1. INTRODUCTION

Nowadays importance is given to the wireless technology and effective system. Computers and technology have become part of our life for accessing almost everything we basically do. Life now is full of indispensable technological advancement and in this technological era it is very difficult for any organization to survive without technology. The World Wide Web contributes enormously to the creation of an ever-increasing global information database. It could also be used as a system to share information within an enterprise. This led to a unique web-based attendance management system developed specifically to support the attendance systems in colleges. Attendance management system is an online application uses Quick Responsive codes to mark the attendance. This system helps the teachers to mark the attendance without much of manual work like calling out names of students or to mark them manually in the books. It helps in effective and efficient utilization of the hardware and the software resources.

The proposed system is an application that has been developed for marking and maintaining the student attendance simplifying the manual work by automating the process of marking the attendance. The existing systems consist of manually calling out of names which would be consuming more time which could lead to marking of proxy attendance in a huge crowd. The other systems which would use biometrics or semi-conductor chips to mark attendance are expensive since the shortage of the semi-conductor chips in recent years. This could be overcome by automating the system with only software technologies that could be utilized by everyone. Hence, this application is more efficient, user-friendly for marking and managing the attendance by saving time.

1.1. PROBLEM STATEMENT

Student attendance has become one of the concerns at colleges in recent years. Teachers marking the attendance of students manually has become a tedious job and time consuming. The existing systems consist of manually calling out of names which would be consuming more time which could lead to marking of proxy attendance in a huge crowd. The other systems which would use biometrics or semi-conductor chips to mark attendance are expensive since the shortage of the semi-conductor chips in recent years.

1.2 GOALS & OBJECTIVES

The goal of the project is to design an application that can automate the process of marking the attendance of the students using the devices that could be accessible by everyone which would be cost-effective for the institutions. The proposed system shall provide an efficient and user-friendly platform for teachers as well as students. Hence, this system minimizes the limitation of the existing system saving time and displaying accurate attendance details to the students and teachers.

1.3 MOTIVATION

In this generation there are many applications that are made for betterment of our day-to-day life for simplifying time consuming works. “Attendance Management System using QR-code Verification” is one such application which was developed to simplify the work of students and faculties by providing a user-friendly interface and helping them to solve some of the time-consuming work such as marking of attendance manually, computing the percentage of attendance and viewing the same using the developed system. All these works are done easily as the application marks and computes attendance automatically as and when the QR code is scanned by both student and teacher respectively which reduces their time consumption.

One of the major concerns in this generation is the accuracy of data. This application helps in providing the accurate details of attendance to students and faculties. These features helped us design this application in a way which is beneficial for every user to save their time by making use of an interactive and user-friendly application.

1.4 APPLICATION

- Using this application, the attendance of the students can be marked automatically.
- The student can track his attendance details and manage it well using the application.
- The false marking of attendance can be avoided by since no more than one device can be used for logging in to the application.
- The students can view their attendance details and manage up the same.
- The faculties can view the attendance details of the students of the respective subject they conduct.
- The classroom module automatically generates QR codes as soon as we log in to the classroom module.
- The QR code generated will be automatically based on the room number and a unique string which is generated when we log in to the classroom.
- The admin can avoid any misuse of the application has he will be having the complete authorization of the application.

1.5 OVERVIEW OF TECHNICAL AREA

We are using MySql as our database since we required Relational Database for this application as it was cost effective. The data in the database can be managed easily as it stores them in the table form unlike other latest databases which stores data as key value pair which is difficult to manage and is much costlier than Relational database. Hence, we are using MySql rather than other advanced databases like MongoDb.

1.6 DATABASE USED

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.

1.7 BRIEF OVERVIEW

The proposed system is aimed to automate the whole process by marking the attendance of the students. The system is efficient enough to mark the attendance of the students to the respective subjects that are handled by the respective teachers only. To avoid the proxy of the attendance by students the proposed system uses Media Access Control (MAC) address to avoid the problem helping the software to be efficient. The students can easily manage the attendance by viewing the details through the application. The teachers can also view the attendance details of the students for the respective subject they take. The system has been developed to be user-friendly, smooth and feasible for the users from the beginning.

2. LITERATURE SURVEY

Student attendance has become one of the concerns at colleges in recent years. Teachers marking the attendance of students manually has become a tedious job and time consuming.[1] The other way is by introducing recourses such as semi-conductor chips in the student’s identity card which are not cost-effective as semi-conductor chips put in extra cost. Along with that there has been a shortage of semiconductor chips in recent years and post pandemic the technology has evolved a lot due to online classes and many more.[2] To overcome, we need a cost-effective and an automated attendance management system is required. Attendance Management System Using QR-Code Verification Technique is one such online application developed for the ease of marking the attendance of the students. The other QR-code based attendance systems use other approaches by combining the QR-codes with Global Positioning System (GPS)[3] or Facial recognition which would again be expensive for the software since storing them and fetching accurate results would lead to have high end servers which would be expensive. The proposed system is one of the effective, time saving and cost-effective software developed. The software uses Quick Responsive codes (QR codes) to mark the attendance of students by reducing the manual work. The problems of a student creating clone version of app or creating parallel space to log in to other student’s account to mark proxy can arise. Another problem that might arise in the existing systems are that the QR codes can be shared among the students due to which the students can mark their attendance from anywhere even though they are not attending the classes. The proposed application comes with advanced features where all these problems are resolved, assuring the application to be dynamic and safe. The students can even view and maintain their attendance from this application by attending classes. This makes the system efficient in speeding up the process. The lecturer can even track the student attendance easily with the application round the clock. This application has been made completely smooth and responsive to the user making it feasible from the beginning.

3. SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

- 2 GB RAM
- HDD 128 GB Hard Disk Space or Above

3.2 SOFTWARE REQUIREMENTS

- Operating System:
  - Windows 7 or above.
  - Android 5.1 or above.
- Browser.

