# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"Jnana Sangama", BELAGAVI – 590018**

**A MINI PROJECT REPORT**

**ON**

# "SCHOOL MANAGEMENT SYSTEM"

Submitted in partial fulfillment of requirements for the *course*
**DBMS Laboratory with Mini Project [18CSL58]** *of Fifth Semester*
*of Bachelor of Engineering in Computer Science & Engineering*
*during the academic year 2021-22.*

**Submitted By**

| | |
|---|---|
| **"Prajwal Y P"** | **"Sudarshan K Hemminge"** |
| **"4MH19CS071"** | **"4MH19CS100"** |

**Under the Guidance of**

**"Prof. Pratap M S"**
**Assistant Professor,**
**Dept. of CS&E,**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE**

**Belawadi, S.R. Patna ( T ), Mandya ( D ) – 571477.**

**2021 - 2022**

# MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

**Belawadi, S.R. Patna ( T ), Mandya ( D ) – 571477.**

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## CERTIFICATE

This is to certify that the mini project work entitled **“School Management System”** is a bonafide work carried out by **Prajwal Y P [4MH19CS071]** and **Sudarshan K Hemminge [4MH19CS100]** in partial fulfillment for the **DBMS Laboratory with Mini Project (18CSL58)** prescribed by the Visvesvaraya Technological University, Belagavi during the year 2021-2022 for the fifth semester B.E in Computer Science and Engineering. The mini project report has been approved as it satisfies the academic requirements.

| | |
|---|---|
| Signature of Guide | Signature of HOD |
| **(Prof.Pratap M S)** | **(Dr.Shivamurthy R C)** |
| Assistant Professor, Dept. of CS&E | Professor & Head, Dept. of CS&E |
| MIT Mysore | MIT Mysore |

| Name of the Examiners | Signature with date |
|---|---|
| 1..................................... | …............................... |
| 2..................................... | …............................... |

# ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. B.G. Naresh Kumar, Principal, Maharaja Institute of Technology Mysore**, for having supported us in our academic endeavors.

We are extremely thankful to **Dr.Shivamurthy R C, Professor & Head, Department of Computer Science and Engineering,** for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide **Prof.Pratap M S, Assistant Professor, Department of Computer Science and Engineering,** for the consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff members** of **Department of Computer Science and Engineering,** for the valuable information provided by them in their respective field's. We are grateful for their co-operation during the period of our mini project.

**Prajwal Y P (4MH19CS071)**
**Sudarshan K Hemmige (4MH19CS100)**

# ABSTRACT

School Management Software is an administration tool for educational institutions. School management has always been — and still is — an extensive undertaking on the part of educational institutions around the world. It requires painstakingly monitoring the academic progress (or otherwise) of all parties involved, towards constant learning. Careful examination of resulting data will ensure optimal operations for any educational organization. Efficient and just management is of paramount importance towards satisfied students and staff, altogether.

A School Management System is a large database system which can be used for managing school's daily work. It is configurable and can be configured to meet most individual school's needs. It is a user system and can be used to store data by user at same time.Altogether it give a easy way of accessing and manipulating all the school physical records in a digital manner with the features like Security, Quality of data, Quality of services and no way of accidental damages. Here the user gets more statistical overview on the school performance with regards to academic education, sports, events, holidays and many more with a good and easy operative user-interface web page across the globe which is access by the server.

# ~~~~~ TABLE OF CONTENTS ~~~~~

## Chapter 01

# INTRODUCTION

### 1.1 Aim of the Project :

The aim of this project is School Management System can be used by education institutes to maintain the records of school easily and is intended to help any organization to store, retrive, maintain, and manage its school's personal data.

Here the School under consideration is "Sri Sharadha High School" which is located at Beckesodlur, Kodagu.

### 1.2 Overview of the Project :

School Management Software is an administration tool for educational institutions. School management has always been — and still is — an extensive undertaking on the part of educational institutions around the world. It requires painstakingly monitoring the academic progress (or otherwise) of all parties involved, towards constant learning. Careful examination of resulting data will ensure optimal operations for any educational organization. Efficient and just management is of paramount importance towards satisfied students and staff, altogether.

This project is used to allow the administrator of any Educational organization(Shri Sharadha High School) to edit and find out the personal details of a student, teachers and allows the student as well as faculties(teachers) to keep "up to date" their profile. It will also facilitate keeping all the records of students, such as their id, name, age gender,phone number, class. so all the information about an student will be available in a few seconds. Overall, it will make School Management an easier job for the administrator and the student of any organization. Without a school Management System, managing and maintaining the details of the student's as well as teachers is a tedious job for any organization. School Management system will store all the details of the students as well as teachers including their background information,personal details and all the information related to their academic Career.

The main three modules of this system are:
1. Login Module
2. Sign-up Module
3. Data Module
4. Update Module

Login module will help in authentication of user account( Lets say Admin). Users who have valid email id and password that have been registered in the database can only login into the School Management System Dashboard.

This module will help the Users(lets say admin) get registered. This module will really simplify the task of on paper registration, also after successful registration the user is identified as a authenticated valid User .

This module is very much helpful to the educational organization institution to store the details of students, faculties and infrastructures(like classes with theirs respective sections). Also on the day to day basic it allows the admin user to add a new entry of student as well as teachers following details:

Students :

(i) Universal Students Registration Number(Id)
(ii) Name
(iii) Age
(iv) Gender
(v) Phone Number
(vi) Class
(vii) Section

Teachers :

(i) Universal Students Registration Number(Id)
(ii) Name
(iii) Gender
(iv) Phone Number
(v) Date_of_joining
(vi) Respective class Assigned

All these Confidential data of the students, teachers as well as classes are store in such a way that no other malicious user's(like hackers) are restrained from accessing the data. It is because all the data is stored in Password protected Mysql database.

This module is very much helpful to the educational organization institution to update/alter the details of students, faculties and infrastructures(like classes with theirs respective sections) that have been already stored in the existing database.

## 1.3 Outcome of the Project :

Most School Management Software products evolve around four main pillars. And regardless of any extensive list of features, we can find a basic feature set around these pillars, aiming to accommodate daily routines for nearly every type of educational institution.

(i) Student admissions and records — information management :

Student registration and distribution to class groups and enrollment to classes and courses is one of the core features in School Management Software. In the long run, all information stored will be used to monitor and analyze individual progress, until graduation.

(ii) Academic management :

School faculty is just as difficult to coordinate and supervise, as students are. With roles, such as teachers, secretaries, administrative assistants and all supporting staff, School Management Software helps efficiently manage all daily academic procedures.

(iii) Security management :

This management is primarily a security feature. It's built to allow access only to information a user is allowed to view or use. That's, introducing better security for personal information, at the same time.

Benefits of employing School Management Software:

(i) An easy, automated structure for any type of school
(ii) Improved and efficient accounting data
(iii) Detailed record keeping
(iv) Powerful yet Secure and simple, in the server

## 1.4 Software Requirements :

To develop this School management system from the scratch we have used these following Software :

(i) Hypertext Markup Language(HTML) :

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.

(ii) Cascading Style Sheets(CSS) :

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML

(iii) Hypertext Preprocessor(PHP) :

PHP is a general-purpose scripting language geared towards web development.

It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!

It is deep enough to run large social networks!

It is also easy enough to be a beginner's first server side language!

(iv) Visual Studio Code(VS Code) :

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.

Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

(v) cross-platform, Apache, MySQL, PHP and Perl(XAMPP) :

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

With help of VS code all these Html code, Css code and even the php code are written and tested. In our source code we have Merged the HTML as well as PHP
Together to perform certain efficient operations. .

## Chapter 02

# DESIGN

### 2.1 Schema Diagram :

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database.

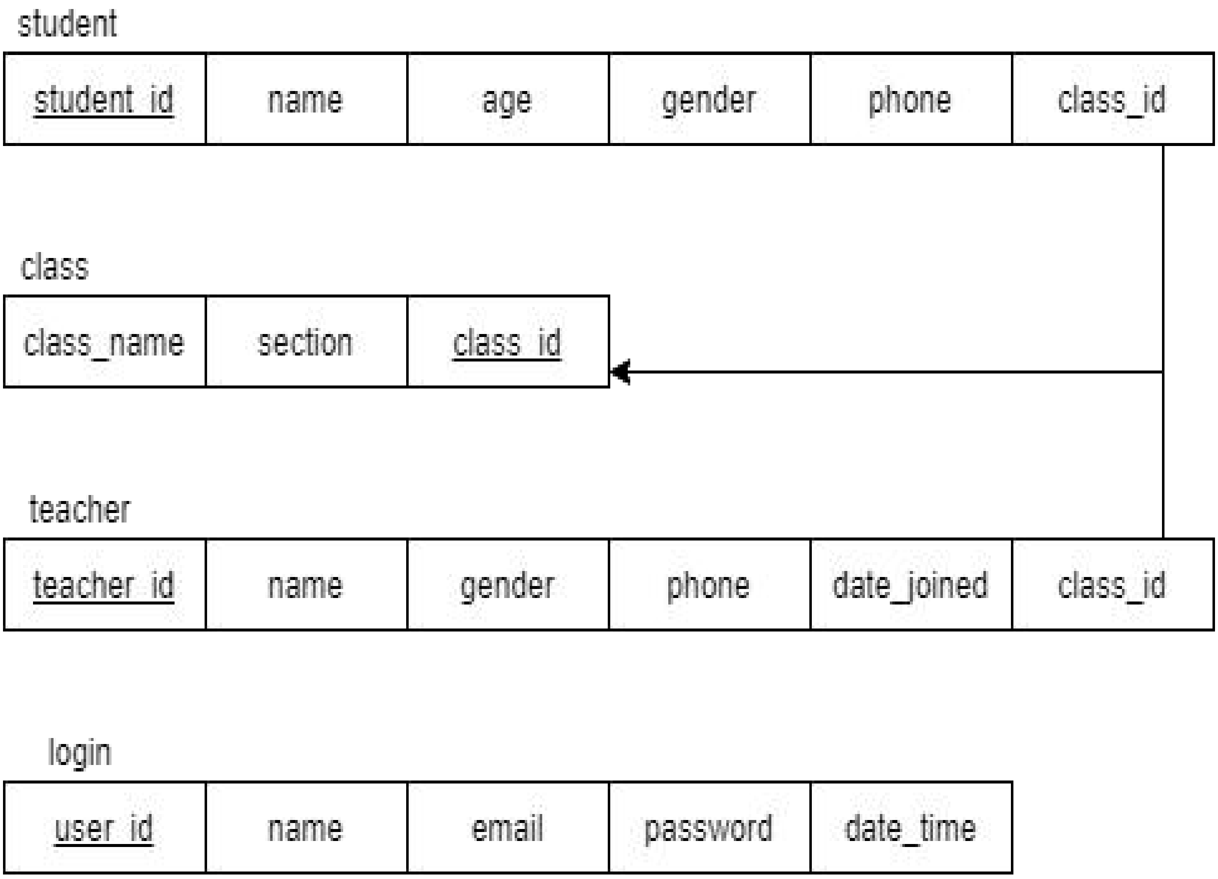


Fig 2.1 Schema Diagram

## 2.2 Entity Relationship Diagram :

ER-Diagram is a pictorial representation of data that describes how data is communicated and related to each other. Any object, such as entities, attributes of an entity, sets of relationship, and other attributes of relationship, can be characterized with the help of the ER diagram.

