1. ABOUT THE PROJECT

i. **Introduction**: The Virtualinit Attendance System is made to use by government and private schools, colleges and universities. It is software developed for daily student attendance in schools, colleges and universities. This project consists of two sub-projects.

1. ID Card Generator – Web/ Desktop Application
2. Instant Attendance System by Mobile Application.

ii. **Abstract of the project:** The abstract of this project is to generate an ID card which will be designed for the mobile app especially it can scanning and attendance can be taken. In ID Card Generator, we will be storing student's information and print option to print in A4 size sheet. And teachers will have the mobile app for students and they will scan their ID card and their attendance will be stored and they don't have to enter any information like roll no, student name etc. All will be automatic and very fast.

iii. **Title of the project:** "VIRTUALINIT ATTENDANCE SYSTEM"

iv. **Aim of the project:** The main aim of this project is to computerized the attendance system and reduce time consumption and also give a transparent report to the higher authority.

Our project has the following objectives:-

- To store the attendance of the student easily by using their ID card
- To generate live reports.
- To enhance the reliability, accuracy and efficiency of the system.

v. Drawbacks of the existing system: These are the following drawbacks with the existing system:

Difficulty in report generating: They require more calculations and it takes time to generate the report and it may happen that the students may not get a single chance to improve their attendance.

Manual control: All calculations to generate report are done manually so there is greater chance of errors.

Lots of paperwork: Existing system requires lot of paper work. Loss of even a single register/record led to difficult situation because all the papers are needed to generate the reports.

Time consuming: Every work is done manually so they cannot generate report in the middle of the session or as per the requirement because it is very time consuming.

vi. Solution strategy: To replace the existing system and its problems, new system is proposed where we will use Android Application as one of the component to take attendance of the students. Using the Android Application the faculty can install this application in their Smartphones and they can login once and then they can take attendance according to their classes by scanning the QR code from the students ID. The student's attendance will be store in the database and reports can be generating easily when required.

These are the characteristic of the proposed system:

• **User Friendly:** The proposed system is user friendly because the retrieval and storing of data is fast and data is maintained efficiently.

• **Reports are easily generated:** Reports can be easily generated in the proposed system so user can generate the report as per the requirement (monthly) or in the middle of the session. User can give the notice to the students so he/she become regular.

• **Very less paper work:** The proposed system requires very less paper work. All the data is feted into the computer immediately and reports can be generated through computers. Moreover work becomes very easy because there is no need to keep data on papers.

2. FEASIBILITY STUDY

Depending on the initial investigation, the survey is extended /expanded to a more detailed feasibility study. It is a test of a system proposal according to its workability, impact on the organization, ability to meet users' needs and effective use of resource. The objective of a feasibility study is not to solve the problem but to acquire a sense of its scope. During the study the problem definitions is crystallized and aspects of the problem to be included in the system are determined.

NEED FOR FEASIBILITY STUDY: The feasibility study is carried out to test whether the proposed system is worth being implemented. Feasibility study is a test of system proposed regarding its workability, its impact on the organization ability to meet user needs and effective use of resources. The main objective of this feasibility study is not to solve the problem but to acquire a sense of its scope and its efficiency.

The key consideration involve in the feasibility study are:

1. Technical feasibility
2. Behavioral feasibility
3. Economic feasibility

1. **Technical feasibility:**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will leads to high demands on the available technical resources.

This software is created in a very friendly environment which will enable the user to use it efficiently. No need of any programming knowledge but a user will need only internet connection. The user can check the website, they can register and login if they want to use. And they can download a mobile application to take attendance for the student.

From the above we can infer that our project is technically feasible.

2. **Behavioral feasibility:**

   Behavioral feasibility is when a system has possible features of good and friendly attachments. The system being used is developed well and can be access and used by a non-technical person. This application is behaviorally feasible since the organization can download easily and it is best suited for them to use and operate easily this attendance. It is feasible for both the teacher and the student because the teacher can take attendance easily by scanning their ID card from the mobile application and the student also can keep track of their own attendance if they have less attendance.

3. **Economic feasibility :**

   To assess economic feasibility, management has to analyze costs and benefits associated with the proposed project. Economically it is feasible since the software's being used is free and open source.

3. PROJECT PLANNING AND SCHEDULED

i. SYSTEM DESIGN

Iterative: Process starts with a simple implementation of a subset of the software requirements and it changes iteratively and evolves until the full version of the software is produces, at each iteration design, coding is being added and it can be updated daily or upon the developer's need.

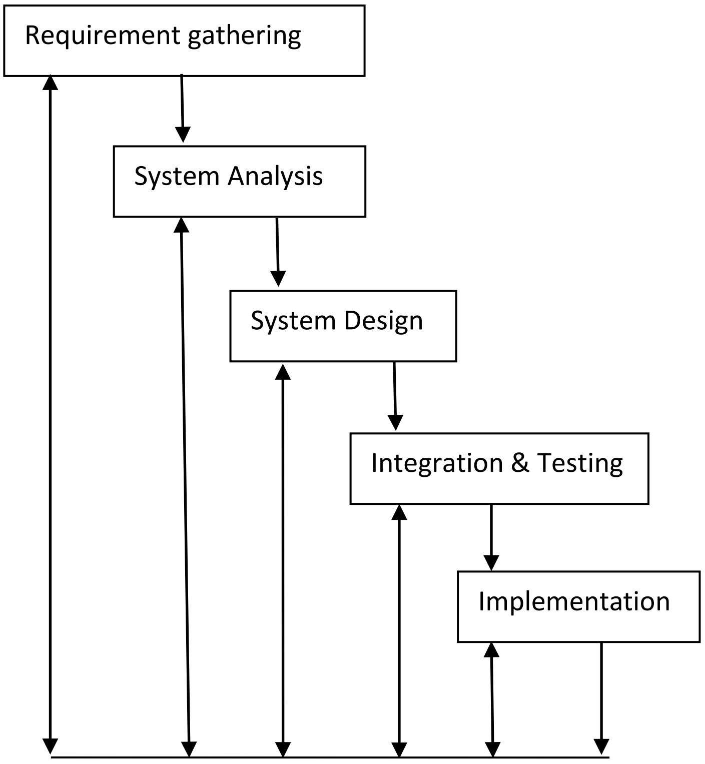


Figure 1: Iterative Model of Virtualinit Attendance System

The software development methodology I choose is an **Iterative Model**

Because of the following reasons:

i. The requirements of the software are not all gathered and analyzed at once

ii. The design process of the software is not yet finalized, frequent changes are modified everyday

iii. There are many changes that are always being added

iv. Day by day new ideas are being gathered and implemented for the betterment of the application

ii. **GANTT CHART**

Time Activity	FEBRUARY		MARCH				APRIL				MAY				JUNE		
Requirements Gathering	🟦																
Analysis		🟥	🟥	🟥	🟥	🟥											
Design							🟩	🟩									
Coding									🟪	🟪	🟪	🟪	🟪				
Testing														🟧			
Implementation															🟨	🟨	🟨
Documentation	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Literature	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3

Note: W are the number of weeks on a month.

Figure 2: Gantt chart of Virtualinit Attendance System

4. SYSTEM REQUIREMENT SPECIFICATION

i. MINIMUM SOFTWARE REQUIREMENTS:

- Windows XP, Windows 7/8/10, Linux or any OS
- Browsers (Chrome, Mozilla Firefox and others)
- Android application requires Android OS

ii. MINIMUM HARDWARE REQUIREMENTS

- Pentium Intel Core Processor
- Monitor
- Memory it must be at least 512mb
- Hard disk (2 GB minimum)
- Standard Keyboard
- Mouse
- Printer (Laser)

5. SOFTWARE DESIGN DESCRIPTION

i. **DFD (DATA FLOW DIAGRAM):** It is an important technique for modelling a system’s high level detail by showing how input data is transformed to output results through a sequence of functional transformations. It reveals the relationship among and between the various components in a program or system. It consists of four major components:

- **Entities**
- **Processes**
- **Data stores and**
- **Data flows.**

When it comes to conveying how information data flows through the system and how that data is transformed in the process, DFD’s are the method of choice over technical descriptions for three principle reasons, and they are:

- It is easier for both technical and non-technical audience to understand.
- It can provide high level system overview, complete with boundaries and connections to other systems.
- It can provide a detailed representation of system component.