3.3 NON-FUNCTIONAL REQUIREMENTS

- Face Recognition: To implement face recognition, which improves security against proxy.

3.4 FUNCTIONAL REQUIREMENTS

- Automated QR Code Generator:
  - Generate random string.
  - Generate QR code based on classroom number and random string.
  - Display the QR code.

3.5 COST OVERVIEW

The cost overview is used to define the amount of cost that is expected to be for creating and running the application. The cost overview for this project is:

Hosting: 2000Rs.

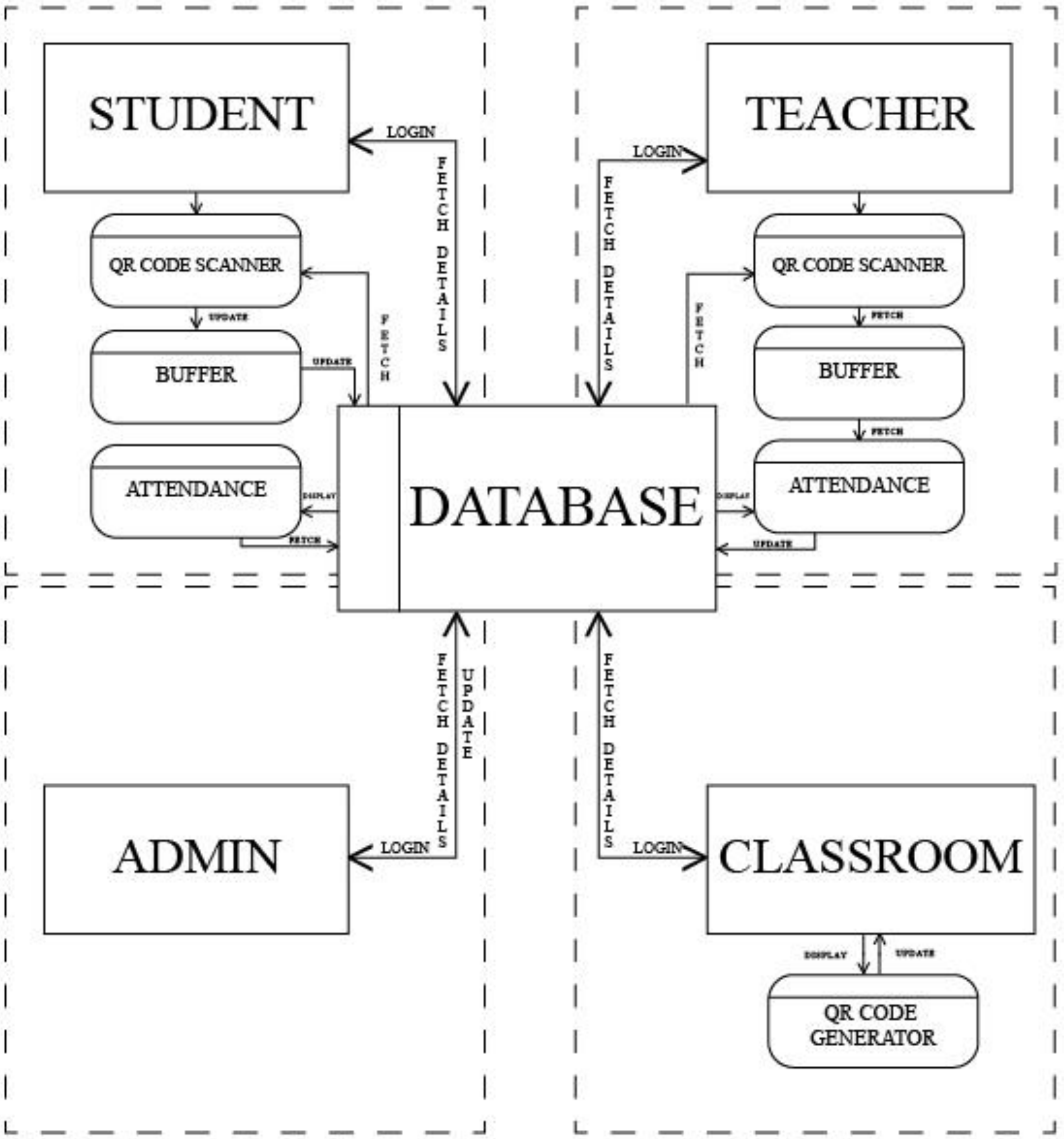
4. SYSTEM ARCHITECTURE

The System Architecture basically defines the way the application has been designed. Our application is designed User-friendly and in a way such that, the user using the application for the first time would find the application very simple. The administrator will have complete control over the database. The proposed system consists of 4 modules. They are:

- Student Module: In this module, students have to scan the QR-codes displayed on the classroom entrance in order to mark their attendance. They can even view their attendance details in this module. This module has 3 more sub modules:
  - QR Code Scanner Module: This module facilitates the scanning of QR codes to record student attendance.
  - Buffer Module: This module is used to generate a resume automatically based on the details provided by the students on their profile. The recruiters who would want to check the student’s details will be provided directly with student’s resume.
  - Attendance Module: Once the QR code is scanned, this module records student attendance.
- Teacher Module: In this module, teachers can mark the attendance of the students and can view the student attendance of their respective subject. This module has 3 more sub modules:
  - QR Code Scanner Module: In this module, after the teacher scans the QR code, the student can scan the QR code to mark his or her attendance.
  - Buffer Module: This module is used to generate a resume automatically based on the details provided by the students on their profile. The recruiters who would want to check the student’s details will be provided directly with student’s resume.
  - Attendance Module: This module collects students who are present after they mark their attendance.

- Classroom Module: This module generates a QR-code which can be scanned by students and teachers to mark their attendance. This module has 1 sub modules:
    - QR Code Generator Module: As soon as the user logs in, this module generates the QR code.
- Admin Module: This module has the access for the administrator of the application where they have complete authorization over all the users of the application. He can control the complete application and can perform any of the operations on any of the module.