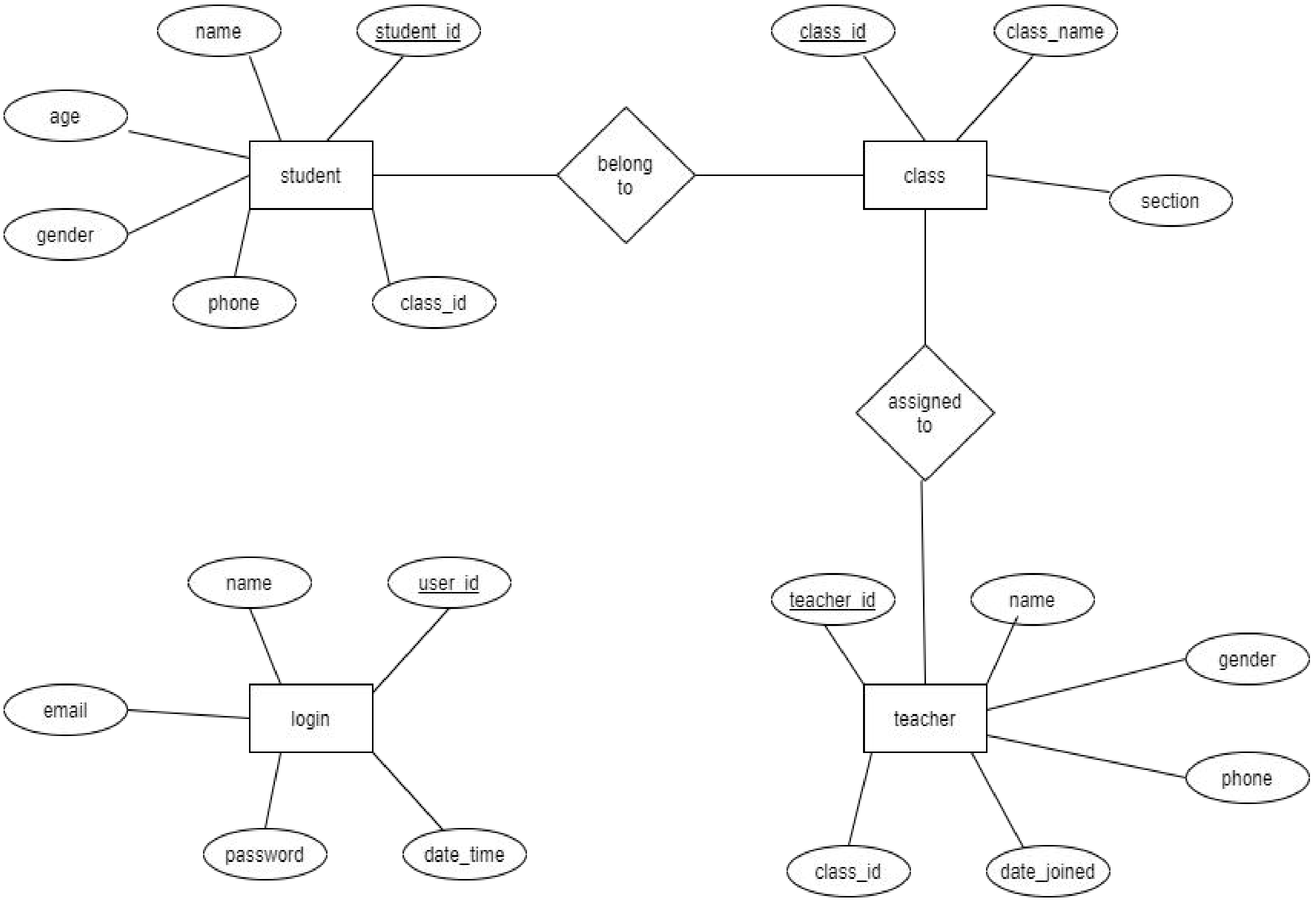


Fig 2.2 Entity Relationship

## 2.3 Use Case Diagram :

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

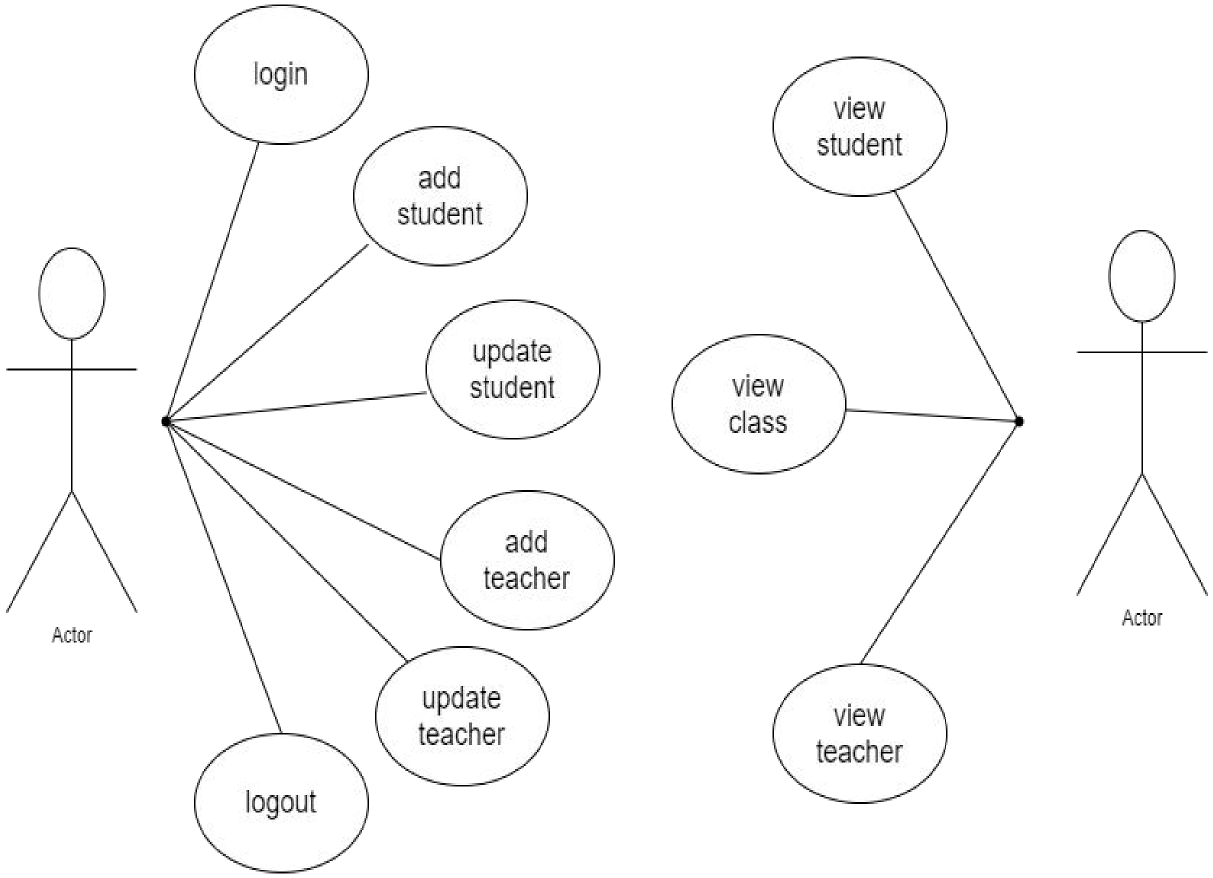


Fig 2.3 Use Case Diagram

## 2.4 Data Flow Diagram :

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

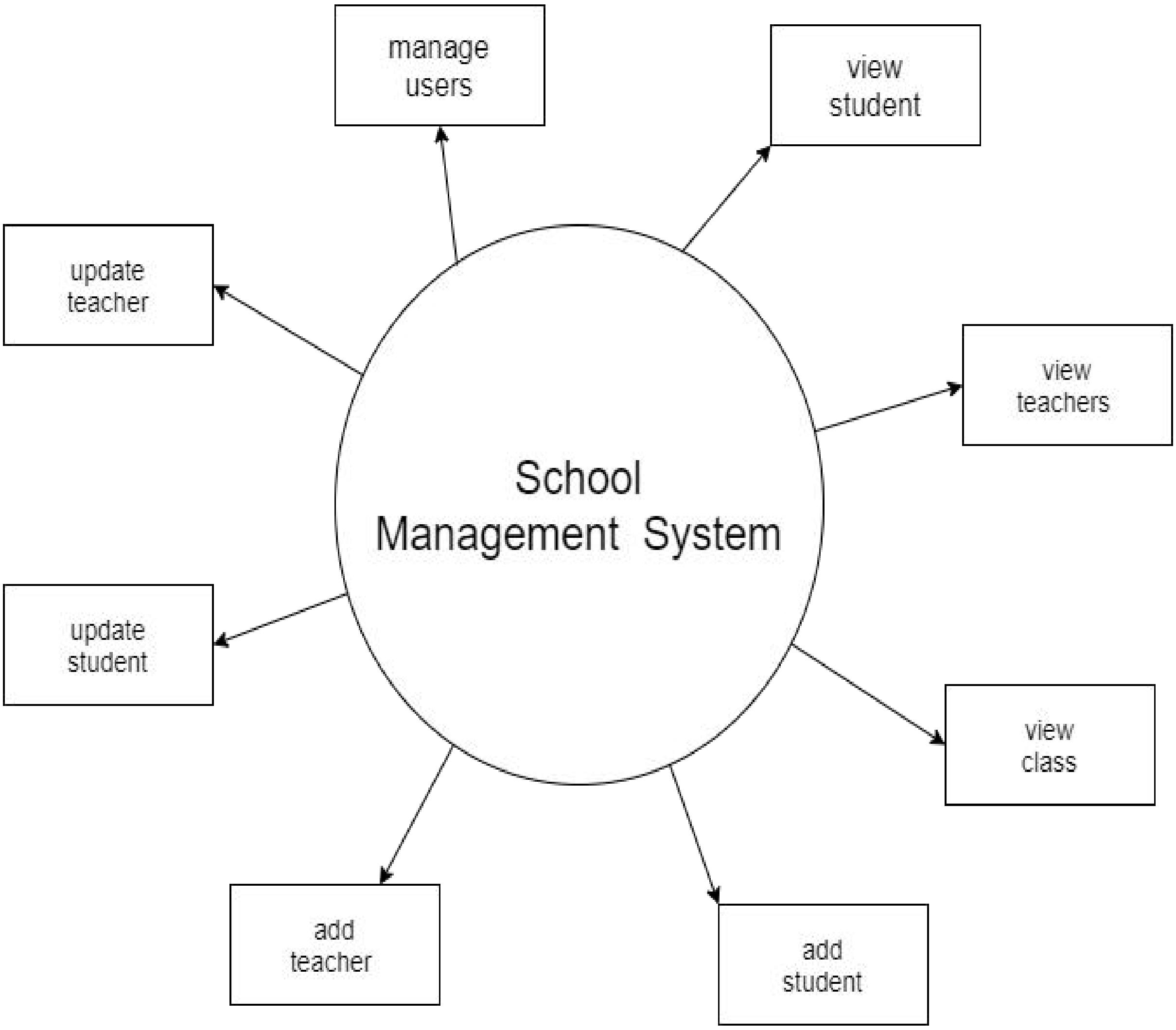


Fig 2.4 Data Flow Diagram

## 2.5 Sequence Diagram :

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

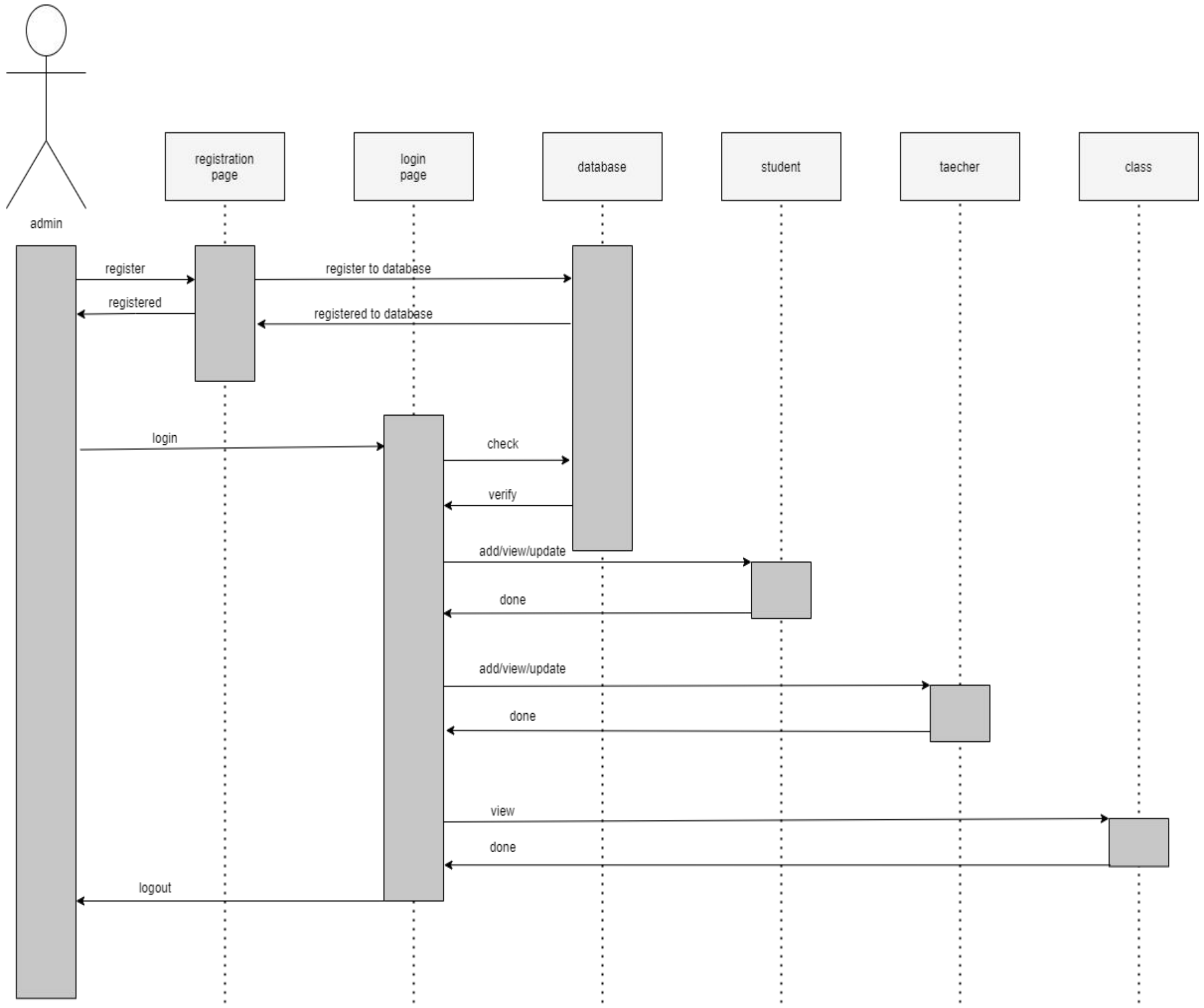
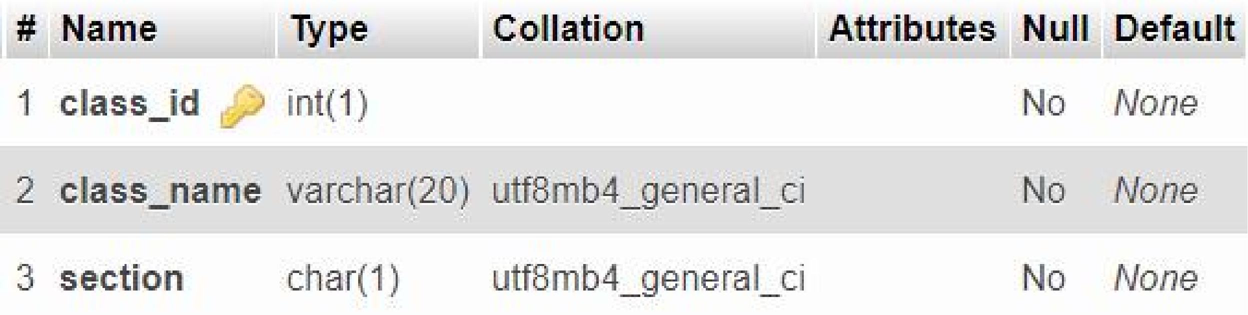


Fig 2.5 Sequence Diagram

## Chapter 03

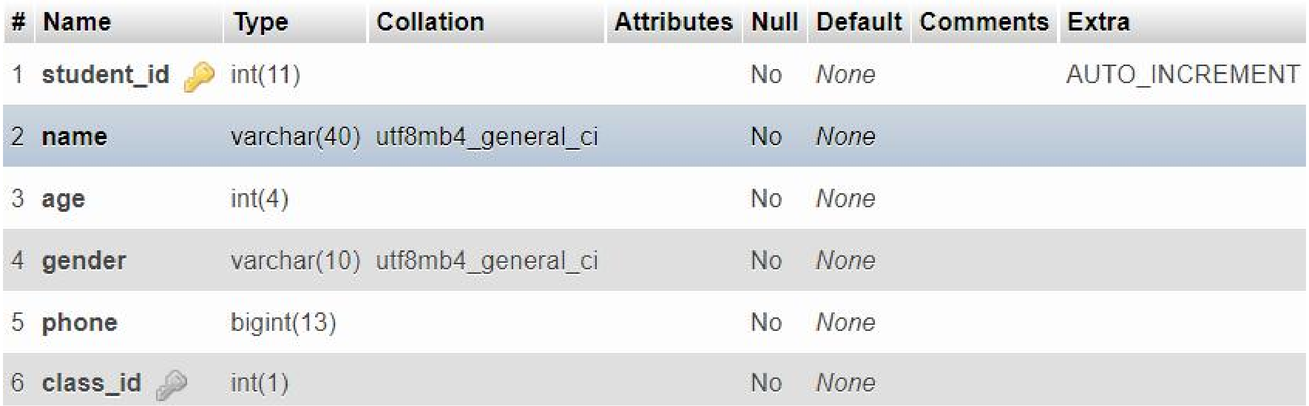
# IMPLEMENTATION

### 3.1 Description of Tables



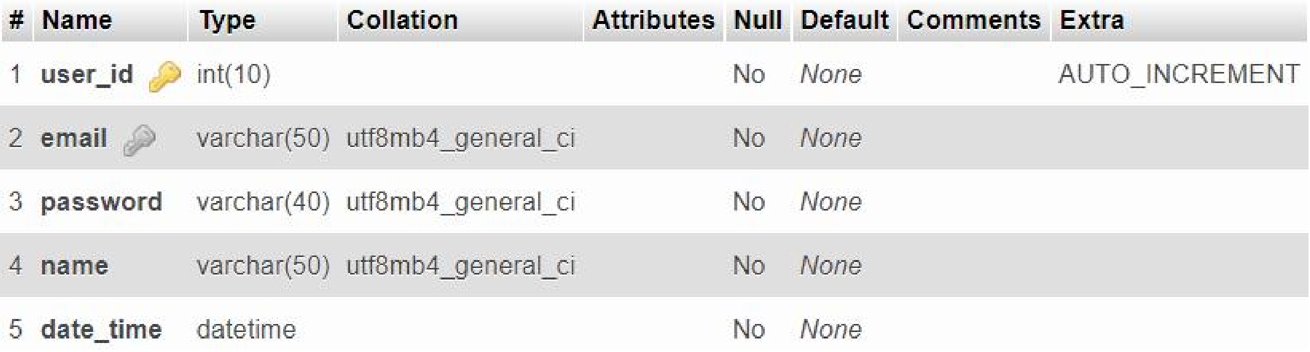