i. DATA FLOW DIAGRAM OF VIRTUALINIT ATTENDANCE SYSTEM

LEVEL-0

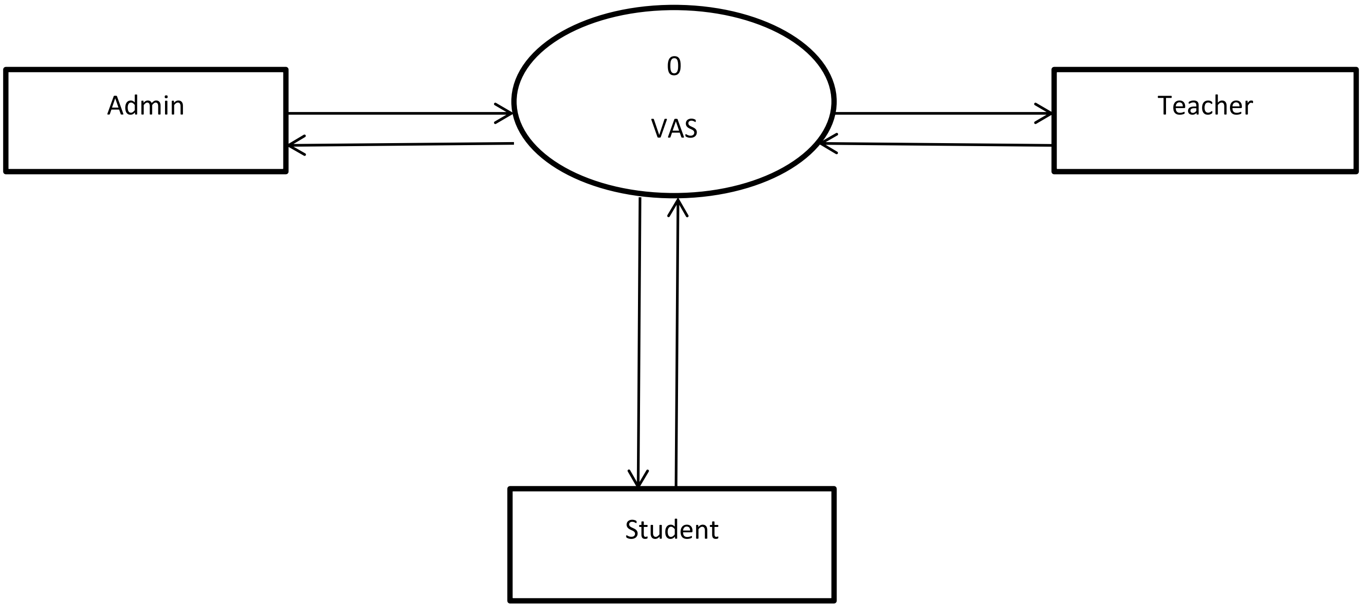


Figure 1: DFD Level-0 of Virtualinit Attendance System

LEVEL-1 of Teacher

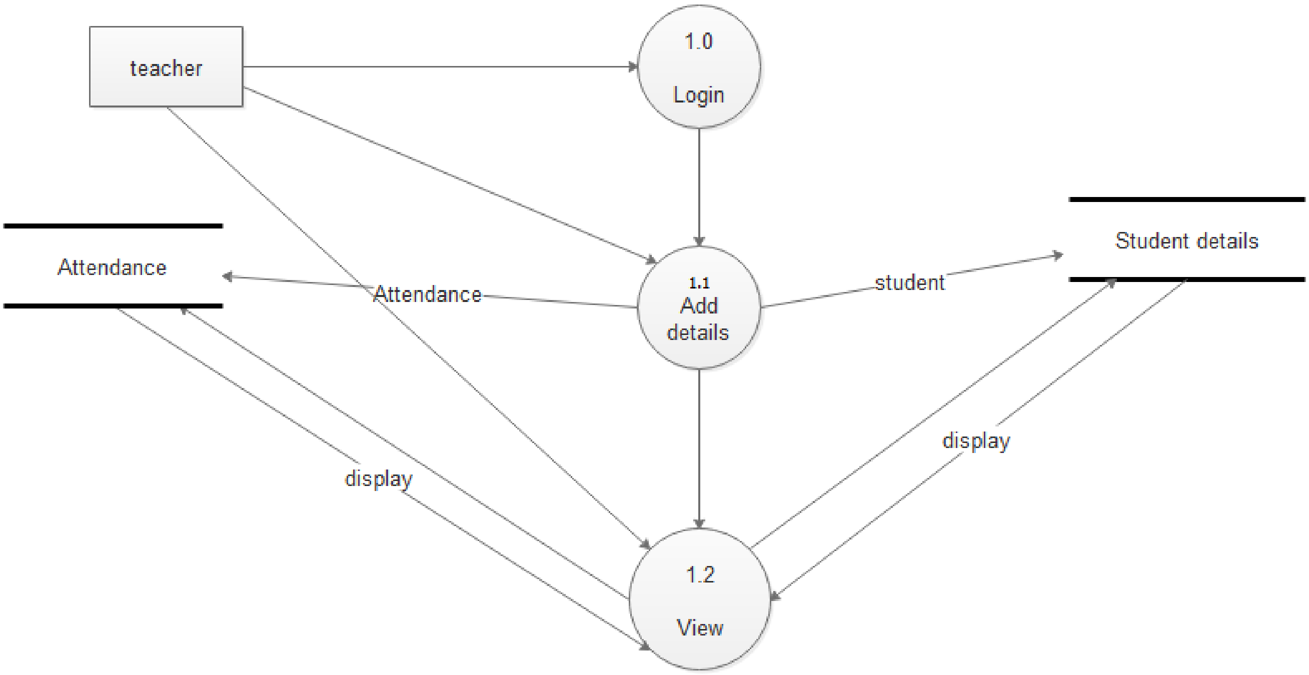


Figure 2: DFD Level-2 of Virtualinit Attendance System

LEVEL-1 of Admin

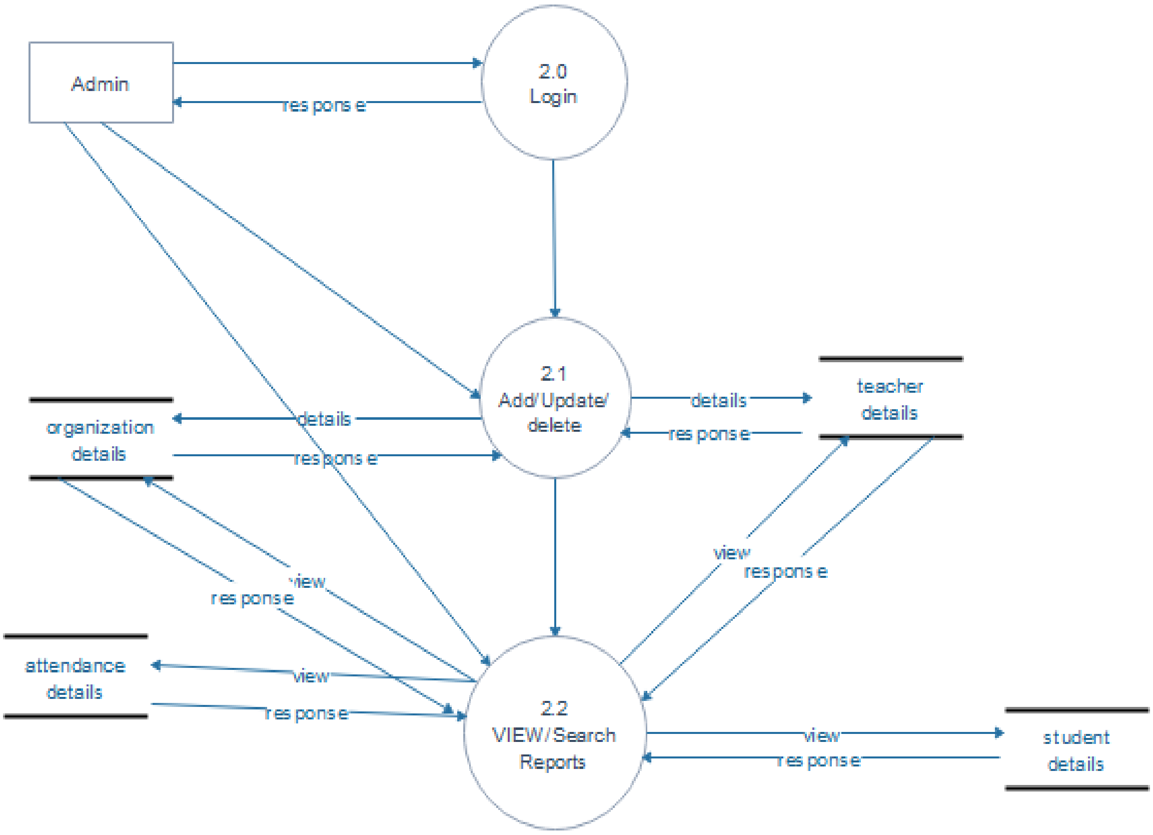


Figure 3: DFD Level-2 of Virtualinit Attendance System

ii. **ER-DIAGRAM:** Entity relation design represents the object relationship pairs in graphical forms thus we can say that the primary goal of ER-diagrams is to represents data along with their relationships.