4.1 DATA FLOW DIAGRAM



This diagram represents the complete flow of data by the application which includes various operations such as fetching, updating and displaying the details from the database.

5. IMPLEMENTATION

The project has been developed with HTML, CSS, Java script and Bootstrap for developing the front-end. The MySql database has been used as the backend database to store the data. To connect the front-end with the back-end we have used Flask which is a framework of python.

The database management system handles requests generated from the MySql interface, producing or modifying of data in response to these requests.

5.1 TABLE CREATION

USERS:

```
CREATE TABLE `users` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `role` int NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

STUDENTS:

```
CREATE TABLE `students` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(245) NOT NULL,
  `email` varchar(45) NOT NULL,
  `phone_number` varchar(13) NOT NULL,
  `course` varchar(45) NOT NULL,

  `semester` varchar(45) NOT NULL,

  `student_mac` varchar(245) DEFAULT NULL,

  `password` varchar(245) NOT NULL,

  PRIMARY KEY (`id`),

  UNIQUE KEY `email_UNIQUE` (`email`),

  UNIQUE KEY `id_UNIQUE` (`id`),

  UNIQUE KEY `phone_number_UNIQUE` (`phone_number`)

) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

TEACHER:

```
CREATE TABLE `teacher` (

  `id` int NOT NULL AUTO_INCREMENT,

  `name` varchar(45) NOT NULL,

  `email_id` varchar(45) NOT NULL,

  `phone_no` varchar(45) NOT NULL,

  `qualification` varchar(45) NOT NULL,

  `course` varchar(45) NOT NULL,

  `subject` varchar(45) NOT NULL,

  `password` varchar(245) NOT NULL,

  PRIMARY KEY (`id`),

  UNIQUE KEY `id_UNIQUE` (`id`),

  UNIQUE KEY `email_UNIQUE` (`email_id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

CLASSROOM:

```
CREATE TABLE `classroom` (

  `id` int NOT NULL AUTO_INCREMENT,

  `room_no` varchar(45) NOT NULL,

  `password` varchar(245) NOT NULL,

  `qrvalue` varchar(45) NOT NULL,

  PRIMARY KEY (`id`),

  UNIQUE KEY `id_UNIQUE` (`id`),

  UNIQUE KEY `room_no_UNIQUE` (`room_no`)

) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

5.2 TABLE VALUES INSERTION

USERS:

```
INSERT INTO `users` VALUES
(1,'Admin','admin@email.com','f3727763aeb0eb73cdfa3e90a4f23b7a',1);
```

STUDENTS:

```
INSERT INTO `student` VALUES (1,'Dhanush Raju S
S','dhanush@gmail.com','1234567890','BCA','6',NULL,'900150983cd24fb0d6963f7d28e17f7
```

2'),(2,'manoj','manoj@gmail.com','9876543103','BCA','6',NULL,'5e81f9859d223ea420aca99 3c647b839');

TEACHER:

INSERT INTO `teacher` VALUES (1,'Kiran Kumar MN','kiran@gmail.com','2354678910','MCA','BCA','Unix & Shell Programming','39c63ddb96a31b9610cd976b896ad4f0'),(2,'Roopashree CS','roopashree@gmail.com','9875463834','B.Tech','BCA','Software Testing','04a17a7617eefaa232c6aa8a0092fe81');

CLASSROOM:

INSERT INTO `room` VALUES (1,'101','38b3eff8baf56627478ec76a704e9b52','vuo');