| # | Name | Type | Collation | Attributes | Null | Default |
|---|---|---|---|---|---|---|
| 1 | class_id 🔑 | int(1) | | | No | *None* |
| 2 | class_name | varchar(20) | utf8mb4_general_ci | | No | *None* |
| 3 | section | char(1) | utf8mb4_general_ci | | No | *None* |

Fig 3.1 class



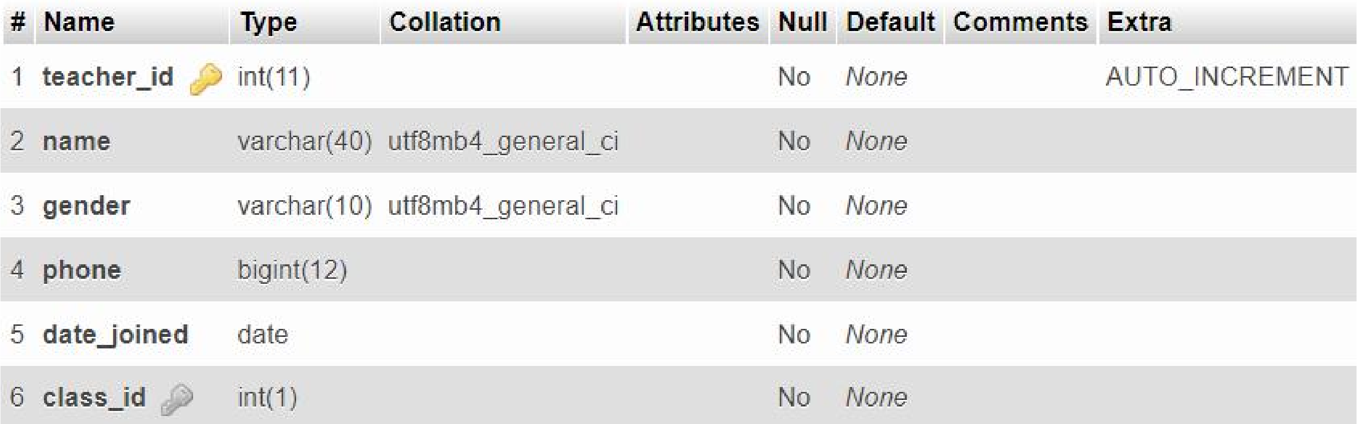
| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | student_id 🔑 | int(11) | | | No | *None* | | AUTO_INCREMENT |
| 2 | name | varchar(40) | utf8mb4_general_ci | | No | *None* | | |
| 3 | age | int(4) | | | No | *None* | | |
| 4 | gender | varchar(10) | utf8mb4_general_ci | | No | *None* | | |
| 5 | phone | bigint(13) | | | No | *None* | | |
| 6 | class_id 🔑 | int(1) | | | No | *None* | | |

