores one or more contact no of staffs by the help of staff_id.

Candidate Keys:

1. staff_list: staff_id
2. staff_category: category_id
3. qualification: qualification_id

Composite Primary Keys:

We haven’t used composite primary key in our database.

Primary Keys and why you choose the particular one as PK

staff_id:

staff_id is primary key of the entity staff_list of our database system. We chose staff_id as our primary key because it helps in uniquely determining the information about all the staff working and also for the programming purposes.

category_id:

category_id is primary key of the entity named staff_category. We selected this attribute as primary key so that it can be used to show ‘is-a’ relation between staff_list and staff_category.

qualification_id:

qualification_id is the primary key of the entity qualification. It was taken as the primary key so that it could show the ‘is-a’ relation between staff_list and staff_qualification

Partial dependencies/ transitive dependencies/Functional Dependencies/ Tables before normalization and after normalization

Functional Dependencies:

1. staff_list:

   Key: staff_id

   F = {staff_id → first_name, last_name, father_first_name, father_last_name, email, dob, citizenship_no, qualification, category, temp_address, permanent_address}

2. staff_category

   Key: category_id

   F = {category_id → category}

3. qualification

   Key: qualification_id

   F={qualification_id → qualification}

Before Normalization:

The initial table of our subsystem was

Staff_list (staff_id, first_name, last_name, father_first_name, father_last_name, email, dob, citizenship_no, qualification, category, temp_address, permanent_address, contact)

Normalization:

1NF

We perform 1NF in table staff_list by making weak entity contact_number table because if we include contact_number in staff_list table column of contact_number may have variable numbers of data because one staff can have more than one numbers data which generates the problem of data redundancy.

Tables after 1NF

i. staff_list (staff_id, first_name, last_name, father_first_name, father_last_name, email, dob, citizenship_no, qualification, category, temp_address, permanent_address)

ii. contact_number(staff_id,contact_number)

2NF

We apply 2NF rule in table staff_list and made the another table named as staff_category and staff_qualification as qualification and category were no dependent upon the given key staff_id.

Tables after 2NF

i. staff_list (staff_id, first_name, last_name, father_first_name, father_last_name, email, dob, citizenship_no, temp_address, permanent_address)

ii. contact_number(staff_id,contact_number)

iii. qualification(qualification_id, qualification)

iv. staff_qualification(staff_id,qualification_id)

v. staff_category(category_id,category)

vi. staff_roles(staff_id,category_id)

3NF

In 3NF normalization there should not be any transitive relation between the attributes. There is no requirement for 3NF because there is no transitive relation between the attributes in any of the tables of the sub-system.

Simple Indexing

Indexing was don on the staff_id column of the staff_list table.

```
CREATE INDEX staff_index  ON staff_list(staff_id)
```

Final ER Design with (Cardinalities/ weak entities/ foreign Keys/ unique keys)