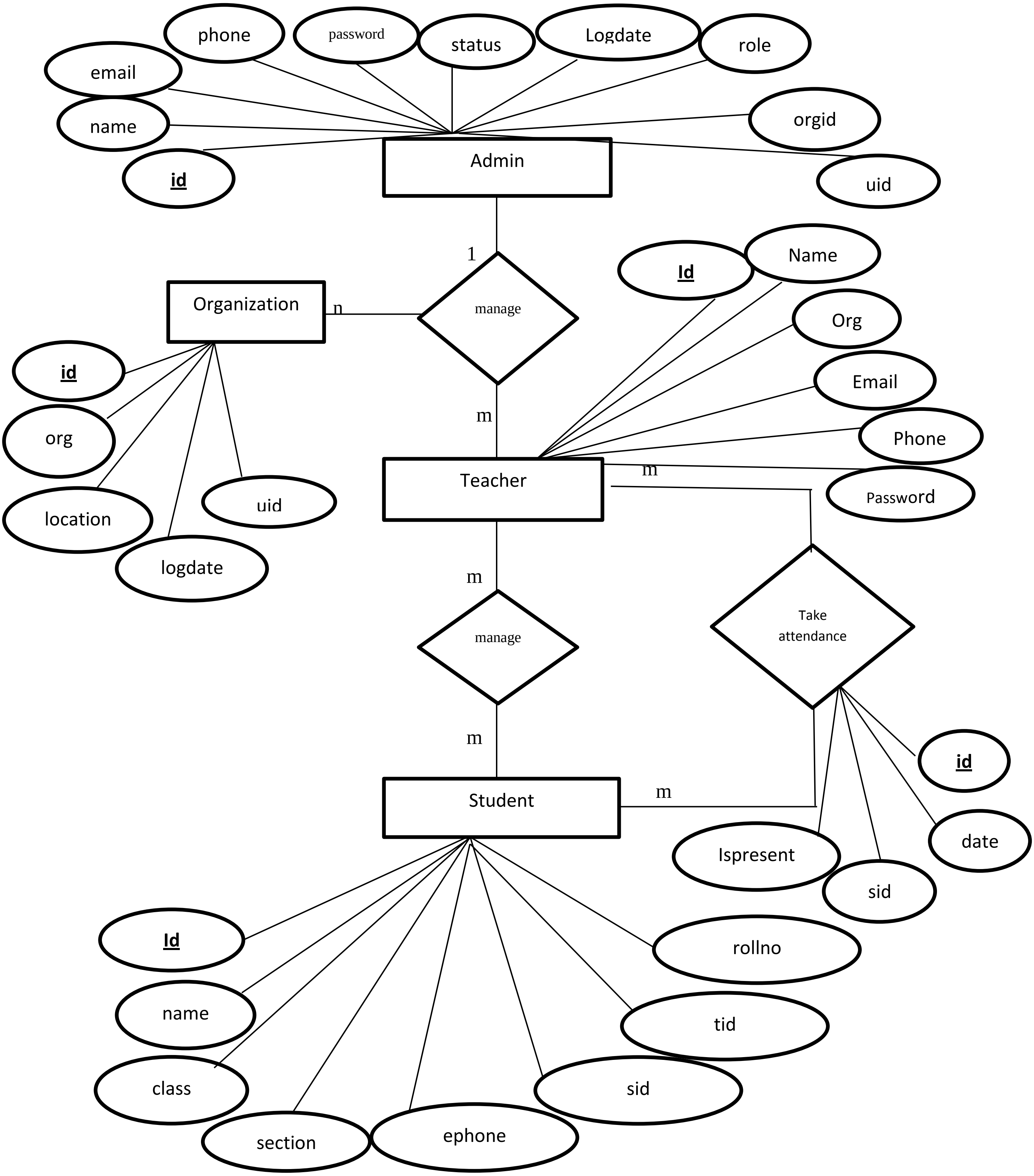
ER model uses three features to describe data :

- **Entities** which satisfy distinct real world tangible/intangible items in an application.
- **Relationships** connecting different entities and representing meaningful dependence between them
- **Attributes** which specify various properties of entities and relations involved in a system.

Its components are:

- Rectangle's representing entity sets.
- Ellipse's representing attributes.
- Diamonds representing relationship sets.
- Lines linking attribute to entity sets and entity sets to relationships sets.

ii. Database Design- ER Diagram(Entity Relationship)



iii. DATA DICTIONARY

ADMIN/USERS TABLE

Field	Type	Key	Null	Description
Id	Integer	Primary key	No	User id to be taken
Name	Varchar(50)		No	To store name of the user
Email	Varchar(50)		No	To store email of the user
Phone	Varchar(50)		No	To store phone number of the user
Password	Varchar(50)		No	Password of the user
Status	Integer(11)		No	Knowing whether the user is active or inactive
Logdate	Timestamp		No	
Role	Integer(11)		No	Role of the user. 0 means admin, 1 is teacher
Orgid	Integer(11)	Foreign key	No	
Uid	Integer(11)	Foreign key	No	

STUDENT TABLE

Field	**Type**	**Key**	**Null**	**Description**
Id	Integer(11)	Primary key	No	Student unique id to be stored
Name	Varchar(50)		No	To store the name of the student
Class	Varchar(25)		No	To store the class of the student
Section	Varchar(10)		No	To store the section of the student
Emergency-phone	Varchar(25)		No	To store the emergency no like their parents no

Logdate	Timestamp		No	
Status	Integer(11)		No	
Sid	Integer(11)	Foreign key	No	School added by the teacher
Uid	Integer(11)	Foreign key	No	
Roll no	Integer(11)		No	Roll no of the student
Security	Varchar(50)		No	Unique security code to identify student for attendance

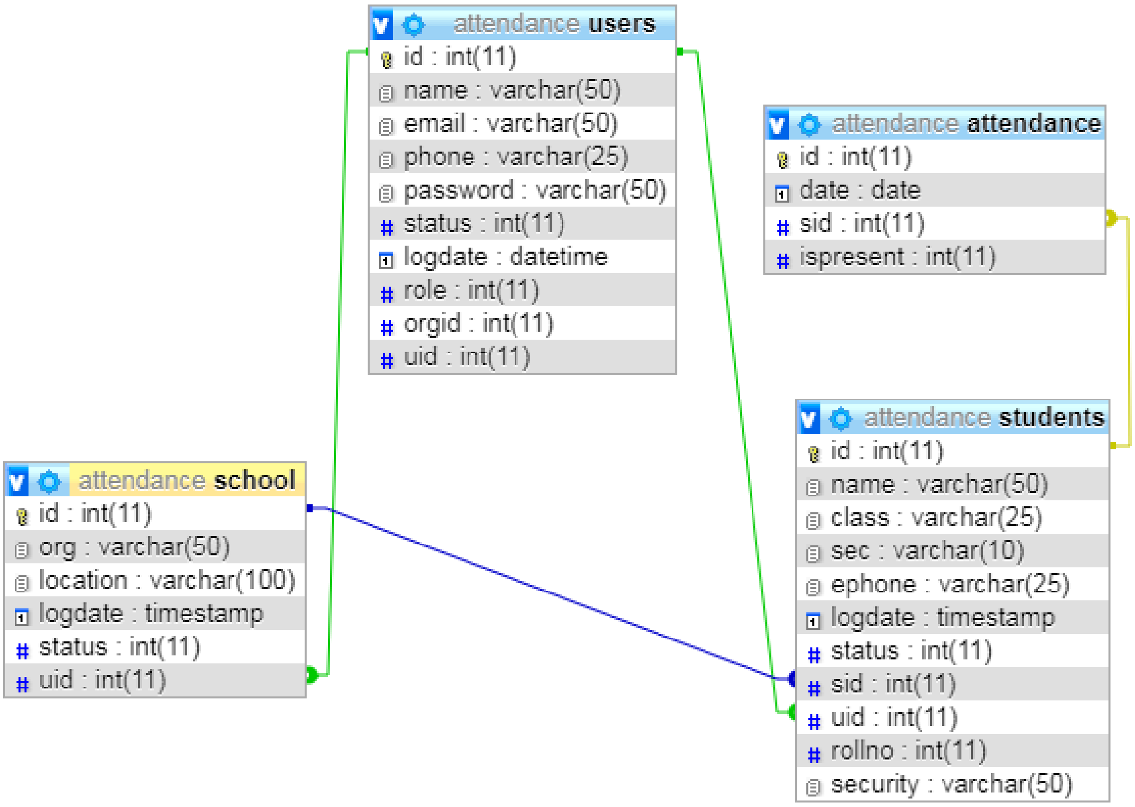
SCHOOL TABLE

Field	**Type**	**Key**	**Null**	**Description**
Id	Integer(11)	Primary key	No	School unique id to be stored
Organization	Varchar(50)		No	To store the name of the school or organization
Location	Varchar(100)		No	To store the location of the school
Logdate	Timestamp		No	
Status	Integer(11)		No	
Uid	Integer(11)	Foreign key	No	

ATTENDANCE TABLE

Field	**Type**	**Key**	**Null**	**Description**
Id	Integer(11)	Primary Key	No	Attendance unique id to be stored
Date	Date		No	To store the date which attendance is taken
Sid	Integer(11)	Foreign key	No	
Ispresent	Integer(11)		No	To store the attendance of the student if they are present

iv. TABLE STUCTURE



attendance users
id : int(11)
name : varchar(50)
email : varchar(50)
phone : varchar(25)
password : varchar(50)
status : int(11)
logdate : datetime
role : int(11)
orgid : int(11)
uid : int(11)