Fig 3.2 student



| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | user_id 🔑 | int(10) | | | No | *None* | | AUTO_INCREMENT |
| 2 | email 🔑 | varchar(50) | utf8mb4_general_ci | | No | *None* | | |
| 3 | password | varchar(40) | utf8mb4_general_ci | | No | *None* | | |
| 4 | name | varchar(50) | utf8mb4_general_ci | | No | *None* | | |
| 5 | date_time | datetime | | | No | *None* | | |

Fig 3.3 login



| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | teacher_id 🔑 | int(11) | | | No | *None* | | AUTO_INCREMENT |
| 2 | name | varchar(40) | utf8mb4_general_ci | | No | *None* | | |
| 3 | gender | varchar(10) | utf8mb4_general_ci | | No | *None* | | |
| 4 | phone | bigint(12) | | | No | *None* | | |
| 5 | date_joined | date | | | No | *None* | | |
| 6 | class_id 🔑 | int(1) | | | No | *None* | | |

Fig 3.4 teacher

## 3.2 Constraints on Tables

(i) In table ‘login’ user_id is primary key

(ii) In table ‘class’ class_id is primary key

(iii) In table ‘student’ student_id is primary key and class_id is foreign key

(iv) In table ‘teacher’ teacher_id is primary key and class_id is foreign key

## 3.3 Back End Implementations

-- The MySQL server

```
-- default-character-set=utf8mb4
-- port=3306
-- socket="C:/xampp/mysql/mysql.sock"
-- basedir="C:/xampp/mysql"
-- tmpdir="C:/xampp/tmp"
-- datadir="C:/xampp/mysql/data"


-- Host: localhost
-- $cfg['Servers'][$i]['host'] = '127.0.0.1';
-- $cfg['Servers'][$i]['connect_type'] = 'tcp';
-- Generation Time: 25/Dec/2021:14:40:02 +0530
-- Server version: 10.4.22-MariaDB
-- PHP Version: 8.0.13


-- Table structure for table `class` --

CREATE TABLE `class` (
`class_id` int(1) NOT NULL,
`class_name` varchar(20) NOT NULL,
`section` char(1) NOT NULL,
PRIMARY KEY (`class_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4

-- Table structure for table `login` --

CREATE TABLE `login` (
`user_id` int(10) NOT NULL AUTO_INCREMENT,
`email` varchar(50) NOT NULL,
`password` varchar(40) NOT NULL,
`name` varchar(50) NOT NULL,
`date_time` datetime NOT NULL,
PRIMARY KEY (`user_id`),
UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4

-- Table structure for table `student` --

CREATE TABLE `student` (
`student_id` int(11) NOT NULL AUTO_INCREMENT,
`name` varchar(40) NOT NULL,
`age` int(4) NOT NULL,
`gender` varchar(10) NOT NULL,
`phone` bigint(13) NOT NULL,
`class_id` int(1) NOT NULL,
PRIMARY KEY (`student_id`),
KEY `class_info` (`class_id`),
CONSTRAINT `class_info` FOREIGN KEY (`class_id`) REFERENCES `class` (`class_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4

-- Table structure for table `teacher` --

CREATE TABLE `teacher` (
`teacher_id` int(11) NOT NULL AUTO_INCREMENT,
`name` varchar(40) NOT NULL,
`gender` varchar(10) NOT NULL,
`phone` bigint(12) NOT NULL,
`date_joined` date NOT NULL,
`class_id` int(1) NOT NULL,
PRIMARY KEY (`teacher_id`),
KEY `class_assigned` (`class_id`),
CONSTRAINT `class_assigned` FOREIGN KEY (`class_id`) REFERENCES `class` (`class_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4

-- Insert structure for table `login` --
```

```
INSERT INTO `login` (`email`, `password`, `name`, `date_time`)
VALUES
('$email', '$pssd', '$name', current_timestamp());

-- Insert structure for table `student` --

INSERT INTO `student` (`name`, `age`, `gender`, `phone`, `class_id`)
VALUES
('$name', '$age', '$gender', '$phone', '$class_id');

-- Insert structure for table `teacher` --

INSERT INTO `teacher`
(`name`, `gender`, `phone`, `date_joined`, `class_id`)
VALUES
('$name', '$gender', '$phone', '$date', '$class_id');

-- Update structure for table `student` --

(1)     UPDATE `student`
        SET `name` = '$name'
        WHERE `student`.`student_id` = $id;

(2)     UPDATE `student`
        SET `age` = '$age'
        WHERE `student`.`student_id` = $id;

(3)     UPDATE `student`
        SET `gender` = '$gender'
        WHERE `student`.`student_id` = $id;

(4)     UPDATE `student`
        SET `phone` = '$phone'
        WHERE `student`.`student_id` = $id;
```

```
(5)    UPDATE `student`
       SET `class_id` = '$class_id'
       WHERE `student`.`student_id` = $id;
```

-- Update structure for table `teacher` --

```
(1)    UPDATE `teacher`
       SET `name` = '$name'
       WHERE `teacher`.`teacher_id` = $id;

(2)    UPDATE `teacher`
       SET `date_joined` = '$date'
       WHERE `teacher`.`teacher_id` = $id;

(3)    UPDATE `teacher`
       SET `gender` = '$gender'
       WHERE `teacher`.`teacher_id` = $id;

(4)    UPDATE `teacher`
       SET `phone` = '$phone'
       WHERE `teacher`.`teacher_id` = $id;

(5)    UPDATE `teacher`
       SET `class_id` = '$class_id'
       WHERE `teacher`.`teacher_id` = $id;
```

## 3.4 Front End Implementations

### 3.4.1 Connection

```
$server="localhost";
$username="root";
$password="";
$database="school_db";
```

```
        $flag='F';
        $con=mysqli_connect($server,$username,$password,$database);
        if(!$con){
            die("Connection to this Database failed due to ".mysqli_connect_error());
        }
```

### 3.4.2 Add User

```
    if(isset($_POST['sub'])){
    $email = $_POST['email'];
    $name = $_POST['name'];
    $pssd = $_POST["pssd"];
    $pssd1=$_POST["pssd1"]
    $sql="INSERT INTO `login` (`email`, `password`, `name`, `date_time`)
VALUES ('$email', '$pssd', '$name', current_timestamp());";

    if (empty($email)) { array_push($errors, "Email is required !!"); }
    if (empty($name)) { array_push($errors, "Username is required !!"); }
    if (empty($pssd)) { array_push($errors, "Password is required !!"); }
    if ($pssd != $pssd1) {
        array_push($errors, "The two passwords do not match !!");
    }

    $user_check_query = "SELECT * FROM `login` WHERE `email` LIKE '$email'
OR `name` LIKE '$name'";
    $result = mysqli_query($con, $user_check_query);
    $user = mysqli_fetch_assoc($result);

    if ($user) { // if user exists
      if ($user['name'] === $name) {
        array_push($errors, "Username already exists !!");
      }

      if ($user['email'] === $email) {
        array_push($errors, "Email already exists !!");
```

```
            }
        }
        if (count($errors) == 0){
            if($con->query($sql) == true){
                $flag='T';
            }
        }
    }
```

### 3.4.3 Check User

```
        if(isset($_POST['sub'])){
        $email = $_POST['email'];
        $pssd = $_POST["pssd"];

        if (empty($email)) {
            array_push($errors, "Email is required !!");
        }
        if (empty($pssd)) {
            array_push($errors, "Password is required !!");
        }

        if (count($errors) == 0) {
            $query = "SELECT * FROM `login` WHERE `email` LIKE '$email' AND
`password` LIKE '$pssd'";
            $results = mysqli_query($con, $query);
            if (mysqli_num_rows($results) == 1){
                header('location: h1.html');
            }
            else{
                array_push($errors, "Wrong username/password !!!");
            }
        }
    }
```

### 3.4.3 Add student

```
if(isset($_POST['sub'])){
$name = $_POST['name'];
$age = $_POST["age"];
$gender = $_POST["gender"];
$phone = $_POST["phone"];
$class_id = $_POST["class"];
$section = $_POST["section"];

$sql="INSERT INTO `student` (`name`, `age`, `gender`, `phone`, `class_id`) VALUES ('$name', '$age', '$gender', '$phone', '$class_id');";

if (empty($name)) { array_push($errors, "Student name is required !!"); }
if (empty($age)) { array_push($errors, "Student age is required !!"); }
if (empty($phone)) { array_push($errors, "Student Phone Number is required !!"); }

if (count($errors) == 0){
   if($con->query($sql) == true){
      $flag='T';
   }
}
}
```

### 3.4.4 View student

```
$sql="SELECT * FROM `student`;";
$results = mysqli_query($con, $sql);

foreach ($results as $x){
   $c += 1;
}
    <?php foreach ($results as $res) : ?>
         <tr>
```

```
                <td> <?php echo $res['student_id']."  ";  ?> </td>
                <td> <?php echo $res['name']."  "; ?> </td>
                <td> <?php echo $res['age']."  ";  ?> </td>
                <td> <?php echo $res['gender']."  ";  ?> </td>
                <td> <?php echo $res['phone']."  ";  ?> </td>

                <?php
                  $id=$res['class_id'];
                  $sql2="SELECT * FROM `class` WHERE `class_id` = $id;";
                  $results2 = mysqli_query($con, $sql2);
                  $res2 = mysqli_fetch_assoc($results2);
                ?>

                <td> <?php echo $res2['class_name']."  ";  ?> </td>
                <td> <?php echo $res2['section']."  ";  ?> </td>
              </tr>
            <?php endforeach ?>
```

### 3.4.5 Update student

```
     if(isset($_POST['sub'])){
    $id = $_POST['sub'];

    if( !empty($_POST['name'])){
      $name = $_POST['name'];
      $sql="UPDATE `student` SET `name` = '$name' WHERE
`student`.`student_id` = $id;";
      if($con->query($sql) == true){
        $flag='T';
      }
    }
    if( !empty($_POST['age'])){
      $age = $_POST['age'];
      $sql="UPDATE `student` SET `age` = '$age' WHERE `student`.`student_id` =
$id;";
```

```
            if($con->query($sql) == true){
                $flag='T';
            }
        }
        if( !empty($_POST['gender'])){
            $gender = $_POST['gender'];
            $sql="UPDATE `student` SET `gender` = '$gender' WHERE `student`.`student_id` = $id;";
            if($con->query($sql) == true){
                $flag='T';
            }
        }
        if( !empty($_POST['phone'])){
            $phone = $_POST['phone'];
            $sql="UPDATE `student` SET `phone` = '$phone' WHERE `student`.`student_id` = $id;";
            if($con->query($sql) == true){
                $flag='T';
            }
        }
        if( !empty($_POST['class'])){
            $class_id = $_POST['class'];
            $sql="UPDATE `student` SET `class_id` = '$class_id' WHERE `student`.`student_id` = $id;";
            if($con->query($sql) == true){
                $flag='T';
            }
        }
<?php foreach ($results as $res) : ?>
                <option name="id" value=<?php echo $res['student_id'];?>><?php echo $res['student_id'].' -   [ '.$res['name'].' ]' ; ?></option>
                <?php endforeach ?>
<?php
            if(isset($_POST["up"])){
                $sel_id=$_POST['id'];
```

```
          $tsql="SELECT * FROM `student` WHERE `student_id` = $sel_id";
          $tres=mysqli_query($con, $tsql);
          $ans=mysqli_fetch_assoc($tres);
        }
      ?>

<?php
          if($ans['class_id']==1){
            $cid="8th a";
          }
          elseif($ans['class_id']==2)
            $cid="8th b";
          elseif($ans['class_id']==3)
            $cid="8th c";
          elseif($ans['class_id']==4)
            $cid="9th a";
          elseif($ans['class_id']==5)
            $cid="9th b";
          elseif($ans['class_id']==6)
            $cid="9th c";
          elseif($ans['class_id']==7)
            $cid="10th a";
          elseif($ans['class_id']==8)
            $cid="10th b";
          elseif($ans['class_id']==9)
            $cid="10th c";
        ?>

        <p><?php echo 'class : '.$cid; ?> </p>
        <div class="op">
          <label for="class">Update students class</label>
          <select name="class" id="class">
            <option name="class" value="" selected disabled hidden>Update
class</option>
            <option name="class" value="1">8th 'A'</option>
            <option name="class" value="2">8th   'B'</option>
            <option name="class" value="3">8th   'C'</option>
            <option name="class" value="4">9th   'A'</option>
            <option name="class" value="5">9th  'B'</option>
            <option name="class" value="6">9th  'C'</option>
            <option name="class" value="7">10th  'A'</option>
            <option name="class" value="8">10th  'B'</option>
            <option name="class" value="9">10th  'C'</option>
          </select>
        </div>

        <button class="btn" value= <?php echo $sel_id; ?> name="sub">Submit</button> <br>
      <?php endif ?>
```