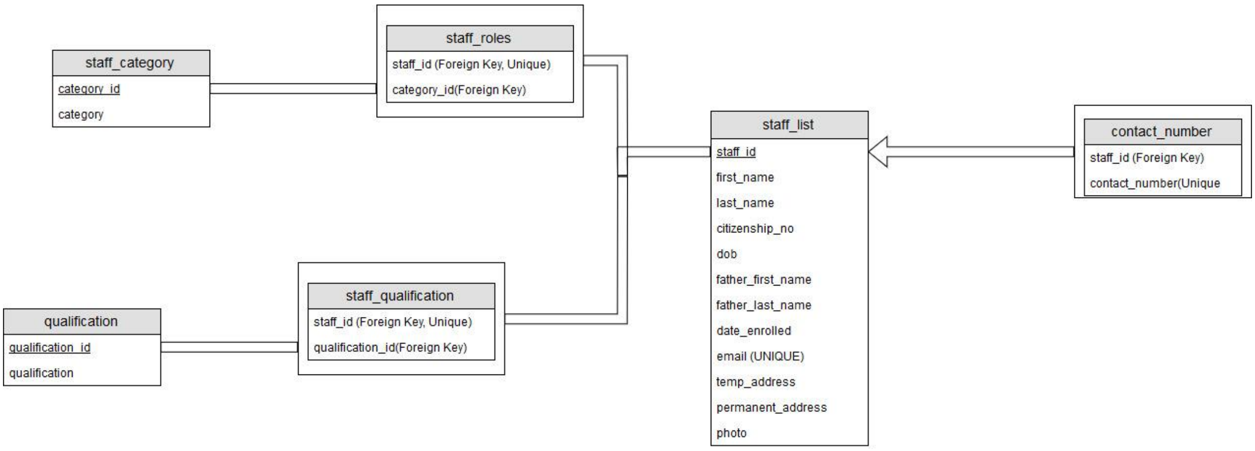
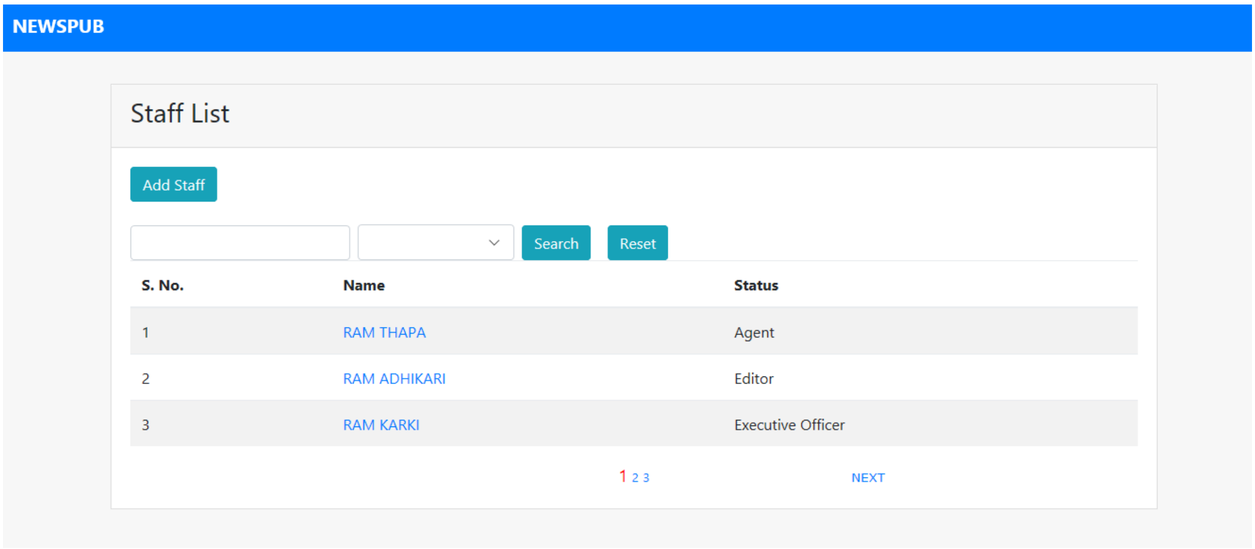