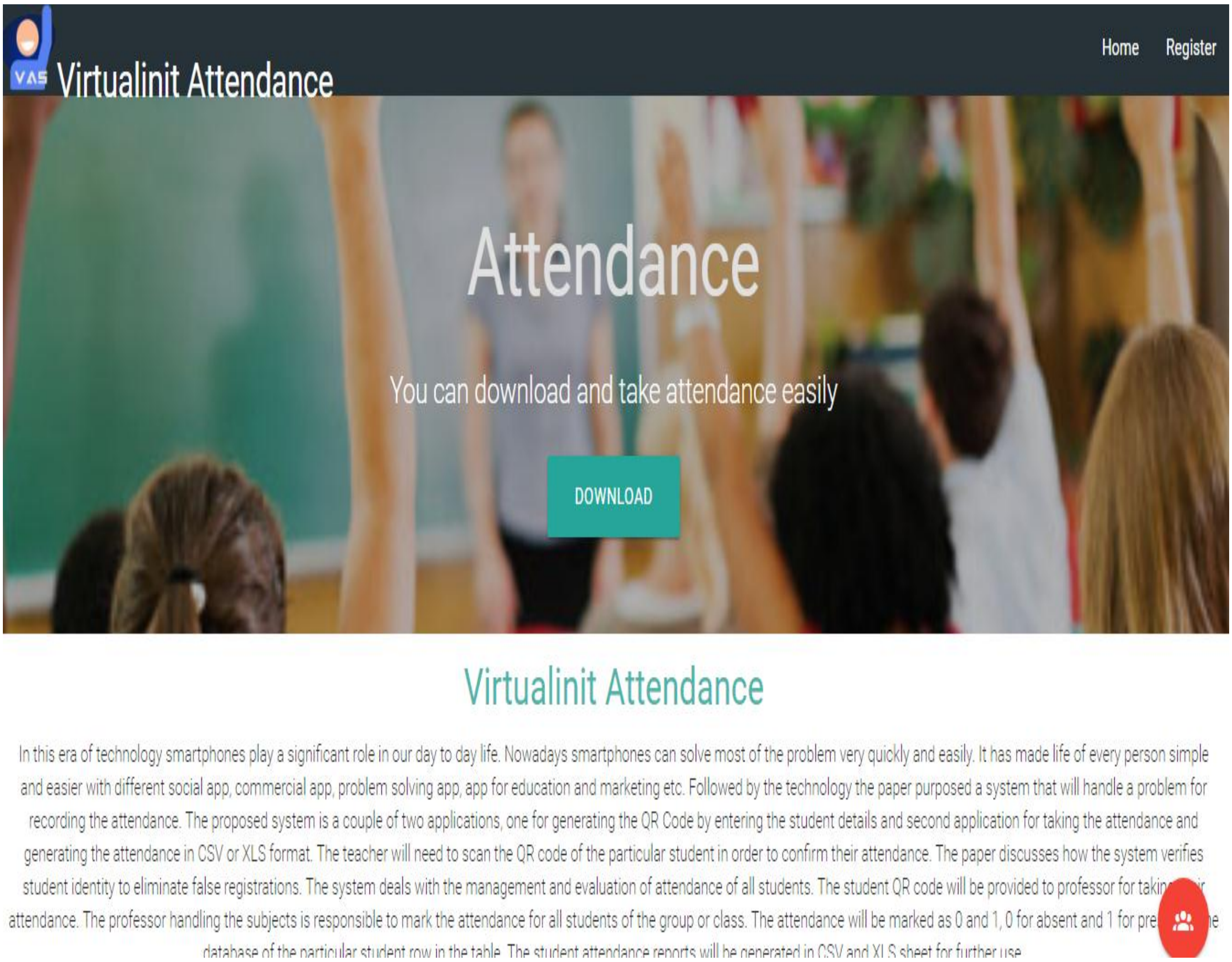
attendance attendance
id : int(11)
date : date
sid : int(11)
ispresent : int(11)

attendance school
id : int(11)
org : varchar(50)
location : varchar(100)
logdate : timestamp
status : int(11)
uid : int(11)

attendance students
id : int(11)
name : varchar(50)
class : varchar(25)
sec : varchar(10)
ephone : varchar(25)
logdate : timestamp
status : int(11)
sid : int(11)
uid : int(11)
rollno : int(11)
security : varchar(50)

v. USER INTERFACE DESIGN

a) <u>Home Page</u>



Attendance

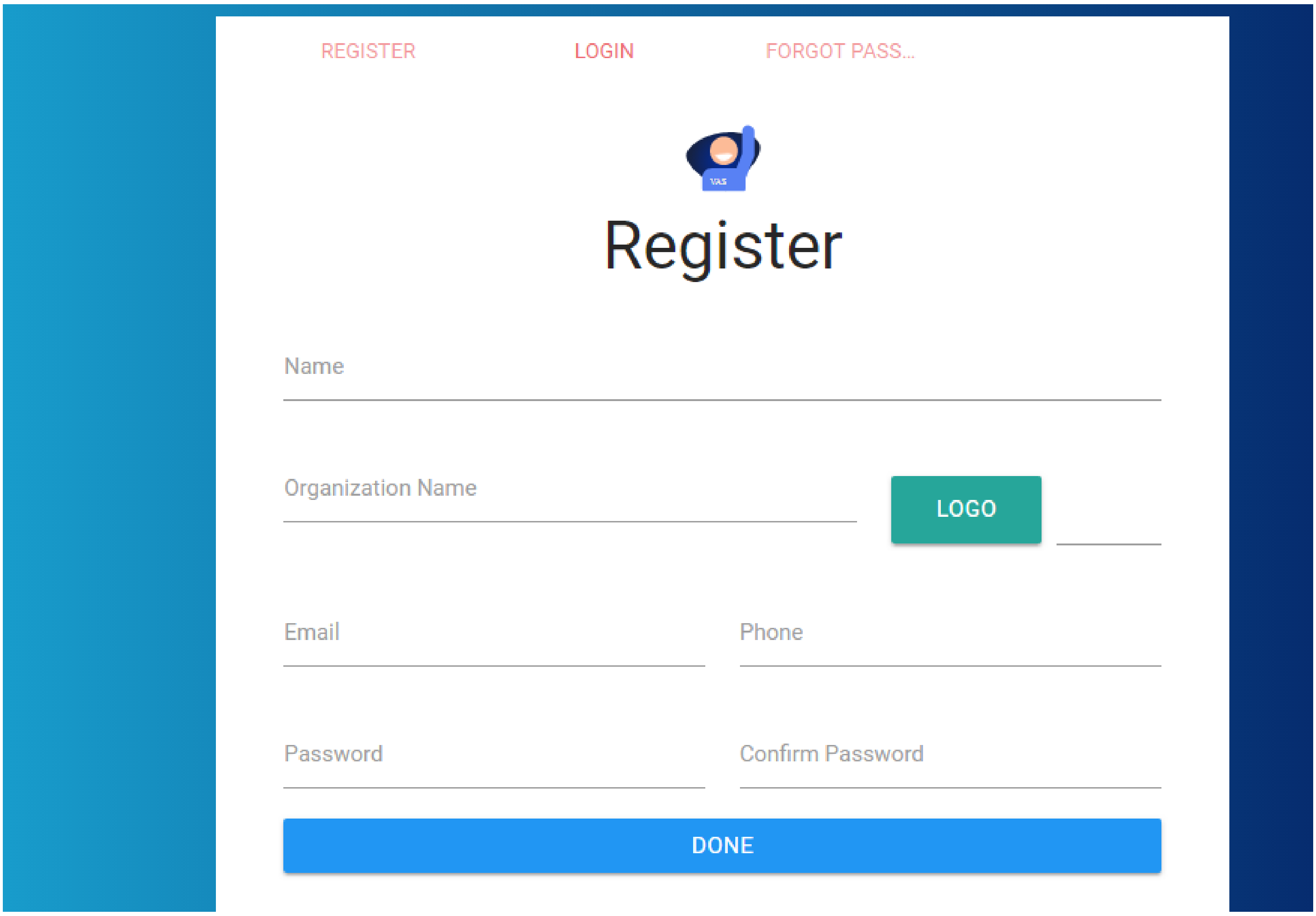
You can download and take attendance easily

Virtualinit Attendance

In this era of technology smartphones play a significant role in our day to day life. Nowadays smartphones can solve most of the problem very quickly and easily. It has made life of every person simple and easier with different social app, commercial app, problem solving app, app for education and marketing etc. Followed by the technology the paper purposed a system that will handle a problem for recording the attendance. The proposed system is a couple of two applications, one for generating the QR Code by entering the student details and second application for taking the attendance and generating the attendance in CSV or XLS format. The teacher will need to scan the QR code of the particular student in order to confirm their attendance. The paper discusses how the system verifies student identity to eliminate false registrations. The system deals with the management and evaluation of attendance of all students. The student QR code will be provided to professor for takin ir attendance. The professor handling the subjects is responsible to mark the attendance for all students of the group or class. The attendance will be marked as 0 and 1, 0 for absent and 1 for pre e database of the particular student row in the table. The student attendance reports will be generated in CSV and XLS sheet for further use