### 3.4.6 Add teacher

```
$sql="INSERT INTO `teacher` (`name`, `gender`, `phone`, `date_joined`, `class_id`) VALUES ('$name', '$gender', '$phone', '$date', '$class_id');";

if (empty($name)) { array_push($errors, "Teacher's name is required !!"); }
if (empty($date)) { array_push($errors, "Teacher's start_date is required !!"); }
if (empty($phone)) { array_push($errors, "Teacher's Phone Number is required !!"); }

if (count($errors) == 0){
  if($con->query($sql) == true){
    $flag='T';
  }
}
```

### 3.4.7 View teacher

```
  $sql="SELECT * FROM `teacher`;";
$results = mysqli_query($con, $sql);
```

```
    foreach ($results as $x){
        $c += 1;
    }
<?php foreach ($results as $res) : ?>
            <tr>
                <td> <?php echo $res['teacher_id']."  ";  ?> </td>
                <td> <?php echo $res['name']."  "; ?> </td>
                <td> <?php echo $res['gender']."  ";  ?> </td>
                <td> <?php echo $res['phone']."  ";  ?> </td>
                <td> <?php echo $res['date_joined']."  ";  ?> </td>



                <?php
                    $id=$res['class_id'];
                    $sql2="SELECT * FROM `class` WHERE `class_id` = $id;";
                    $results2 = mysqli_query($con, $sql2);
                    $res2 = mysqli_fetch_assoc($results2);
                ?>

                <td> <?php echo $res2['class_name']."  ";  ?> </td>
                <td> <?php echo $res2['section']."  ";  ?> </td>
            </tr>
        <?php endforeach ?>
```

### 3.4.8 Update teacher

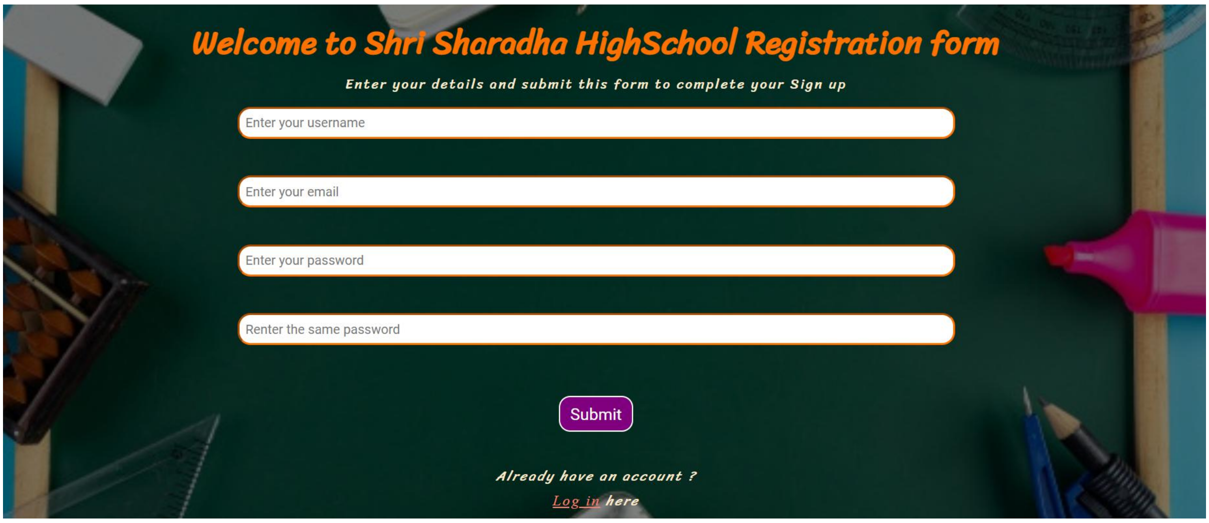
```
       if(isset($_POST['sub'])){
     $id = $_POST['sub'];

     if( !empty($_POST['name'])){
        $name = $_POST['name'];
        $sql="UPDATE `teacher` SET `name` = '$name' WHERE
`teacher`.`teacher_id` = $id;";
        if($con->query($sql) == true){
            $flag='T';
          }
        }
        if( !empty($_POST['date'])){
          $date = date('Y-m-d', strtotime($_POST['date']));
          echo $date;
          $sql="UPDATE `teacher` SET `date_joined` = '$date' WHERE
`teacher`.`teacher_id` = $id;";
          if($con->query($sql) == true){
            $flag='T';
          }
        }
        if( !empty($_POST['gender'])){
          $gender = $_POST['gender'];
          $sql="UPDATE `teacher` SET `gender` = '$gender' WHERE
`teacher`.`teacher_id` = $id;";
          if($con->query($sql) == true){
            $flag='T';
          }
        }
        if( !empty($_POST['phone'])){
          $phone = $_POST['phone'];
          $sql="UPDATE `teacher` SET `phone` = '$phone' WHERE
`teacher`.`teacher_id` = $id;";
          if($con->query($sql) == true){
            $flag='T';
          }
        }
        if( !empty($_POST['class'])){
          $class_id = $_POST['class'];
          $sql="UPDATE `teacher` SET `class_id` = '$class_id' WHERE
`teacher`.`teacher_id` = $id;";
          if($con->query($sql) == true){
            $flag='T';
          }}}
```

## Chapter 04

# RESULT ANALYSIS

## 4.1 Snapshots



Welcome to Shri Sharadha HighSchool Registration form

Enter your details and submit this form to complete your Sign up

Enter your username

Enter your email

Enter your password

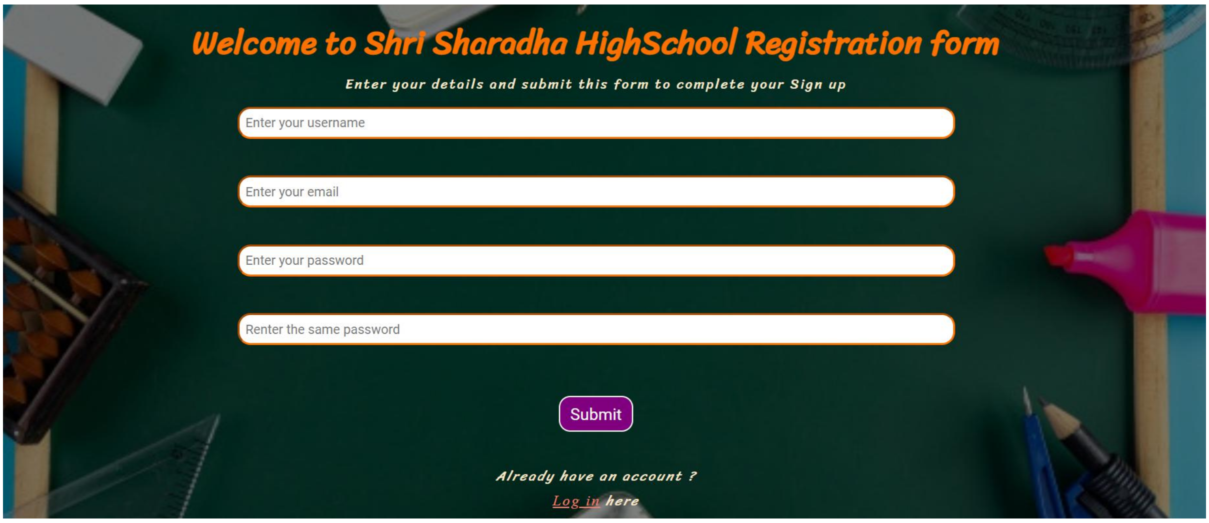
Renter the same password

Submit

Already have an account ?

Log in here

Fig 4.1 Login



Welcome to Shri Sharadha HighSchool Registration form

Enter your details and submit this form to complete your Sign up

Enter your username

Enter your email

Enter your password

Renter the same password

Submit

Already have an account ?