5.3 SOURCE CODE:

PYTHON CODE:

```
import hashlib

from flask import Flask, render_template, request, redirect, session, url_for

from flaskext.mysql import MySQL

from functools import wraps

from werkzeug.security import generate_password_hash, check_password_hash

from handler import *



app = Flask(__name__)

app.secret_key = 'Mage is the best!'



# MySQL
```

```
mysql = MySQL()

app.config['MYSQL_DATABASE_USER'] = "root"

app.config['MYSQL_DATABASE_PASSWORD'] = "root"

app.config['MYSQL_DATABASE_DB'] = "square"

app.config['MYSQL_DATABASE_HOST'] = "localhost"

mysql.init_app(app)




def login_required(f):

    @wraps(f)

    def wrapped(*args, **kwargs):

        if 'authorised' not in session:

            return render_template('login.html')

        return f(*args, **kwargs)

    return wrapped




@app.context_processor

def inject_tables_and_counts():

    data = count_all(mysql)

    return dict(tables_and_counts=data)
```

```
@app.route('/')

@app.route('/index')

@login_required

def index():

        return render_template('index.html')




@app.route("/attendence")

@login_required

def attendence():

        data = fetch_all(mysql, "attendence")

        return render_template('attendence.html', data=data, table_count=len(data))




@app.route('/edit_attendence/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required

def edit_attendence(modifier_id, act):

        if act == "add":

                return render_template('edit_attendence.html', data="", act="add")

        else:

                data = fetch_one(mysql, "attendence", "id", modifier_id)



                if data:
```

```
            return render_template('edit_attendence.html', data=data, act=act)

        else:

            return 'Error loading #%s' % modifier_id




@app.route("/buffer")

@login_required

def buffer():

    data = fetch_all(mysql, "buffer")

    return render_template('buffer.html', data=data, table_count=len(data))




@app.route('/edit_buffer/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required

def edit_buffer(modifier_id, act):

    if act == "add":

        return render_template('edit_buffer.html', data="", act="add")

    else:

        data = fetch_one(mysql, "buffer", "id", modifier_id)


        if data:

            return render_template('edit_buffer.html', data=data, act=act)

        else:
```

```
                        return 'Error loading #%s' % modifier_id



@app.route("/course")

@login_required

def course():

        data = fetch_all(mysql, "course")

        return render_template('course.html', data=data, table_count=len(data))



@app.route('/edit_course/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required

def edit_course(modifier_id, act):

        if act == "add":

                return render_template('edit_course.html', data="", act="add")

        else:

                data = fetch_one(mysql, "course", "id", modifier_id)


                if data:

                        return render_template('edit_course.html', data=data, act=act)

                else:

                        return 'Error loading #%s' % modifier_id
```

```
@app.route("/room")
@login_required
def room():
    data = fetch_all(mysql, "room")
    return render_template('room.html', data=data, table_count=len(data))



@app.route('/edit_room/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])
@login_required
def edit_room(modifier_id, act):
    if act == "add":
        return render_template('edit_room.html', data="", act="add")
    else:
        data = fetch_one(mysql, "room", "id", modifier_id)

        if data:
            return render_template('edit_room.html', data=data, act=act)
        else:
            return 'Error loading #%s' % modifier_id



@app.route("/semester")
```

```
@login_required

def semester():

    data = fetch_all(mysql, "semester")

    return render_template('semester.html', data=data, table_count=len(data))




@app.route('/edit_semester/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required

def edit_semester(modifier_id, act):

    if act == "add":

        return render_template('edit_semester.html', data="", act="add")

    else:

        data = fetch_one(mysql, "semester", "id", modifier_id)



        if data:

            return render_template('edit_semester.html', data=data, act=act)

        else:

            return 'Error loading #%s' % modifier_id




@app.route("/student")

@login_required

def student():
        data = fetch_all(mysql, "student")

        return render_template('student.html', data=data, table_count=len(data))




@app.route('/edit_student/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required

def edit_student(modifier_id, act):

        if act == "add":

                return render_template('edit_student.html', data="", act="add")

        else:

                data = fetch_one(mysql, "student", "id", modifier_id)



                if data:

                        return render_template('edit_student.html', data=data, act=act)

                else:

                        return 'Error loading #%s' % modifier_id




@app.route("/subject")

@login_required

def subject():

        data = fetch_all(mysql, "subject")

        return render_template('subject.html', data=data, table_count=len(data))
```

```
@app.route('/edit_subject/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])
@login_required
def edit_subject(modifier_id, act):
    if act == "add":
        return render_template('edit_subject.html', data="", act="add")
    else:
        data = fetch_one(mysql, "subject", "id", modifier_id)

        if data:
            return render_template('edit_subject.html', data=data, act=act)
        else:
            return 'Error loading #%s' % modifier_id



@app.route("/teacher")
@login_required
def teacher():
    data = fetch_all(mysql, "teacher")
    return render_template('teacher.html', data=data, table_count=len(data))
```

```
@app.route('/edit_teacher/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required

def edit_teacher(modifier_id, act):

    if act == "add":

        return render_template('edit_teacher.html', data="", act="add")

    else:

        data = fetch_one(mysql, "teacher", "id", modifier_id)



        if data:

            return render_template('edit_teacher.html', data=data, act=act)

        else:

            return 'Error loading #%s' % modifier_id




@app.route("/users")

@login_required

def users():

    data = fetch_all(mysql, "users")

    return render_template('users.html', data=data, table_count=len(data))




@app.route('/edit_users/<string:act>/<int:modifier_id>', methods=['GET', 'POST'])

@login_required
```

```
def edit_users(modifier_id, act):
    if act == "add":
        return render_template('edit_users.html', data="", act="add")
    else:
        data = fetch_one(mysql, "users", "id", modifier_id)

        if data:
            return render_template('edit_users.html', data=data, act=act)
        else:
            return 'Error loading #%s' % modifier_id



@app.route('/save', methods=['GET', 'POST'])
@login_required
def save():
    cat = ''
    if request.method == 'POST':
        post_data = request.form.to_dict()
        if 'password' in post_data:
            post_data['password'] = generate_password_hash(post_data['password'])
        if post_data['act'] == 'add':