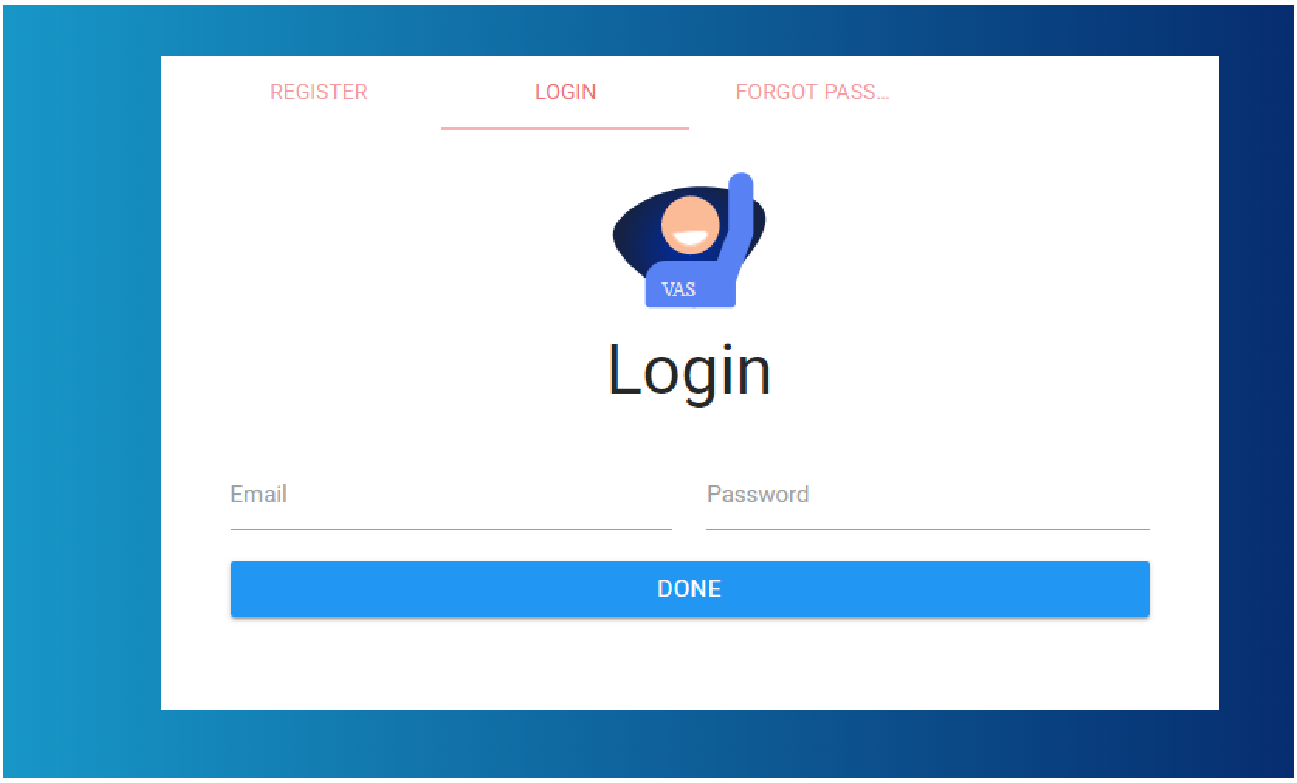
A home screen of the Virtualinit Attendance System and the login and registration form

b) Register



A registration screen for a user to register in the Virtualinit Attendance System

c) Login



REGISTER LOGIN FORGOT PASS...