Figure 1: ER Diagram

Front END design

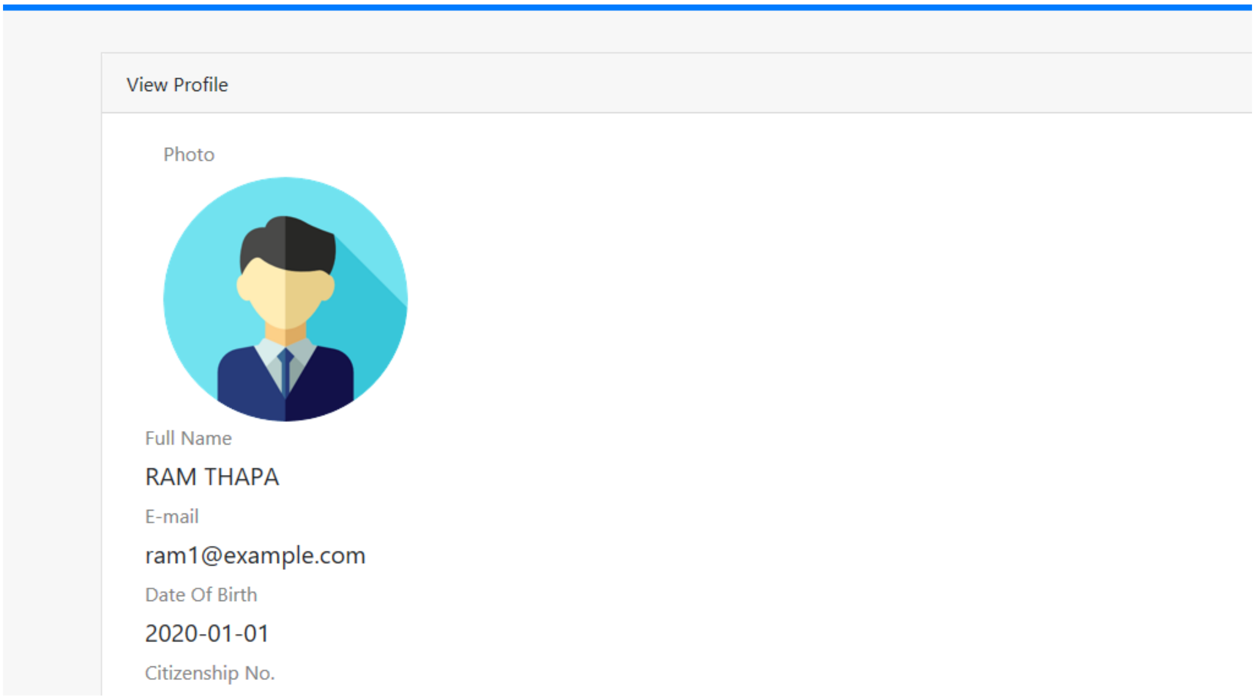
1.Staff List Page


NEWSPUB

Staff List

S. No.	Name	Status
1	RAM THAPA	Agent
2	RAM ADHIKARI	Editor
3	RAM KARKI	Executive Officer

2.Staff Profile Page



View Profile

Photo

Full Name

RAM THAPA

E-mail

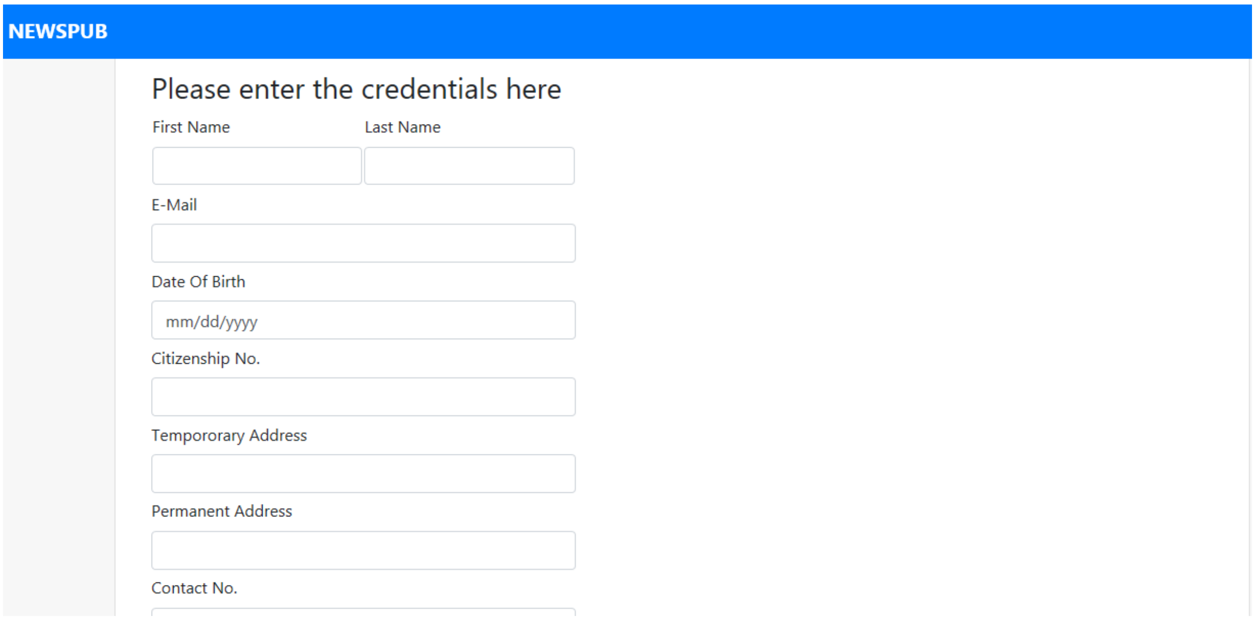
ram1@example.com