            cat = post_data['cat']
            insert_one(mysql, cat, post_data)

        elif post_data['act'] == 'edit':

            cat = post_data['cat']

            update_one(mysql, cat, post_data, post_data['modifier'], post_data['id'])

    else:

        if request.args['act'] == 'delete':

            cat = request.args['cat']

            delete_one(mysql, cat, request.args['modifier'], request.args['id'])

    return redirect("./" + cat)




@app.route('/login')

def login():

    if 'authorised' in session:

        return redirect(url_for('index'))

    else:

        error = request.args['error'] if 'error' in request.args else ''

        return render_template('login.html', error=error)




@app.route('/login_handler', methods=['POST'])

def login_handler():

    try:
        email = request.form['email']

        password = request.form['password']

        data = fetch_one(mysql, "users", "email", email)



        if data and len(data) > 0:

            if check_password_hash(data[3], password) or
hashlib.md5(password.encode('utf-8')).hexdigest() == data[3]:

                session['authorised'] = 'authorised',

                session['id'] = data[0]

                session['name'] = data[1]

                session['email'] = data[2]

                session['role'] = data[4]

                return redirect(url_for('index'))

            else:

                return redirect(url_for('login', error='Wrong Email address or
Password.'))

        else:

            return redirect(url_for('login', error='No user'))



    except Exception as e:

        return render_template('login.html', error=str(e))
```

```
@app.route('/logout')

@login_required

def logout():

        session.clear()

        return redirect(url_for('login'))




if __name__ == "__main__":

        app.run(debug=True)

def fetch_all(mysql, table_name):

        cursor = mysql.connect().cursor()

        cursor.execute("SELECT * FROM " + table_name)

        data = cursor.fetchall()

        if data is None:

                return "Problem!"

        else:

                return data




def fetch_one(mysql, table_name, column, value):

        cursor = mysql.connect().cursor()

        cursor.execute("SELECT * FROM " + table_name + " WHERE " + column + " = '" +
str(value) + "'")
```

```
    data = cursor.fetchone()
    if data is None:
        return "Problem!"
    else:
        return data


def count_all(mysql):
    cursor = mysql.connect().cursor()
    cursor.execute("SHOW TABLES")
    tables = cursor.fetchall()
    data = ()
    for (table) in tables:
        data += ((table[0], count_table(mysql, table[0])),)
    return data


def count_table(mysql, table_name):
    cursor = mysql.connect().cursor()
    cursor.execute("SELECT COUNT(*) FROM " + table_name)
    table_count = cursor.fetchone()
    return table_count[0]
```

```
def clean_data(data):

    del data["cat"]

    del data["act"]

    del data["id"]

    del data["modifier"]

    return data



def insert_one(mysql, table_name, data):

    data = clean_data(data)

    columns = ','.join(data.keys())

    values = ','.join([str("'" + e + "'") for e in data.values()])

    insert_command = "INSERT into " + table_name + " (%s) VALUES (%s) " % (columns, values)

    try:

        con = mysql.connect()

        cursor = con.cursor()

        cursor.execute(insert_command)

        con.commit()

        return True

    except Exception as e:

        print("Problem inserting into db: " + str(e))
        return False



def update_one(mysql, table_name, data, modifier, item_id):

    data = clean_data(data)

    update_command = "UPDATE " + table_name + " SET {} WHERE " + modifier + " = " + item_id + " LIMIT 1"

    update_command = update_command.format(", ".join("{}= '{}'".format(k, v) for k, v in data.items()))

    try:

        con = mysql.connect()

        cursor = con.cursor()

        cursor.execute(update_command)

        con.commit()

        return True

    except Exception as e:

        print("Problem updating into db: " + str(e))

        return False



def delete_one(mysql, table_name, modifier, item_id):

    try:

        con = mysql.connect()
```

```
            cursor = con.cursor()

            delete_command = "DELETE FROM " + table_name + " WHERE " + modifier
+ " = " + item_id + " LIMIT 1"

            cursor.execute(delete_command)

            con.commit()

            return True

        except Exception as e:

            print("Problem deleting from db: " + str(e))

            return False

(function() {

  "use strict";



  const select = (el, all = false) => {

    el = el.trim()

    if (all) {

      return [...document.querySelectorAll(el)]

    } else {

      return document.querySelector(el)

    }

  }
```

```
const on = (type, el, listener, all = false) => {

  let selectEl = select(el, all)



  if (selectEl) {

    if (all) {

      selectEl.forEach(e => e.addEventListener(type, listener))

    } else {

      selectEl.addEventListener(type, listener)

    }

  }

}





const scrollto = (el) => {

  window.scrollTo({

    top: 0,

    behavior: 'smooth'

  })

}



on('click', '.mobile-nav-toggle', function(e) {

  select('#navbar').classList.toggle('navbar-mobile')

  this.classList.toggle('bi-list')
```

```
      this.classList.toggle('bi-x')
    })



    on('click', '#navbar .nav-link', function(e) {
      let section = select(this.hash)
      if (section) {
        e.preventDefault()

        let navbar = select('#navbar')
        let header = select('#header')
        let sections = select('section', true)
        let navlinks = select('#navbar .nav-link', true)

        navlinks.forEach((item) => {
          item.classList.remove('active')
        })

        this.classList.add('active')

        if (navbar.classList.contains('navbar-mobile')) {
          navbar.classList.remove('navbar-mobile')
          let navbarToggle = select('.mobile-nav-toggle')
          navbarToggle.classList.toggle('bi-list')
          navbarToggle.classList.toggle('bi-x')
        }

        if (this.hash == '#header') {
          header.classList.remove('header-top')
          sections.forEach((item) => {
            item.classList.remove('section-show')
          })
          return;
        }

        if (!header.classList.contains('header-top')) {
          header.classList.add('header-top')
          setTimeout(function() {
            sections.forEach((item) => {
              item.classList.remove('section-show')
            })
            section.classList.add('section-show')

          }, 350);
        } else {
          sections.forEach((item) => {
              item.classList.remove('section-show')

          })

          section.classList.add('section-show')

        }



        scrollto(this.hash)

      }

   }, true)




   window.addEventListener('load', () => {

      if (window.location.hash) {

         let initial_nav = select(window.location.hash)



         if (initial_nav) {

            let header = select('#header')

            let navlinks = select('#navbar .nav-link', true)



            header.classList.add('header-top')



            navlinks.forEach((item) => {

               if (item.getAttribute('href') == window.location.hash) {

                  item.classList.add('active')
        } else {
          item.classList.remove('active')