Login

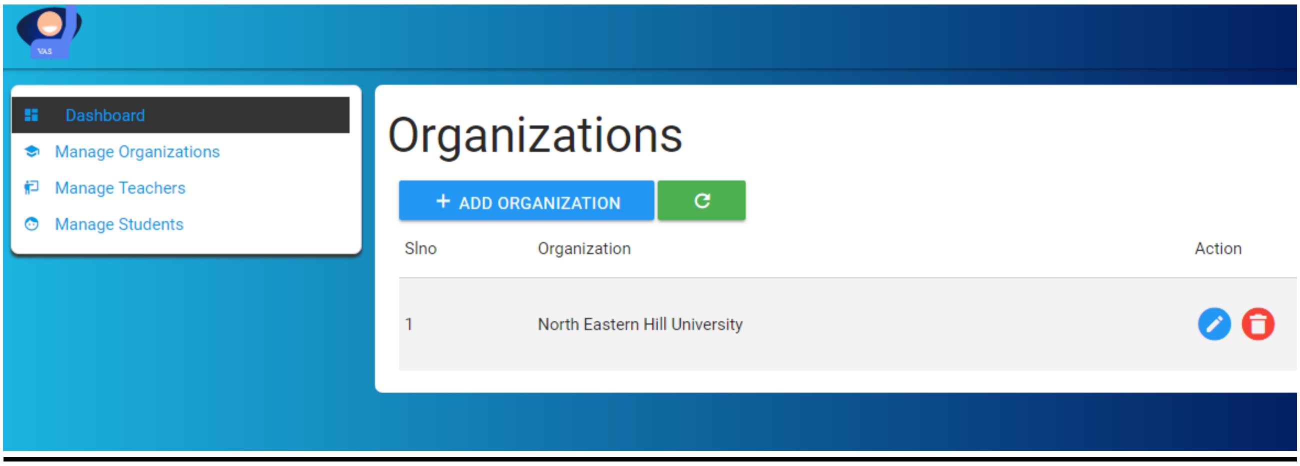
Email

Password

DONE

A login screen for the user to login

d) Dashboard

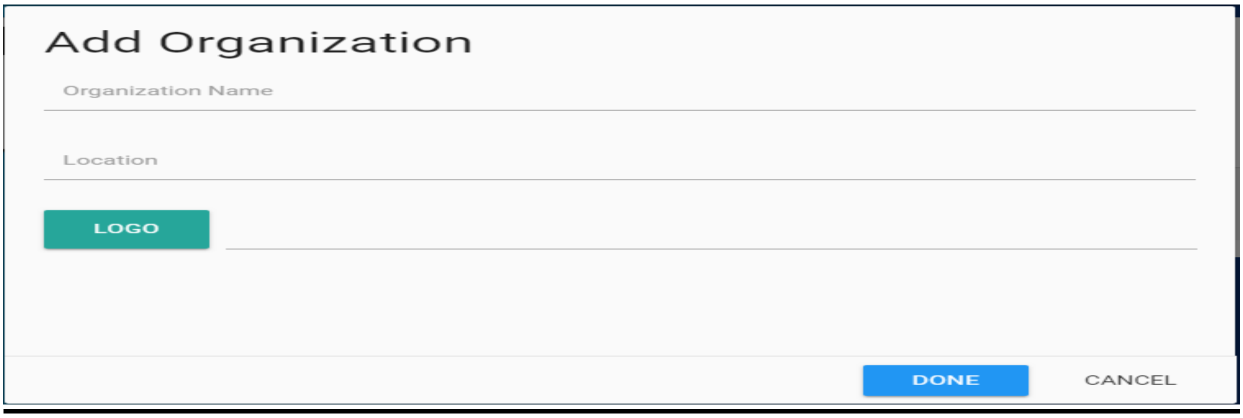


Slno	Organization	Action
1	North Eastern Hill University	

A dashboard panel for the user

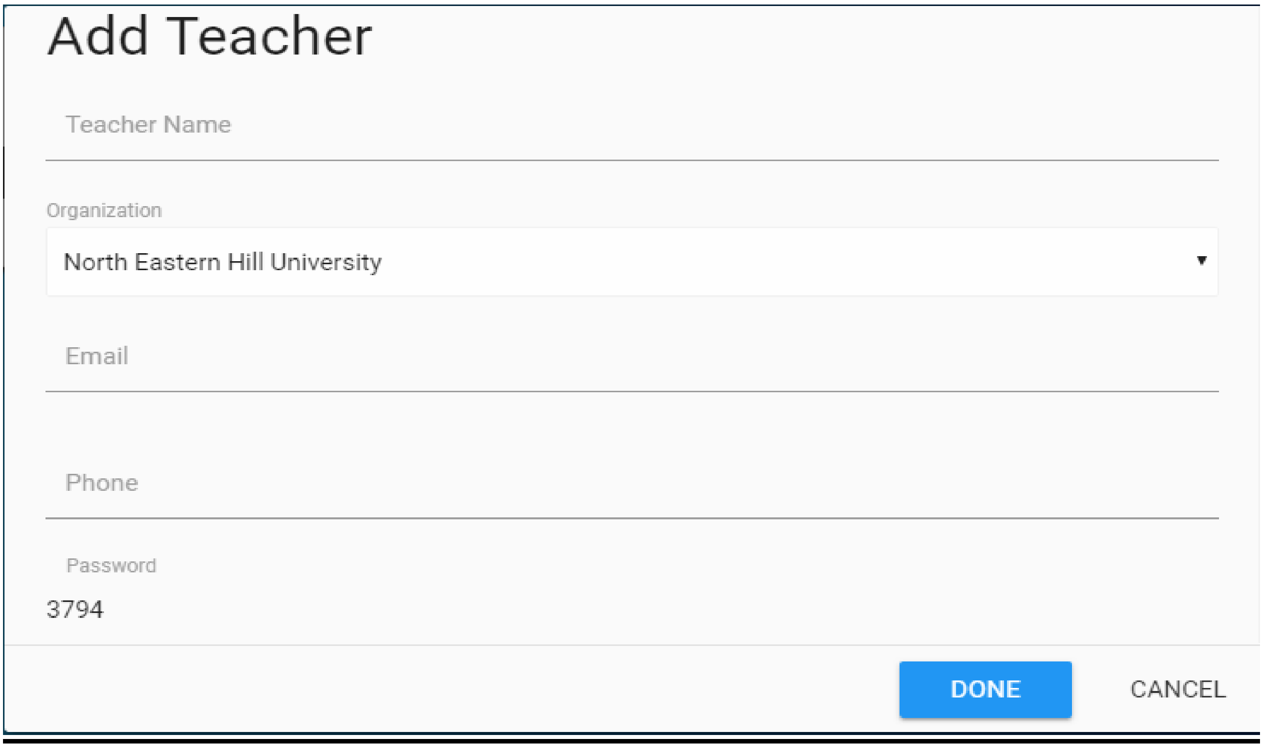
e) Form

i. Organization



A form to add organization by the user

ii. Teacher



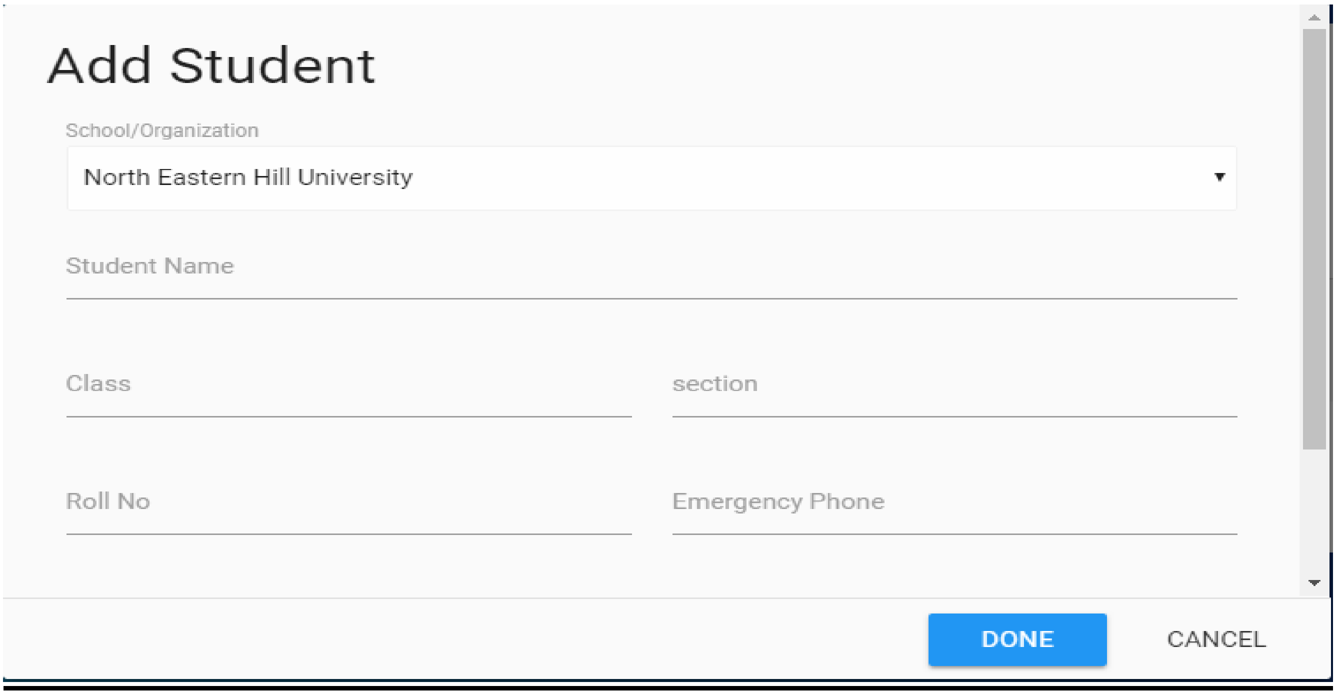
Add Teacher

Organization: North Eastern Hill University

Password: 3794

A form to add teacher by the user

iii. Student

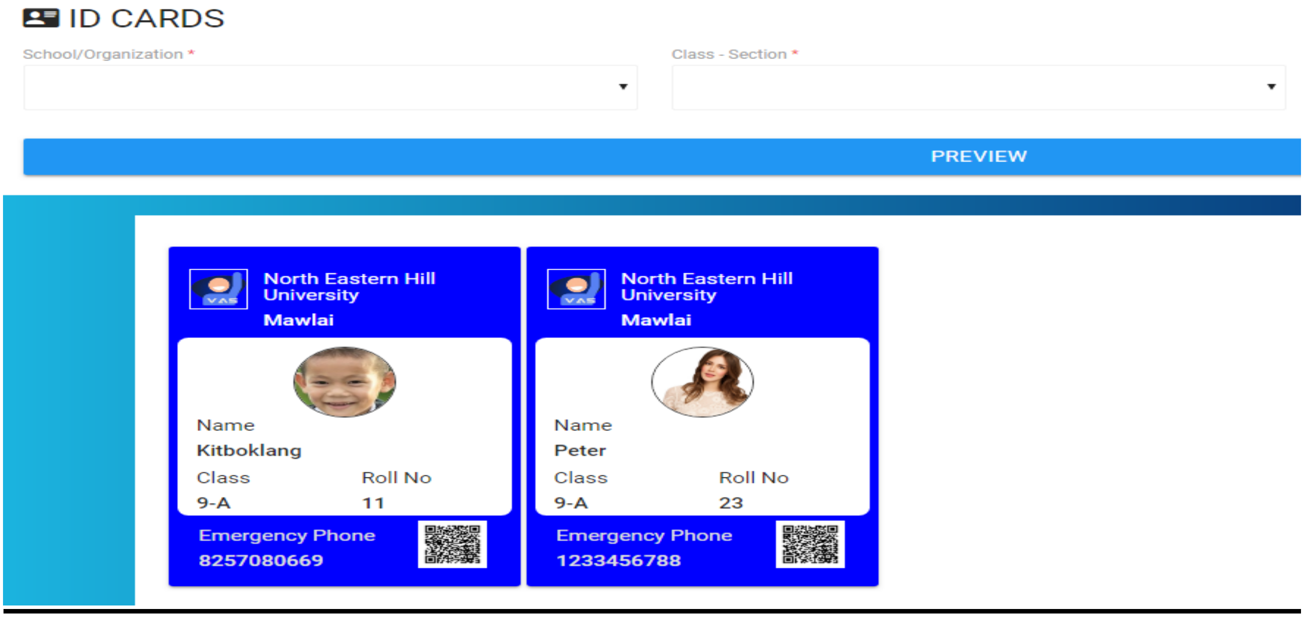


Add Student

School/Organization: North Eastern Hill University

A form to add student by the teacher

iv. ID-Card



North Eastern Hill University
Mawlai

Name
Kitboklang

Class	Roll No
9-A	11

Emergency Phone
8257080669

North Eastern Hill University
Mawlai

Name
Peter

Class	Roll No
9-A	23

Emergency Phone
1233456788

This is preview of an ID Card of the student details to print

6. SOFTWARE TEST DOCUMENTATION

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactory as regard to the expected output from the module.