Log in here

Fig 4.1 Register

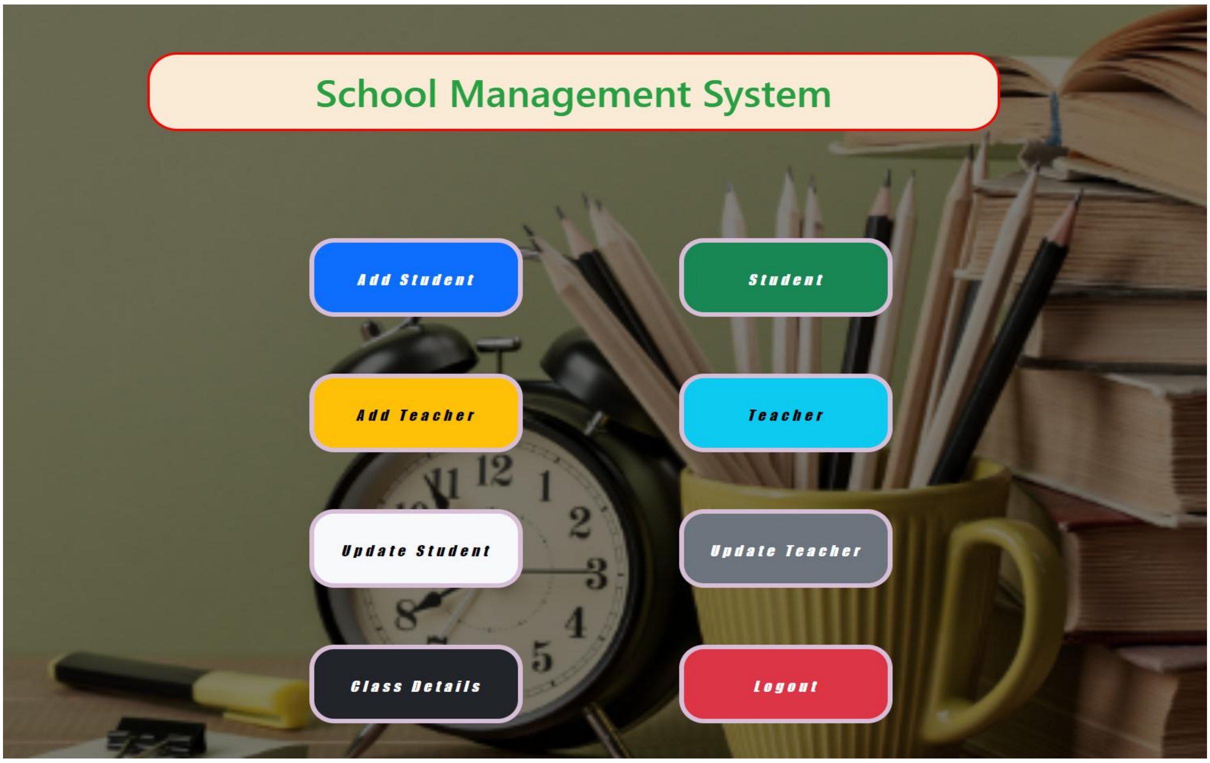
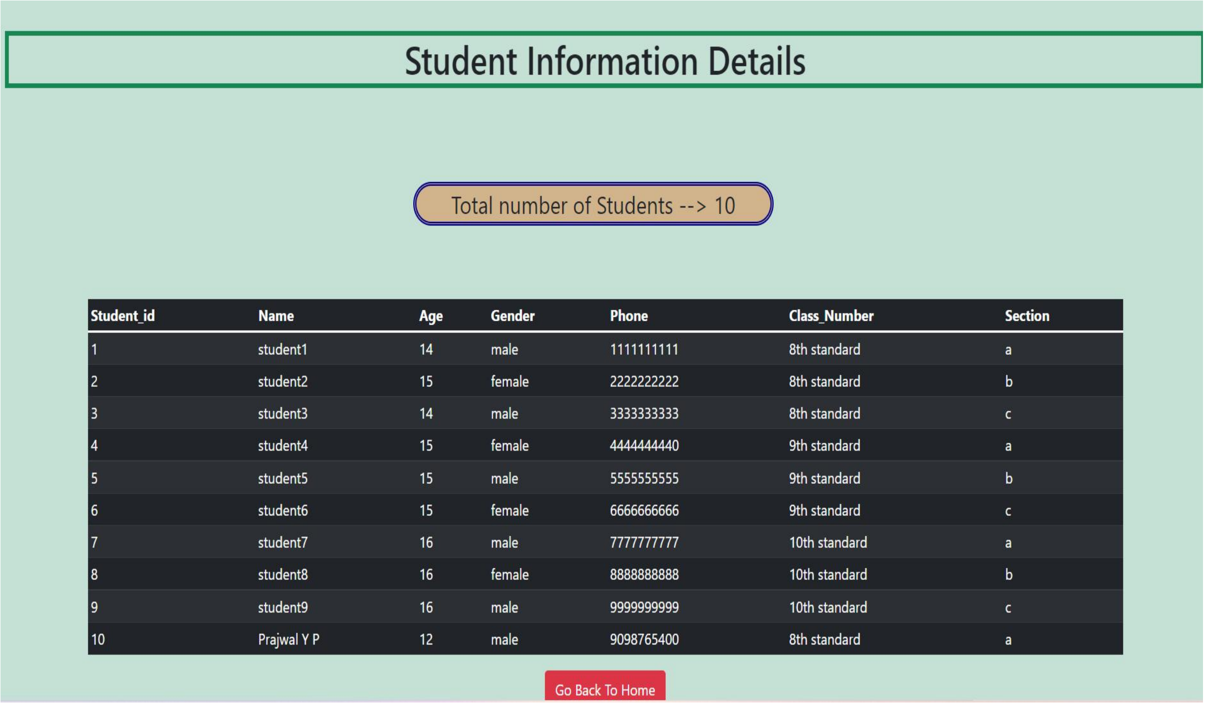


Fig 4.3 Dashboard

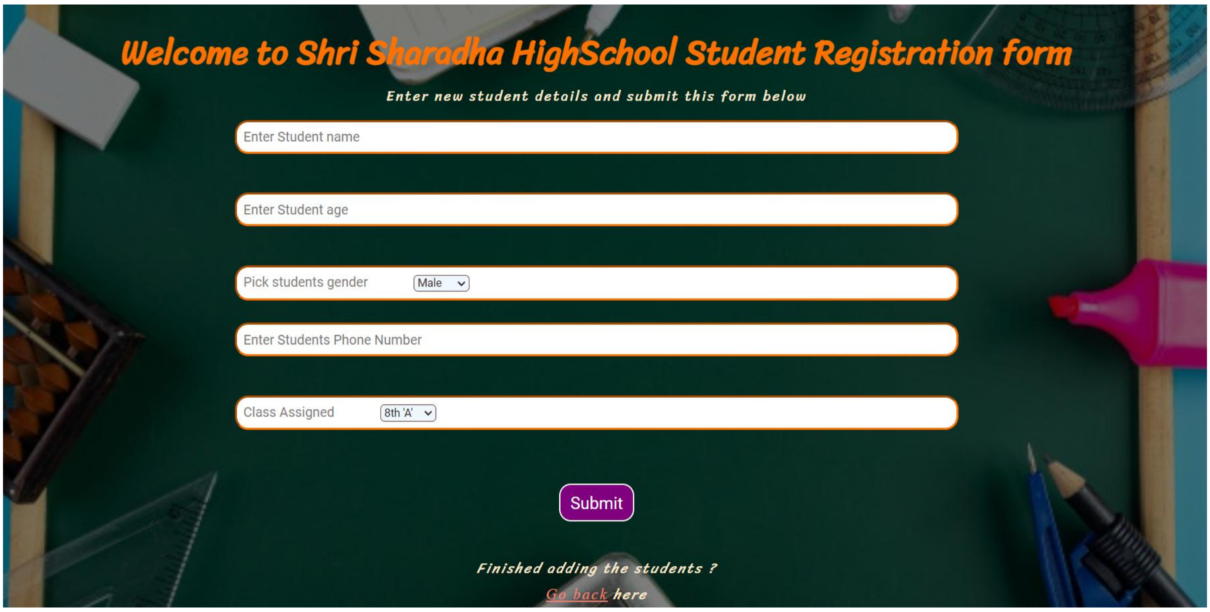


Student Information Details

Total number of Students --> 10

| Student_id | Name | Age | Gender | Phone | Class_Number | Section |
|---|---|---|---|---|---|---|
| 1 | student1 | 14 | male | 1111111111 | 8th standard | a |
| 2 | student2 | 15 | female | 2222222222 | 8th standard | b |
| 3 | student3 | 14 | male | 3333333333 | 8th standard | c |
| 4 | student4 | 15 | female | 4444444440 | 9th standard | a |
| 5 | student5 | 15 | male | 5555555555 | 9th standard | b |
| 6 | student6 | 15 | female | 6666666666 | 9th standard | c |
| 7 | student7 | 16 | male | 7777777777 | 10th standard | a |
| 8 | student8 | 16 | female | 8888888888 | 10th standard | b |
| 9 | student9 | 16 | male | 9999999999 | 10th standard | c |
| 10 | Prajwal Y P | 12 | male | 9098765400 | 8th standard | a |

Fig 4.4 Student detail



Welcome to Shri Sharadha HighSchool Student Registration form

Enter new student details and submit this form below

Enter Student name

Enter Student age

Pick students gender Male

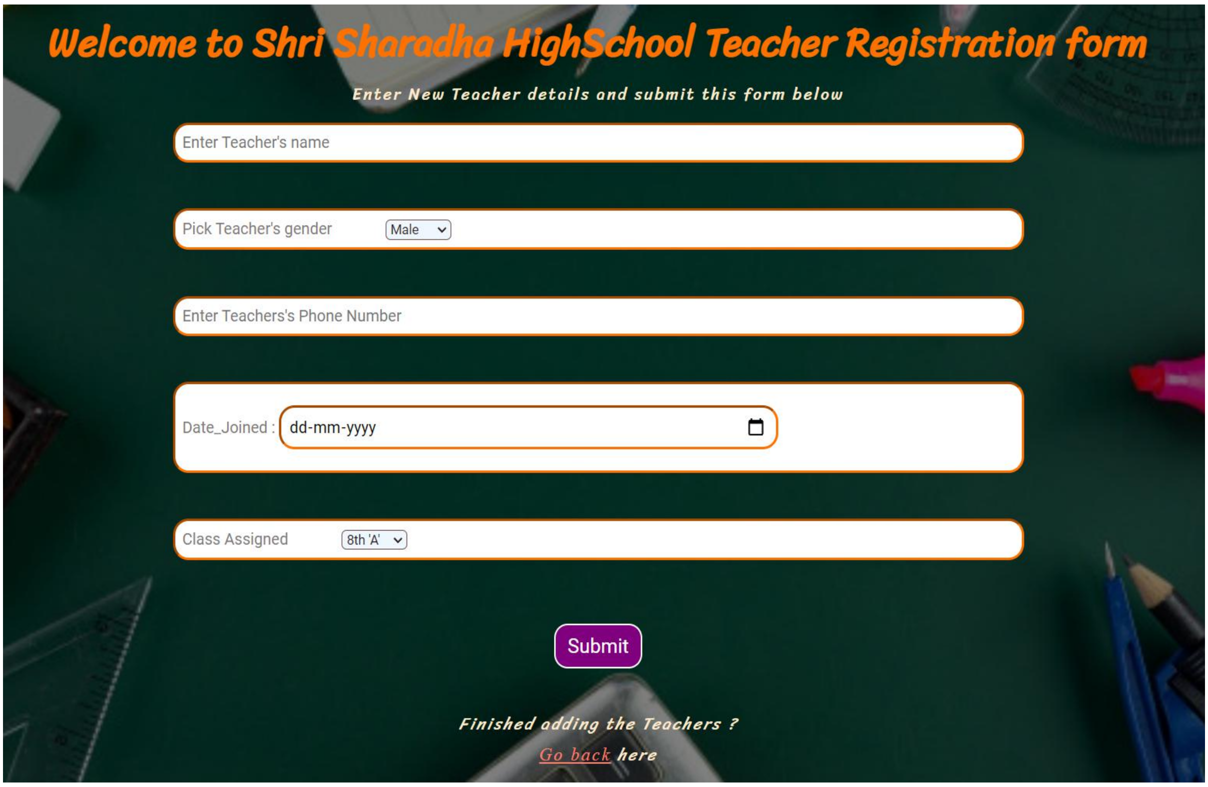
Enter Students Phone Number

Class Assigned 8th 'A'

Finished adding the students ?