        }
      })

      setTimeout(function() {
        initial_nav.classList.add('section-show')
      }, 350);

      scrollto(window.location.hash)
    }
  }
});


let skilsContent = select('.skills-content');
if (skilsContent) {
  new Waypoint({
    element: skilsContent,
    offset: '80%',
    handler: function(direction) {
      let progress = select('.progress .progress-bar', true);
      progress.forEach((el) => {
```

```
                el.style.width = el.getAttribute('aria-valuenow') + '%'
            });
        }
    })
}


new Swiper('.testimonials-slider', {
    speed: 600,
    loop: true,
    autoplay: {
        delay: 5000,
        disableOnInteraction: false
    },
    slidesPerView: 'auto',
    pagination: {
        el: '.swiper-pagination',
        type: 'bullets',
        clickable: true
    },
    breakpoints: {
        320: {
            slidesPerView: 1,
          spaceBetween: 20

        },



        1200: {

          slidesPerView: 3,

          spaceBetween: 20

        }

    }

  });




  window.addEventListener('load', () => {

    let portfolioContainer = select('.portfolio-container');

    if (portfolioContainer) {

        let portfolioIsotope = new Isotope(portfolioContainer, {

          itemSelector: '.portfolio-item',

          layoutMode: 'fitRows'

        });



        let portfolioFilters = select('#portfolio-flters li', true);



        on('click', '#portfolio-flters li', function(e) {

          e.preventDefault();
```

```
            portfolioFilters.forEach(function(el) {

              el.classList.remove('filter-active');

            });

            this.classList.add('filter-active');



            portfolioIsotope.arrange({

              filter: this.getAttribute('data-filter')

            });

          }, true);

      }



  });




  const portfolioLightbox = GLightbox({

      selector: '.portfolio-lightbox'

  });




  const portfolioDetailsLightbox = GLightbox({

      selector: '.portfolio-details-lightbox',

      width: '90%',

      height: '90vh'
```

```
   });



   new Swiper('.portfolio-details-slider', {

      speed: 400,

      loop: true,

      autoplay: {

         delay: 5000,

         disableOnInteraction: false

      },

      pagination: {

         el: '.swiper-pagination',

         type: 'bullets',

         clickable: true

      }

   });


})()


<!DOCTYPE html>

         <html lang="en">

         <head>

                  <meta charset="utf-8">
```

```
        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <meta name="author" content="@housamz">

        <meta name="description" content="Mage">

        <title>Sign In</title>



        <!-- Latest compiled and minified CSS -->

        <!-- <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
integrity="sha384-
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
crossorigin="anonymous"> -->

        <link
href="https://maxcdn.bootstrapcdn.com/bootswatch/3.3.7/cosmo/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
h21C2fcDk/eFsW9sC9h0dhokq5pDinLNklTKoxIZRUn3+hvmgQSffLLQ4G4l2eEr"
crossorigin="anonymous">



        <!-- Custom CSS -->

        <link rel="stylesheet" href="/static/style.css">

        <link href="//cdn.datatables.net/1.10.16/css/dataTables.bootstrap.min.css"
rel="stylesheet" type="text/css" />

        <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->

        <!-- WARNING: Respond.js doesnt work if you view the page via file:// -->

        <!--[if lt IE 9]>
```

```
            <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

            <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

        <![endif]-->

        <style type="text/css">

            body { padding-top: 40px;

            background: url('../static/bg.jpg');}

            h3, h4 {text-align: center;

            color: #fff;

            }

            form { max-width: 330px; margin: 0 auto}

            input, h3, h4 { margin-bottom: 10px; padding: 10px}

        </style>


    </head>

    <body>

        <div class="container">

            <form action="/login_handler" method="post">

                <h3>Welcome to Square Admin Panel</h3>

                <h4>Sign in</h4>


                {% if error is defined and error != " %}
                <div class="alert alert-danger" role="alert">
                    <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
                    <span class="sr-only">Error:</span>
                    {{error}}
                </div>
                {% endif %}


                <label for="inputEmail" class="sr-only">Email address</label>
                <input type="text" name="email" class="form-control" placeholder="Email address" required autofocus>


                <label for="inputPassword" class="sr-only">Password</label>
                <input type="password" name="password" class="form-control" placeholder="Password" required>


                <button class="btn btn-lg btn-primary btn-block" style="background-color: #18d26e;" type="submit">Sign in</button>
            </form>
        </div> <!-- /container -->
    </body>
</html>


{% extends "base.html" %}
{% block content %}
	<table class="table table-striped">
		<tr>
			<th class="not">Table</th>
			<th class="not">Entries</th>
		</tr>

		{% for table_name, table_count in tables_and_counts %}
		<tr>
			<td><a href="./{{table_name}}">{{table_name|capitalize}}</a></td>
			<td>{{ table_count }}</td>
		</tr>
		{% endfor %}

</table>
{% endblock %}



<!-- Sidebar Holder -->
<nav id="sidebar" class="bg-primary">
	<div class="sidebar-header">
		<h3>
```

```
            Square Admin<br>
            <i id="sidebarCollapse" class="glyphicon glyphicon-circle-arrow-left"></i>
        </h3>
        <strong>
            Square<br>
            <i id="sidebarExtend" class="glyphicon glyphicon-circle-arrow-right"></i>
        </strong>

    </div><!-- /sidebar-header -->
    <!-- start sidebar -->
    <ul class="list-unstyled components">
        <li>
            <a href="./" aria-expanded="false">
                <i class="glyphicon glyphicon-home"></i>
                Home
            </a>
        </li>
<li>
    <a href="/attendence">
        <i class="glyphicon glyphicon-euro"></i>
        Attendence
        <span class="pull-right">{{tables_and_counts.0.1}}</span>
    </a>
</li>


<li>
    <a href="/buffer">
        <i class="glyphicon glyphicon-superscript"></i>
        Buffer
        <span class="pull-right">{{tables_and_counts.1.1}}</span>
    </a>
</li>


<li>
    <a href="/course">
        <i class="glyphicon glyphicon-ice-lolly"></i>
        Course
        <span class="pull-right">{{tables_and_counts.2.1}}</span>
    </a>
</li>


<li>
    <a href="/room">
        <i class="glyphicon glyphicon-th-list"></i>
```

```
                Room
                <span class="pull-right">{{tables_and_counts.3.1}}</span>
        </a>
</li>

<li>
        <a href="/semester">
                <i class="glyphicon glyphicon-hand-right"></i>
                Semester
                <span class="pull-right">{{tables_and_counts.4.1}}</span>
        </a>
</li>

<li>
        <a href="/student">
                <i class="glyphicon glyphicon-lamp"></i>
                Student
                <span class="pull-right">{{tables_and_counts.5.1}}</span>
        </a>
</li>

<li>
        <a href="/subject">