REGISTRATION: This works fine

LOGIN: This works fine

ATTENDANCE: This works fine

TEST CASE 1

PROJECT NAME: VIRUALINIT ATTENDANCE SYSTEM

Step	Test steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Registration	Click register to be able to go to login	To be able to login using registration and password	User is navigated to login	Pass
2	Navigate to Login page	Username and password to login	User should be able to login if correct username and password are been given	User is navigated to login	Pass
3	User should navigate to the panel where he can add organization and teacher	Organization added	Organization name should come automatically and teacher can be added	Works fine	Pass
4	Attendance	Teacher can take attendance from the mobile application by scanning QR Code	Successful	Works fine	Pass

7. SOURCE CODE

i. Register & Login

```
<!DOCTYPE html>
  <html>
    <head>
      <?php include('inc/head.php'); ?>
    </head>

    <body class="bg2">



<div class="row formbox">
  <div class="col s12 m6 offset-m3 white hoverable">



      <div class="row">
        <div class="col s12">
          <ul class="tabs">
            <li class="tab col s3"><a href="#register">Register</a></li>
            <li class="tab col s3"><a class="active" href="#login">Login</a></li>
            <li class="tab col s3"><a href="#forgotpass">Forgot Password</a></li>
          </ul>
        </div>
        <div id="register" class="col s12">
           <form id="regform" action="p/register" method="POST" onsubmit="return false;"
enctype="multipart/form-data">

             <div class="row">
                 <div class="col s12">
                   <h3 class="center-align"><img src="assets/img/logo.svg" height="50px"
class="rounded" /><br />Register</h3>
                   <div class="row">
                     <div class="input-field col s12 ">
                       <input id="first_name" name="name" type="text" class="validate">
                       <label for="first_name">Name</label>
```

```
</div>
<div class="input-field col s12 m8">
  <input id="last_name" type="text" name="org" class="validate">
  <label for="last_name">Organization Name</label>
</div>
<div class="input-field col s12 m4">
  <div class="file-field input-field">
        <div class="btn">
          <span>Logo</span>
          <input type="file" name="logo" accept=" .jpg, .png, .jpeg">
        </div>
        <div class="file-path-wrapper">
          <input class="file-path validate truncate" type="text">
        </div>
      </div>
</div>
<div class="input-field col s12 m6">
  <input id="email" type="email" name="email" class="validate">
  <label for="email">Email</label>
</div>
<div class="input-field col s12 m6">
  <input id="phone" type="tel" name="phone" class="validate">
  <label for="phone">Phone</label>
</div>
<div class="input-field col s12 m6">
  <input id="password" type="password" name="password" class="validate">
  <label for="password">Password</label>
</div>
<div class="input-field col s12 m6">
  <input id="cpassword" type="password" name="cpass" class="validate">
  <label for="cpassword">Confirm Password</label>
</div>
<div class="col s12">
  <input type="hidden" name="csrf" value="<?php echo $_SESSION['token']; ?>"
/>
```

```
                    <span class="btn blue btn-block" onclick="dataSubmit('#regform')"
>Done</span>
                  </div>
                 </div>
                </div>
              </div>

            </form>
          </div>
          <div id="login" class="col s12">
            <form id="loginform" action="p/login" method="POST" onsubmit="return false;">

              <div class="row">
                <div class="col s12 center-align">
                  <h3 class="center-align">
                  <img src="assets/img/logo.svg" height="100px" class="rounded" /><br />
                    Login</h3>
                  <div class="row">
                    <div class="input-field col s12 m6">
                      <input id="email2" type="email" name="email" class="validate">
                      <label for="email2">Email</label>
                    </div>
                    <div class="input-field col s12 m6">
                      <input id="password2" type="password" name="password" class="validate">
                      <label for="password2">Password</label>
                    </div>
                    <div class="col s12">
                      <input type="hidden" name="csrf" value="<?php echo $_SESSION['token']; ?>"
/>
                       <span class="btn blue btn-block" onclick="eliSubmit('#loginform')"
>Done</span>
                    </div>
                  </div>
                </div>
              </div>
```

```
     </form>
    </div>
    <div id="forgotpass" class="col s12">
     <form id="forgotform" action="p/recover" method="POST" onsubmit="return false;">

      <div class="row">
        <div class="col s12">
         <h3 class="center-align"><img src="assets/img/logo.svg" height="100px" class="rounded" /><br />Forgot Password</h3>
         <div class="row">
          <div class="input-field col s12">
           <input id="email3" type="email" name="email" class="validate">
           <label for="email3">Registered Email</label>
          </div>
          <div class="col s12">
           <input type="hidden" name="csrf" value="<?php echo $_SESSION['token']; ?>" />
            <span class="btn blue btn-block" onclick="eliSubmit('#forgotform')" >Done</span>
          </div>
         </div>
        </div>
       </div>

     </form>
    </div>

   </div>



 </div>
</div>



<div class="result transparent-text"></div>
<?php include('inc/bottomjs.php'); ?>
```

```
            </body>
</html>



        <?php
        $csrf_got = @$_POST['csrf'];
        $validateEmail = false; // VALIDATE EMAIL (true/false)

        $postedpage = $page[1];
        switch ($postedpage){
            case "login":
          //   print_r($_POST);
           $validate = new VALIDATOR;

          if(isset($_POST['email']) && $validate->email($_POST['email']) && $validate->required($_POST['email'],'Email') && isset($_POST['password'])  && $validate->required($_POST['password'],'Password')  && check_csrf($csrf_got))
              {
                //$name = addslashes($_POST['name']);
               $email = addslashes($_POST['email']);
                $pass = addslashes($_POST['password']);
                $sql = "SELECT id,name,email,phone,role,orgid,uid FROM users WHERE email='$email'  AND password='$pass'";

                //echo $sql;
                 $result = new DATABASE;
                 $data = $result->query($sql);
                 $id = @$data[0]['id'];

                 //--------//
                     //    print_r($data);

                 if(COUNT($data)>0)
                 {
                   $_SESSION['user'] = $data[0];
                   // logit('login');
                echo "Successfull Login";
                echo "<script>setTimeout(function(){
                  window.location.href= 'home.php';
                },2000);</script>";
            }
            else
            {
                 echo "Invalid Login, Try Again!";
            }
          }
          else
          {
             echo "Invalid Login, Try Again!";
          }
break;

      case "register":

          $validate = new VALIDATOR;
         // print_r($_POST);
          if(isset($_POST['name']) && $validate->name($_POST['name']) &&
isset($_POST['email']) && $validate->email($_POST['email']) && isset($_POST['password'])
&& $validate->confirmpass($_POST['password'],$_POST['cpass'])  && check_csrf($csrf_got))
          {
            $formdata = $validate->need('name,email,password,phone',$_POST);
             $logdate = date("Y-m-d H:i:s");

              $result = new DATABASE;
              $emailcheck = "SELECT COUNT('email') total FROM `users` WHERE `email` LIKE
'$formdata[email]'";
              $echeck = $result->query($emailcheck);
              if($echeck[0]['total']==0)
              {
              $data = $result->insert_row('users',$formdata);
             // print_r($data);
              $userid = $data['id'];

              if($data['success'])
              {
                $sdata = $validate->need('org',$_POST);
                $sdata['uid'] = $data['id'];
                $result->insert_row('school',$sdata);

                echo  $data['message'];

              }
               else
               {
                 echo "Unable to Save Data, Kindly Try again!";
               }
              }
              else
              {
                 echo "Seems you already register, Kindly Login!";
              }
          }
          else
          {
              echo "Kindly fill your valid data correctly";
          }
           break;
default:
          $purl = $page[1];
          $pageurl = VIEW."$purl";
          if(file_exists($pageurl))
          {
              include($pageurl);
          }
          else
          {
            include(VIEW.'404.php');
```

```
                }
                break;
        }
?>
        <?php

        /**
         *  ELI FRAMEWORK
         *  EAGLE LITE - ELI FRAMEWORK
         *
         *
         */
        CLASS DATABASE{

        function connect_db($dbtype='mysql',$log='N')
        {

          $dbhost = $GLOBALS['config']['dbhost'];
          $dbuser = $GLOBALS['config']['dbuser'];
          $dbpass = $GLOBALS['config']['dbpass'];
          $dbname = $GLOBALS['config']['dbname'];

        switch ($dbtype)
        {
            case "mysql":
            try {
                $conn = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser, $dbpass);
                // set the PDO error mode to exception
                $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
                echo ($log=='Y' OR $log=='y')?"Connected successfully":"";
                return $conn;
                }
            catch(PDOException $e)
                {
                echo "Connection failed: ";
                echo $err->getMessage() . "<br/>";
                file_put_contents('PDOErrors.txt',$err, FILE_APPEND);  // write some details to an error-log outside public_html
                die();  //  terminate connection
                }
            break;

            case "sqllite":
                try {

                  $con = new PDO('sqlite:'._BASEPATH_.$dbhost.$DB_name,$DB_user,$DB_pass);
                  $con->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION );
                  //$con->exec("SET CHARACTER SET utf8");  //  return all sql requests as UTF-8
                                                                                          return $con;
                }
                catch (PDOException $err) {
                  echo "Database Connection failed.";
                  echo $err->getMessage() . "<br/>";
                  file_put_contents('PDOErrors.txt',$err, FILE_APPEND);  // write some details to an error-log outside public_html
                  die();  //  terminate connection
                }
            break;
        }

        }



        function disconnect_db()
        {
          $conn = self::connect_db();
          $conn = null;
        }


        }
?>
```