Date Of Birth

2020-01-01

Citizenship No.

3.Add a new staff page



NEWSPUB

Please enter the credentials here

First Name

Last Name

E-Mail

Date Of Birth

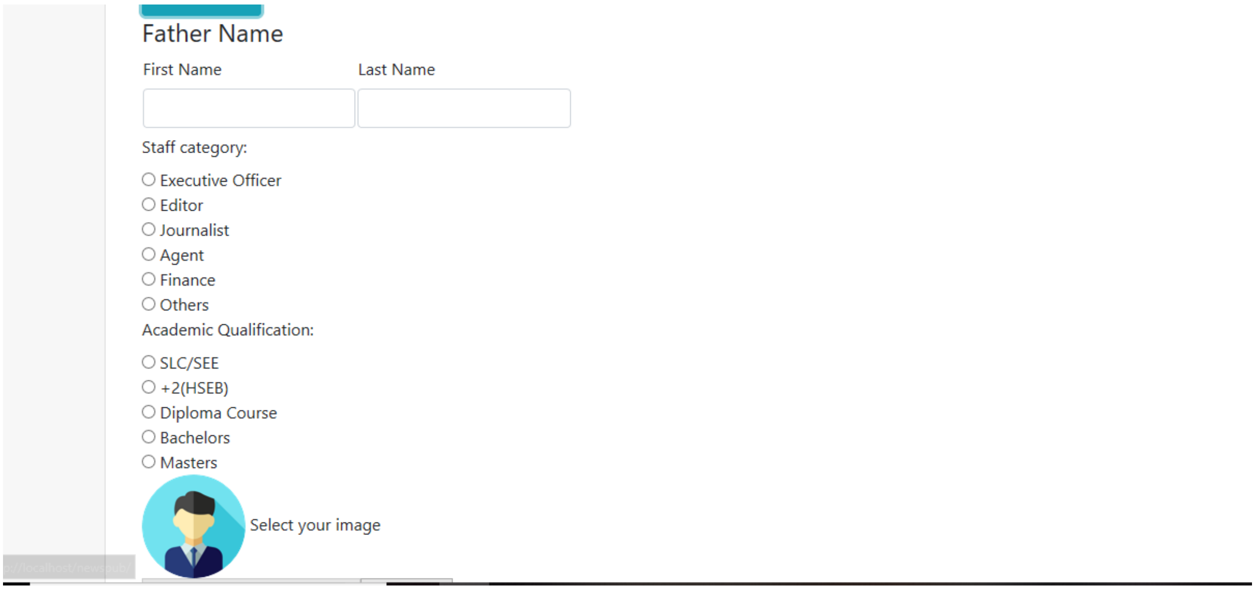
mm/dd/yyyy

Citizenship No.

Tempororary Address

Permanent Address

Contact No.



Father Name

First Name | Last Name

Staff category:

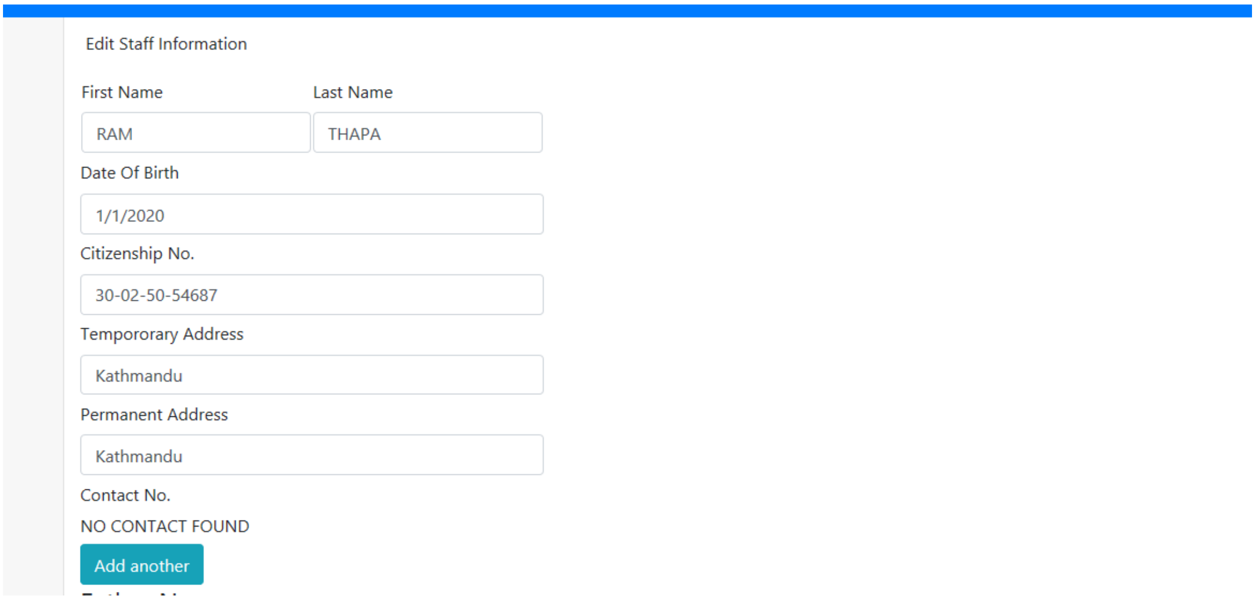
○ Executive Officer
○ Editor
○ Journalist
○ Agent
○ Finance
○ Others

Academic Qualification:

○ SLC/SEE
○ +2(HSEB)
○ Diploma Course
○ Bachelors
○ Masters

Select your image

4.Edit Staff information page



Edit Staff Information

First Name: RAM

Last Name: THAPA

Date Of Birth: 1/1/2020

Citizenship No.: 30-02-50-54687

Tempororary Address: Kathmandu

Permanent Address: Kathmandu

Contact No.

NO CONTACT FOUND

Database script

```
1.  CREATE DATABASE newspub;
2.  CREATE TABLE `staff_list` (
      `staff_id` int PRIMARY KEY AUTO_INCREMENT,
      `first_name` text,
      `last_name` text,
      `dob` date,
      `citizenship_no` text,
      `photo` text,
      `father_first_name` text,
      `father_last_name` text,
      `date_enrolled` date,
      `email` varchar(50) UNIQUE,
      `temp_address` varchar(25),
      `permanent_address` varchar(25)

    );
3.  CREATE TABLE `contact_number` (
      `staff_id` int,
      `contact_number` int UNIQUE,
      FOREIGN KEY(staff_id) REFERENCES staff_list(staff_id)
    );
4.  CREATE TABLE `qualification` (
      `qualification_id` int PRIMARY KEY,
      `qualification` varchar(25) DEFAULT NULL
    )
5.  CREATE TABLE `staff_category` (
      `category_id` int PRIMARY KEY,
      `category` text
    )
6.  CREATE TABLE `staff_qualification` (
      `staff_id` int UNIQUE,
      `qualification_id` int,
      FOREIGN KEY(staff_id) REFERENCES staff_list(staff_id),
      FOREIGN KEY(qualification_id) REFERENCES qualification (qualification_id)
    )
```

7.
   ```
   CREATE TABLE `staff_roles` (
     `staff_id` int UNIQUE,
     `category_id` int,
      FOREIGN KEY(staff_id) REFERENCES staff_list(staff_id),
      FOREIGN KEY(category_id) REFERENCES qualification (category_id)
   )
   ```

8.
   ```
   CREATE INDEX staff_index  ON staff_list(staff_id)
   ```