```

```
                <i class="glyphicon glyphicon-forward"></i>

                Subject

                <span class="pull-right">{{tables_and_counts.6.1}}</span>

        </a>

</li>



<li>

        <a href="/teacher">

                <i class="glyphicon glyphicon-tree-deciduous"></i>

                Teacher

                <span class="pull-right">{{tables_and_counts.7.1}}</span>

        </a>

</li>



<li>

        <a href="/users">

                <i class="glyphicon glyphicon-user"></i>

                Users

                <span class="pull-right">{{tables_and_counts.8.1}}</span>

        </a>

</li>
```

```
        <li><a href="/logout"><i class="glyphicon glyphicon-log-out"></i>
Logout</a></li>

    </ul>



</nav><!-- / end sidebar -->

<!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <meta name="author" content="@housamz">

        <meta name="description" content="Mage">

        <title>Admin Panel</title>



        <!-- Latest compiled and minified CSS -->

        <!-- <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
integrity="sha384-
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
crossorigin="anonymous"> -->

        <link
href="https://maxcdn.bootstrapcdn.com/bootswatch/3.3.7/cosmo/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
h21C2fcDk/eFsW9sC9h0dhokq5pDinLNklTKoxIZRUn3+hvmgQSffLLQ4G4l2eEr"
crossorigin="anonymous">
```

```
        <!-- Custom CSS -->
        <link rel="stylesheet" href="/static/style.css">
        <link href="//cdn.datatables.net/1.10.16/css/dataTables.bootstrap.min.css" rel="stylesheet" type="text/css" />
        <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
        <!-- WARNING: Respond.js doesnt work if you view the page via file:// -->
        <!--[if lt IE 9]>
            <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
            <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
        <![endif]-->


    </head>
    <body>
        <div class="wrapper">
    {% include "sidebar.html" %}


    <!-- Page Content Holder -->
            <div id="content">
              {% block content %}{% endblock %}
    </div>
```

```
                </div>

                <!-- jQuery Version 1.11.1 -->

                <script src="https://code.jquery.com/jquery-1.12.0.min.js"></script>

                <!-- Bootstrap Core JavaScript -->

                <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js" integrity="sha384-
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9aJ7xS"
crossorigin="anonymous"></script>

                <script                                                    type="text/javascript"
src="//cdn.ckeditor.com/4.4.3/standard/ckeditor.js"></script>

                <script                                                    type="text/javascript"
src="//cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>

                <script                                                    type="text/javascript"
src="//cdn.datatables.net/1.10.16/js/dataTables.bootstrap.min.js"></script>

                <script type="text/javascript">

                        $(document).ready(function () {

                                $("#sidebarCollapse, #sidebarExtend").on("click", function () {

                                        $("#sidebar").toggleClass("active");

                                });

                                $("#sorted").DataTable( {

                                        "bStateSave": true,

                                        "sPaginationType": "full_numbers"

                                });

                        });
```

```
        </script>

        <script type="text/javascript">

        function navConfirm(loc) {

            if (confirm("Are you sure?")) {

                window.location.href = loc;

            }

            return false;

        }

        </script>



    </body>

</html>




{% extends "base.html" %}



{% block content %}

    <a class="btn btn-primary" href="./edit_student/add/0">

        <i class="glyphicon glyphicon-plus-sign"></i> Add New Student

    </a>

    <h1>Student</h1>

    <p>This table includes {{table_count}} student</p>

    <table id="sorted" class="table table-striped table-bordered">
```

```
<thead>

        <tr>

                <th>Id</th>

                <th>Name</th>

                <th>Email</th>

                <th>Phone Number</th>

                <th>Course</th>

                <th>Semester</th>

                <th>Student Mac</th>

                <th>Password</th>



                <th>Edit</th>

                <th>Delete</th>

        </tr>

</thead>

<tbody>

        {% for row_data in data %}

        <tr>

                {% for d in row_data %}

                        <td>{{d}}</td>

                {% endfor %}

                <td>

                        <a href="./edit_student/edit/{{row_data[0]}}">
```

```
                                        <i class="glyphicon glyphicon-edit"></i>

                                   </a>

                              </td>

                              <td>

                                   <a

      href="./save?cat=student&act=delete&modifier=id&id={{row_data[0]}}"

                                        onclick="return navConfirm(this.href);">

                                   <i class="glyphicon glyphicon-trash"></i>

                                   </a>

                                   </td>

                         </tr>

                         {% endfor %}

                    </tbody>

          </table>

{% endblock %}




{% extends "base.html" %}



{% block content %}

     <form method="post" action="{{ url_for('save') }}" enctype='multipart/form-data'>

          <fieldset>
```

```
<legend class="hidden-first">Add New Student</legend>

<input name="cat" type="hidden" value='student'>

<input name="modifier" type="hidden" value='id'>

<input name="id" type="hidden" value="{{data[0]}}">

<input name="act" type="hidden" value="{{act}}">



<label>Name</label>

<input

        class="form-control"

        type="text"

        name="name"

        value="{{data[1]}}"  required/><br>

<label>Email</label>

<input

        class="form-control"

        type="email"

        name="email"

        value="{{data[2]}}" required /><br>

<label>Phone Number</label>

<input

        class="form-control"

        type="text"

        name="phone_number"
```

```
                value="{{data[3]}}" required /><br>

        <label>Course</label>

        <input

                class="form-control"

                type="text"

                name="course"

                value="{{data[4]}}"  /><br>

        <label>Semester</label>

        <input

                class="form-control"

                type="text"

                name="semester"

                value="{{data[5]}}"  /><br>

        <label>Student Mac</label>

        <input

                class="form-control"

                type="text"

                name="student_mac"

                value="{{data[6]}}" required /><br>

        <label>Password</label>

        <input

                class="form-control"

                type="text"
                name="password"

                value="{{data[7]}}" /><br><br>

            <input type="submit" value=" Save " class="btn btn-success">