ii. Teacher Form

```
<!DOCTYPE html>
 <html>
  <head>
<?php include('inc/head.php'); ?>
  </head>

  <body class="bg2" onload="getData('v/getteacherslist',",'.teacherslist')" >

   <!-- Dropdown Structure -->
   <?php include('inc/navbar.php'); ?>



<div class="row">
 <div class="col s12 m3">
  <ul class="sidenavbar">
    <li><a href=""><i class="mdi mdi-view-dashboard"></i> Dashboard</a></li>
    <li><a href="org.php"><i class="mdi mdi-school"></i>  Manage Organizations</a></li>
    <li class="active"><a href="teachers.php"><i class="mdi mdi-teach"></i> Manage
Teachers</a></li>
    <li><a href="students.php"><i class="mdi mdi-face"></i> Manage Students</a></li>
  </ul>
 </div>

 <div class="col s12 m9 content white">
   <h3>Teachers</h3>
   <div class="row">
    <div class="col s12">
      <span class="waves-effect waves-light btn blue white-text modal-trigger "
href="#addteacher" ><i class="mdi mdi-plus"></i> Add Teacher</span>
      <span class="btn green white-text" onclick="getData('v/getteacherslist',",'.teacherslist')"
><i class="mdi mdi-refresh"></i></span>
    </div>
     <div class="col s12">
       <table class="responsive-table striped">
```

```
                <thead>
                  <td>Slno</td>
                  <td>Name</td>
                  <td>Email</td>
                  <td>Phone</td>
                  <td>Organization</td>
                  <td>Action</td>
                </thead>
                <tbody class="teacherslist"></tbody>
              </table>
          </div>
        </div>
    </div>

</div>



<div class="result transparent-text"></div>




  <!-- Modal Structure -->
  <div  id="addteacher" class="modal modal-fixed-footer">
    <div class="modal-content">
      <h4>Add Teacher</h4>
<form name="teacheradd" id="teacheraddform" action="p/addteacher" method="POST"
enctype="multipart/form-data" onsubmit="return false;" >
  <div class="input-field col s12">
    <input id="nam" type="text" name="name" class="validate">
    <label for="nam">Teacher Name</label>
  </div>
  <div class="col s12">
    <label>Organization</label>
    <select name="orgid" class="browser-default"><?php combobox('SELECT * FROM school
WHERE uid="'.$_SESSION['user']['id'].'" ORDER BY id desc  ','id','org'); ?></select>
  </div>
  <div class="input-field col s12">
    <input id="enam" type="text" name="email" class="validate">
    <label for="enam">Email</label>
  </div>
  <div class="input-field col s12">
    <input id="pnam" type="text" name="phone" class="validate">
    <label for="pnam">Phone</label>
  </div>
  <div class="input-field col s12">
    <input id="psnam" type="text" name="password" class="validate" value="<?php echo
rand(1000,9999); ?>">
    <label for="psnam">Password</label>
  </div>

  <div class="input-field col s12 hide">
    <div class="file-field input-field">
          <div class="btn">
            <span>PhoTo</span>
            <input type="file" name="logo" accept=" .jpg, .png, .jpeg">
          </div>
          <div class="file-path-wrapper">
            <input class="file-path validate truncate" type="text">
          </div>
        </div>
  </div>
  <input type="hidden" name="csrf" value="<?php echo $_SESSION['token']; ?>" />

</form>
    </div>
    <div class="modal-footer">
       <span class="btn blue " onclick="dataSubmit('#teacheraddform')" >Done</span>
      <a href="#!" class="modal-action modal-close waves-effect waves-red btn-flat ">Cancel</a>
    </div>
  </div>
  <div id="editteacher" class="modal modal-fixed-footer">
    <div class="modal-content">
      <h4>Edit Teacher</h4>
<form id="teachereditform" action="p/updaterow/users" method="POST"
enctype="multipart/form-data" onsubmit="return false;" >
  <div class="input-field col s12">
    <input id="nam" type="text" name="name" class="validate">
    <label for="nam">Teacher Name</label>
  </div>
  <div class="col s12">
    <label>Organization</label>
    <select name="orgid" class="browser-default"><?php combobox('SELECT * FROM school
WHERE uid="'.$_SESSION['user']['id'].'" ORDER BY id desc  ','id','org'); ?></select>
  </div>
  <div class="input-field col s12">
    <input id="enam" type="text" name="email" class="validate">
    <label for="enam">Email</label>
  </div>
  <div class="input-field col s12">
    <input id="pnam" type="text" name="phone" class="validate">
    <label for="pnam">Phone</label>
  </div>
  <div class="input-field col s12">
    <input id="psnam" type="text" name="password" class="validate" value="<?php echo
rand(1000,9999); ?>">
    <label for="psnam">Password</label>
  </div>

  <div class="input-field col s12 hide">
    <div class="file-field input-field">
          <div class="btn">
            <span>PhoTo</span>
            <input type="file" name="logo" accept=" .jpg, .png, .jpeg">
          </div>
          <div class="file-path-wrapper">
```

```
              <input class="file-path validate truncate" type="text">
             </div>
            </div>
       </div>
       <input type="hidden" name="id" value="" />
       <input type="hidden" name="csrf" value="<?php echo $_SESSION['token']; ?>" />

      </form>
         </div>
         <div class="modal-footer">
           <span class="btn blue " onclick="dataSubmit('#teachereditform')" >UPDATE</span>
          <a href="#!" class="modal-action modal-close waves-effect waves-red btn-flat ">Cancel</a>
         </div>
       </div>

       <!--Import jQuery before materialize.js-->
      <?php include('inc/bottomjs.php'); ?>
         </body>
       </html>

      <?php
      $csrf_got = @$_POST['csrf'];
      $validateEmail = false; // VALIDATE EMAIL (true/false)

      $postedpage = $page[1];
switch ($postedpage){

              case "addteacher":

               $validate = new VALIDATOR;

               // print_r($_POST);
               if(isset($_SESSION['user']['id']) && isset($_POST['orgid']) && $validate-
      >required($_POST['orgid'],'Organization') && isset($_POST['name']) && $validate-
      >required($_POST['name'],'Name') && $validate->required($_POST['email'],'Email') &&
      $validate->required($_POST['phone'],'Phone')  && check_csrf($csrf_got))
```

```
         {

          $formdata = $validate->need('name,orgid,email,phone,password',$_POST);
        //    print_r($formdata);
          $formdata['role'] = 1;
          $formdata['uid'] = @$_SESSION['user']['id'];
           $logdate = date("Y-m-d H:i:s");



            $result = new DATABASE;
            $scheck = "SELECT COUNT('id') total FROM `users` WHERE `orgid` =
'$formdata[orgid]' AND  `email` LIKE '$formdata[email]'";
            $echeck = $result->query($scheck);
            if($echeck[0]['total']==0)
            {
            $data = $result->insert_row('users',$formdata);
           // print_r($data);

            $userid = $data['id'];

            if($data['success'])
            {
              $id = @$data['id'];
              $destination = 'uploads/users/'.$id.'/';
               if(isset($_FILES) )
               {
                 //  echo "<pre>";
                 //print_r($_FILES);
                   if(count($_FILES)>0)
                   {
                       $total = count($_FILES);
                       // More than 0 pic
                       for($i=0; $i<$total; $i++)
                       {
                           $ret = $result->upload($_FILES['file_'.$i],$destination);
```

```
                                        // print_r($ret);
                                      }


                                   }
                               }
                              echo  $data['message'];
                             }
                              else
                              {
                                 echo "Unable to Save Data, Kindly Try again!";
                              }
                             }
                             else
                             {
                                 echo "Seems data already registered, Kindly Refresh!";
                             }
                         }
                         else
                         {
                             echo "Kindly fill your valid data correctly";
                         }
                          break;
            default:
                    $purl = $page[1];
                    $pageurl = VIEW."$purl";
                    if(file_exists($pageurl))
                    {
                        include($pageurl);
                    }
                    else
                    {
                      include(VIEW.'404.php');
                    }
                    break;
            }
?>
```

8. CONCLUSION

The project entitled “Virtualinit Attendance System” ended up in a good direction. This whole work is to access the details about the student attendance information and generate final reports. This project provides an offer to the user to enter the data through their respective registration forms. It is very helpful for the teachers and admin to keep and maintain the information about the students easily. The teacher can also take attendance easily.

9. LITERATURE REVIEW /REFERENCE

BOOKS

Leon, Alexis. Et al.2010. Fundamentals of Database Management Systems. Chennai. Tata McGraw Hill.

Roger Pressman, Ph.D. Software Engineering. Sixth Edition. New Delhi. Tata McGraw Hill.

Ramez Elmasri & Shamkant B. Navathe. Sixth Edition. Fundamentals of Database Management Systems. Pearson.

WEBSITES:

Author: Alan Dennis, Barbara Haley Wixom, David Tegarden.
Article: System Development Life Cycle.
http://en.wikipedia.org/wiki/Systems_development_life-cycle.
Date of posted: 24 June 2004. Date of Modified: September 2011.

Author: Scott W. Ambler. Article: Data Flow Diagram.
http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm.

Author: Alvin Wang, Alan Chang, Alex Mark, Kevin Louie.
Article: Materialize.
http://materializecss.com/about.html