Go back here

Fig 4.5 Student Registration



Welcome to Shri Sharadha HighSchool Teacher Registration form

Enter New Teacher details and submit this form below

Enter Teacher's name

Pick Teacher's gender Male

Enter Teachers's Phone Number

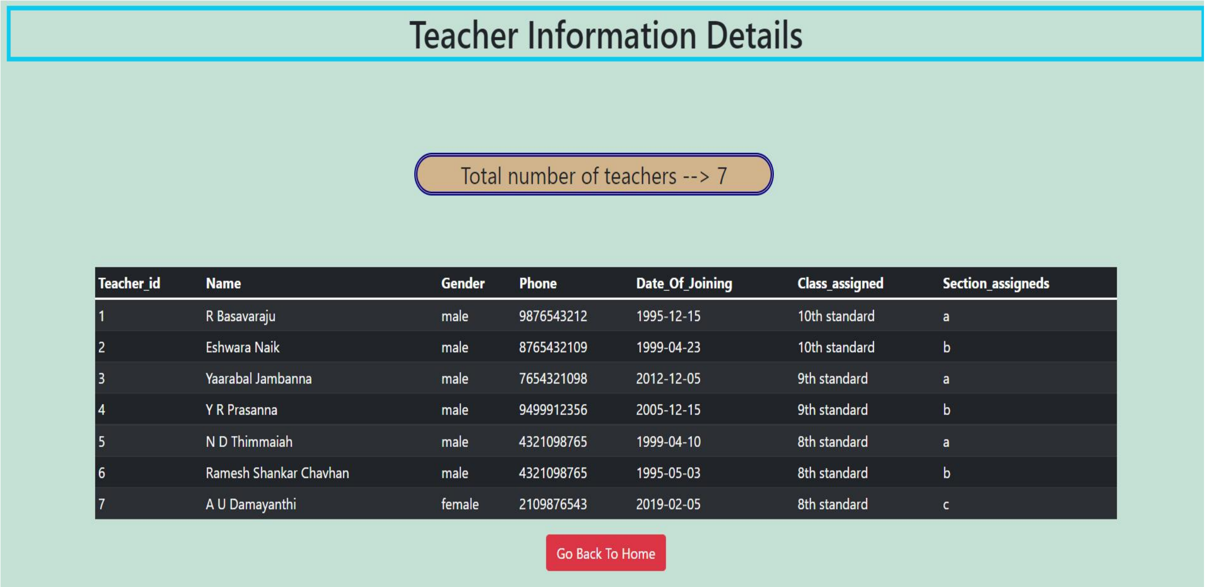
Date_Joined : dd-mm-yyyy

Class Assigned 8th 'A'

Finished adding the Teachers ?

Go back here

Fig 4.6 Teacher registration

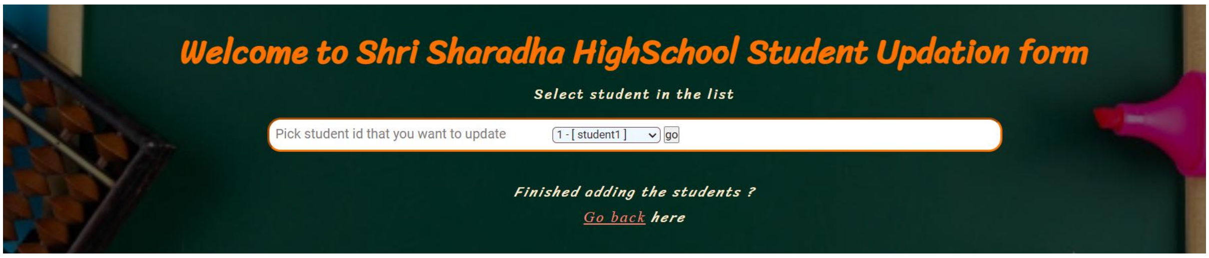


Teacher Information Details

Total number of teachers --> 7

| Teacher_id | Name | Gender | Phone | Date_Of_Joining | Class_assigned | Section_assigneds |
|---|---|---|---|---|---|---|
| 1 | R Basavaraju | male | 9876543212 | 1995-12-15 | 10th standard | a |
| 2 | Eshwara Naik | male | 8765432109 | 1999-04-23 | 10th standard | b |
| 3 | Yaarabal Jambanna | male | 7654321098 | 2012-12-05 | 9th standard | a |
| 4 | Y R Prasanna | male | 9499912356 | 2005-12-15 | 9th standard | b |
| 5 | N D Thimmaiah | male | 4321098765 | 1999-04-10 | 8th standard | a |
| 6 | Ramesh Shankar Chavhan | male | 4321098765 | 1995-05-03 | 8th standard | b |
| 7 | A U Damayanthi | female | 2109876543 | 2019-02-05 | 8th standard | c |

Fig 4.7 Teacher detail



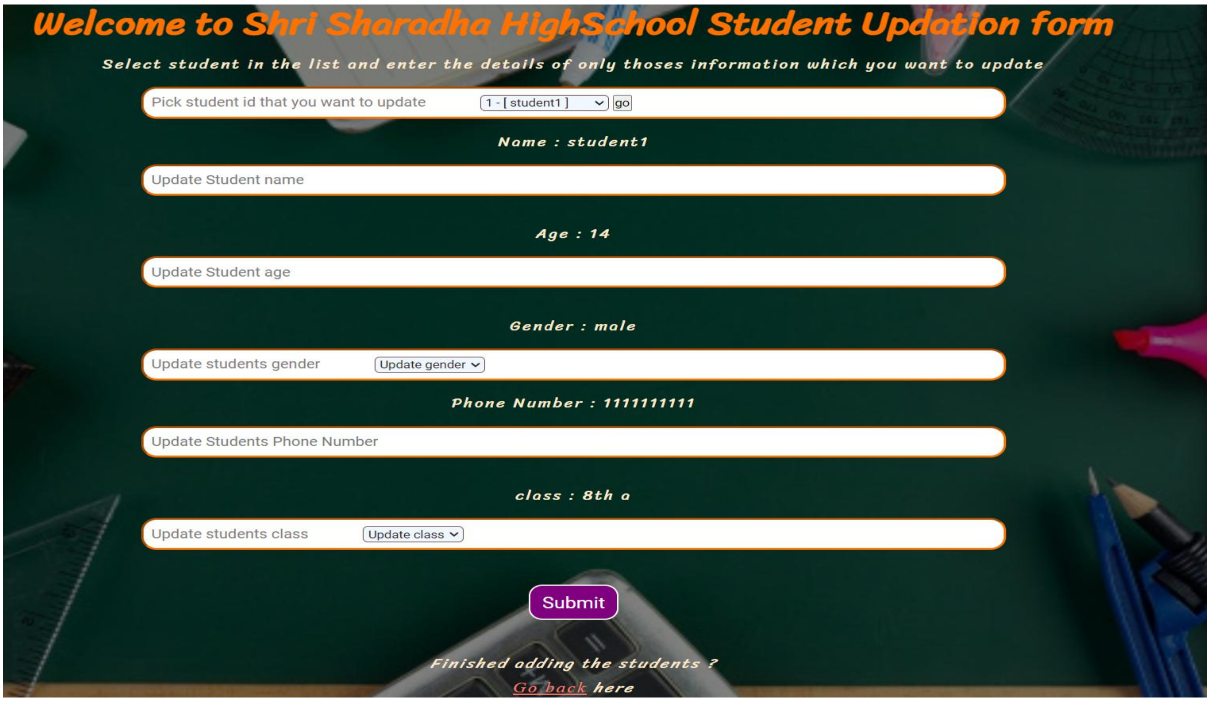
Welcome to Shri Sharadha HighSchool Student Updation form

Select student in the list

Pick student id that you want to update 1 - [ student1 ] go

Finished adding the students ?
Go back here

Fig 4.8 Select Student update



Welcome to Shri Sharadha HighSchool Student Updation form

Select student in the list and enter the details of only thoses information which you want to update

Pick student id that you want to update 1 - [ student1 ] go

Name : student1

Update Student name

Age : 14

Update Student age

Gender : male

Update students gender Update gender

Phone Number : 1111111111

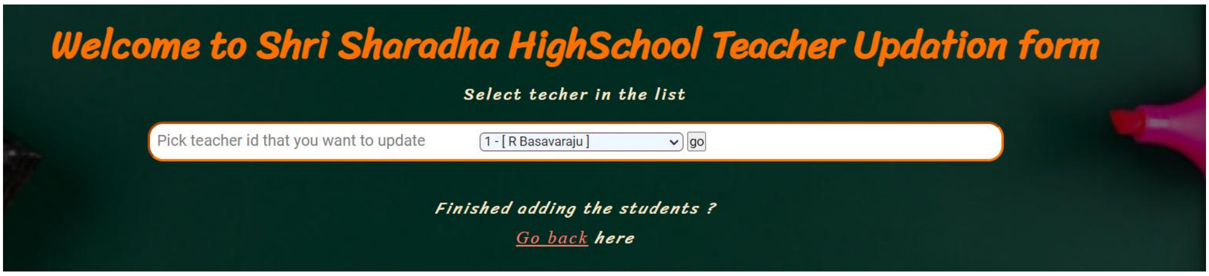
Update Students Phone Number

class : 8th a

Update students class Update class

Finished adding the students ?
Go back here

Fig 4.9 Student update


Welcome to Shri Sharadha HighSchool Teacher Updation form

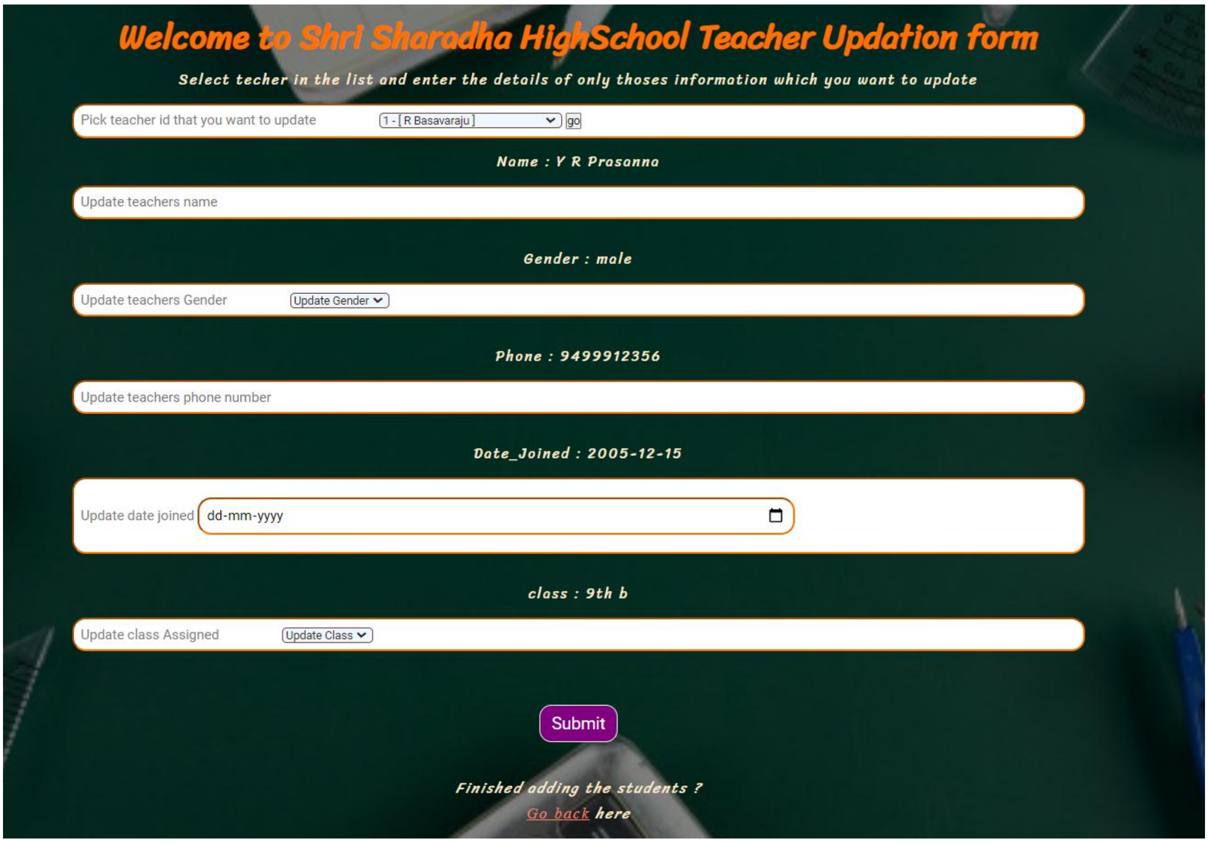
Select techer in the list

Pick teacher id that you want to update 1 - [ R Basavaraju ] go

Finished adding the students ?

Go back here

Fig 4.10 Select teacher update


Welcome to Shri Sharadha HighSchool Teacher Updation form

Select techer in the list and enter the details of only thoses information which you want to update

Pick teacher id that you want to update 1 - [ R Basavaraju ] go

Name : Y R Prasanna

Update teachers name

Gender : male

Update teachers Gender Update Gender

Phone : 9499912356

Update teachers phone number

Date_Joined : 2005-12-15

Update date joined dd-mm-yyyy

class : 9th b

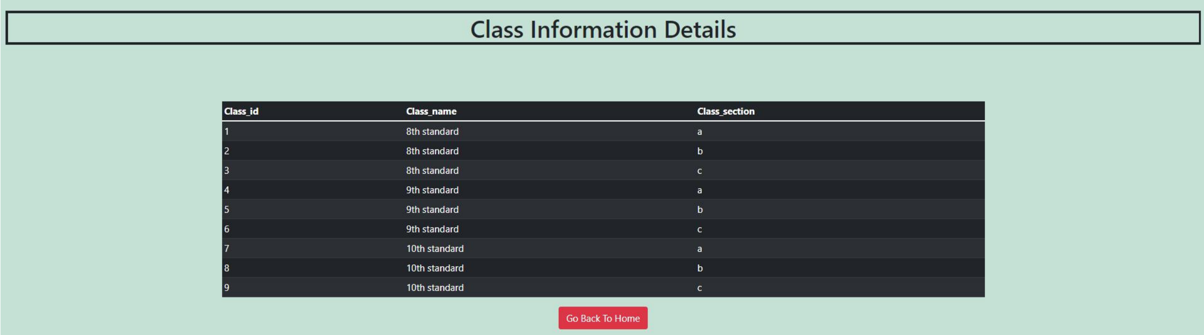
Update class Assigned Update Class

Submit

Finished adding the students ?

Go back here

Fig 4.11 Teacher update


Class Information Details

| Class_id | Class_name | Class_section |
|---|---|---|
| 1 | 8th standard | a |
| 2 | 8th standard | b |
| 3 | 8th standard | c |
| 4 | 9th standard | a |
| 5 | 9th standard | b |
| 6 | 9th standard | c |
| 7 | 10th standard | a |
| 8 | 10th standard | b |
| 9 | 10th standard | c |

Go Back To Home

Fig 4.12 Class detail

## 4.2 Discussion

- Registration Page :

In this page we have provides Friendly modern User-Interface for the user to fill their details in order to get registered.

Following are the details that are accounted:

1. Name
2. Email_ID
3. Password

- Login Page :

In this Page the user-interface is designed in such a way that if user is not registered he is taken into HOME PAGE, if REGISTRED by checking "EMAIL" and "PASSWORD" , user is sent to "DASHBOARD PAGE".

- Home Page :

This user-interface , we can say that it I designed for guest-users or NON_LOGIN USERS.

Guest-Users are provide with these many following options

1. Access STUDENT PAGE
2. Access TEACHER PAGE
3. Access CLASS PAGE

- Dashboard :

This user-interface , we can say that it I designed for Authenticated-users or LOGIN USERS.

Authenticated-Users are provide with these many following options

1. Access STUDENT PAGE
2. Access TEACHER PAGE
3. Access CLASS PAGE
4. Access ADD STUDENT PAGE
5. Access ADD TEACHER PAGE

6. Access UPDATE STUDENT PAGE
7. Access UPDATE TEACHER PAGE

- Student Page :

This UI give user the information of all the students present in the school along with the details in the form of classy tables.

- Teachers Page :

This UI give user the information of all the teachers present in the school along with the details in the form of classy tables.

- Add Student Page :

This UI interface lets user add new student detail to this educational institution database for future references in the form of table.

- Add Teachers Page:

This UI interface lets user add new teacher detail to this educational institution database for future references in the form of table.

- Update Student Page :

This UI interface lets user update/modify/alter existing student detail to this educational institution database for future references in the form of table.

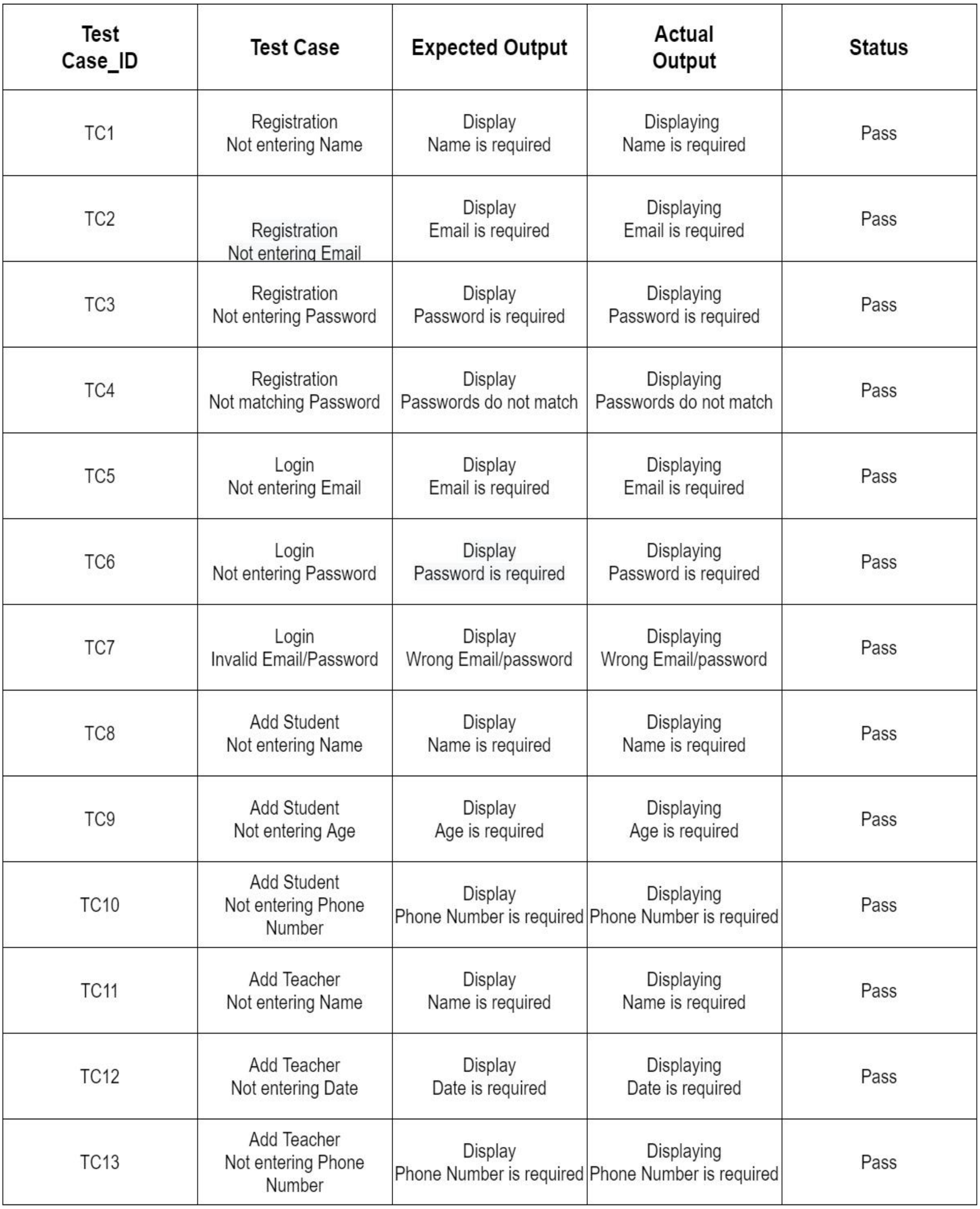
- Update Teacher Page :

This UI interface lets user update/modify/alter existing teacher detail to this educational institution database for future references in the form of table.

- Class Page :

This UI give user the information of all the classes present in the school along with the sub-classes details in the form of trendy tables.

## 4.3 Testing



| Test Case_ID | Test Case | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| TC1 | Registration Not entering Name | Display Name is required | Displaying Name is required | Pass |
| TC2 | Registration Not entering Email | Display Email is required | Displaying Email is required | Pass |
| TC3 | Registration Not entering Password | Display Password is required | Displaying Password is required | Pass |
| TC4 | Registration Not matching Password | Display Passwords do not match | Displaying Passwords do not match | Pass |
| TC5 | Login Not entering Email | Display Email is required | Displaying Email is required | Pass |
| TC6 | Login Not entering Password | Display Password is required | Displaying Password is required | Pass |
| TC7 | Login Invalid Email/Password | Display Wrong Email/password | Displaying Wrong Email/password | Pass |
| TC8 | Add Student Not entering Name | Display Name is required | Displaying Name is required | Pass |
| TC9 | Add Student Not entering Age | Display Age is required | Displaying Age is required | Pass |
| TC10 | Add Student Not entering Phone Number | Display Phone Number is required | Displaying Phone Number is required | Pass |
| TC11 | Add Teacher Not entering Name | Display Name is required | Displaying Name is required | Pass |
| TC12 | Add Teacher Not entering Date | Display Date is required | Displaying Date is required | Pass |
| TC13 | Add Teacher Not entering Phone Number | Display Phone Number is required | Displaying Phone Number is required | Pass |

Table 4.1 Test Cases

## Chapter 05

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

The project work titled “School Management System” has been designed using PHP Hypertext [PHP] where in many user friendly form controls have been added in order to make it a user interactive application. The system is developed in such a way that the user with common knowledge of computers can handle it easily. The System developed has proved to be user friendly and efficient in achieving basic goals. The system takes care of all the constrains which have specified. The system is found to be really beneficial for the concerned aspects. Application developed is realistic and secure.

The main idea was to develop an easy to use front end using concept of HTML, css, JAVA SCRIPT, and AJAX. All the possible user requests are accomplished buttons. Up to date reports can be generated on user requests. All the validations have been done.

## 6.2 Future Work

In future days, I thought to improve this project by adding more functionality like inserting Management of Courses, Marks_Analysis, Attendance_Analysis and many more.

Also have a idea in Displaying the latest updated notices to the users with the related School Memo’s and Overall School grade with respect to academical year.

## Chapter 06

# REFERENCES

1. HTML Tutorial :
    - https://www.youtube.com/watch?v=BsDoLVMnmZs
    - https://www.youtube.com/watch?v=qz0aGYrrlhU
    - https://www.w3schools.com/html/
    - https://www.tutorialspoint.com/html/index.htm

2. CSS Tutorial :
    - https://www.youtube.com/watch?v=u5-K_ua9sOw
    - https://www.youtube.com/watch?v=4ykmsTpIn08
    - https://www.w3schools.com/css/
    - https://www.javatpoint.com/css-tutorial

3. PHP Tutorial :
    - https://www.youtube.com/playlist?list=PLu0W_9lII9aikXkRE0WxDt1vozo3hnmtR
    - https://www.youtube.com/watch?v=1SnPKhCdlsU&t=8049s
    - https://www.w3schools.com/php/
    - https://www.javatpoint.com/php-tutorial