        </fieldset>

    </form>

{% endblock %}




{% extends "base.html" %}


{% block content %}

    <form method="post" action="{{ url_for('save') }}" enctype='multipart/form-data'>

        <fieldset>

            <legend class="hidden-first">Add New Teacher</legend>

            <input name="cat" type="hidden" value='teacher'>

            <input name="modifier" type="hidden" value='id'>

            <input name="id" type="hidden" value="{{data[0]}}">

            <input name="act" type="hidden" value="{{act}}">


            <label>Name</label>

            <input

                class="form-control"

                type="text"
```

```
          name="name"

          value="{{data[1]}}" required/><br>

<label>Email Id</label>

<input

          class="form-control"

          type="email"

          name="email_id"

          value="{{data[2]}}" required /><br>

<label>Phone No</label>

<input

          class="form-control"

          type="text"

          name="phone_no"

          value="{{data[3]}}"required /><br>

<label>Qualification</label>

<input

          class="form-control"

          type="text"

          name="qualification"

          value="{{data[4]}}" /><br>

<label>Course</label>

<input

          class="form-control"
```

```
                        type="text"

                        name="course"

                        value="{{data[5]}}" /><br>

                    <label>Subject</label>

                    <input

                        class="form-control"

                        type="text"

                        name="subject"

                        value="{{data[6]}}" required /><br>

                    <label>Password</label>

                    <input

                        class="form-control"

                        type="text"

                        name="password"

                        value="{{data[7]}}" /><br><br>

                    <input type="submit" value=" Save " class="btn btn-success">

                </fieldset>

            </form>

{% endblock %}




{% extends "base.html" %}
{% block content %}

        <a class="btn btn-primary" href="./edit_attendence/add/0">

                <i class="glyphicon glyphicon-plus-sign"></i> Add New Attendence

        </a>

        <h1>Attendence</h1>

        <p>This table includes {{table_count}} attendence</p>

        <table id="sorted" class="table table-striped table-bordered">

                <thead>

                        <tr>

                                <th>Id</th>

                                <th>Room No</th>

                                <th>Teacher Name</th>

                                <th>Student Id</th>

                                <th>Subject Id</th>

                                <th>Teacher Total</th>

                                <th>Student Total</th>

                                <th>Student Average</th>



                                <th>Edit</th>

                                <th>Delete</th>

                        </tr>

                </thead>

                <tbody>
```

```
                        {% for row_data in data %}

                        <tr>

                                {% for d in row_data %}

                                        <td>{{d}}</td>

                                {% endfor %}

                                <td>

                                        <a href="./edit_attendence/edit/{{row_data[0]}}">

                                                <i class="glyphicon glyphicon-edit"></i>

                                        </a>

                                </td>

                                <td>

                                        <a


        href="./save?cat=attendence&act=delete&modifier=id&id={{row_data[0]}}"

                                                onclick="return navConfirm(this.href);">

                                        <i class="glyphicon glyphicon-trash"></i>

                                        </a>

                                        </td>

                        </tr>

                        {% endfor %}

                </tbody>

        </table>

{% endblock %}
{% extends "base.html" %}


{% block content %}
	<form method="post" action="{{ url_for('save') }}" enctype='multipart/form-data'>
		<fieldset>
			<legend class="hidden-first">Add New Attendence</legend>
			<input name="cat" type="hidden" value='attendence'>
			<input name="modifier" type="hidden" value='id'>
			<input name="id" type="hidden" value="{{data[0]}}">
			<input name="act" type="hidden" value="{{act}}">


			<label>Room No</label>
			<input
				class="form-control"
				type="text"
				name="room_no"
				value="{{data[1]}}" /><br>
			<label>Teacher Id</label>
			<input
				class="form-control"
				type="text"
```

```
          name="teacher_id"

          value="{{data[2]}}" /><br>

<label>Student Id</label>

<input

          class="form-control"

          type="text"

          name="student_id"

          value="{{data[3]}}" /><br>

<label>Subject Id</label>

<input

          class="form-control"

          type="text"

          name="subject_id"

          value="{{data[4]}}" /><br>

<label>Teacher Total</label>

<input

          class="form-control"

          type="text"

          name="teacher_total"

          value="{{data[5]}}" /><br>

<label>Student Total</label>

<input

          class="form-control"
```

```
                type="text"
                name="student_total"
                value="{{data[6]}}" /><br>
            <label>Student Average</label>
            <input
                class="form-control"
                type="text"
                name="student_average"
                value="{{data[7]}}" /><br><br>
            <input type="submit" value=" Save " class="btn btn-success">
        </fieldset>
    </form>
{% endblock %}


body {
    font-family: "Open Sans", sans-serif;
    background-color: #040404;
    color: #fff;
    position: relative;
    background: transparent;
}
body::before {
    content: "";
    position: fixed;

    background: #000000 url("../img/bg.jpg") top right no-repeat;

    background-size: cover;

    left: 0;

    right: 0;

    top: 0;

    height: 100vh;

    z-index: -1;

}



@media (min-width: 1024px) {

    body::before {

        background-attachment: fixed;

    }

}



a {

    color: #18d26e;

    text-decoration: none;

}



a:hover {

    color: #35e888;
    text-decoration: none;

}



h1,

h2,

h3,

h4,

h5,

h6 {

    font-family: "Raleway", sans-serif;

}

.header {

    transition: ease-in-out 0.3s;

    position: relative;

    height: 100vh;

    display: flex;

    align-items: center;

    z-index: 997;

    overflow-y: auto;

}



#header * {

    transition: ease-in-out 0.3s;
```

```
}

#header h1 {
    font-size: 48px;
    margin: 0;
    padding: 0;
    line-height: 1;
    font-weight: 700;
    font-family: "Poppins", sans-serif;
}

#header h1 a,
#header h1 a:hover {
    color: #fff;
    line-height: 1;
    display: inline-block;
}

#header h2 {
    font-size: 24px;
    margin-top: 20px;
    color: rgba(255, 255, 255, 0.8);
}
```

```
#header h2 span {
    color: #fff;
    border-bottom: 2px solid #18d26e;
    padding-bottom: 6px;
}


#header img {
    padding: 0;
    margin: 0;
}


#header .social-links {
    margin-top: 40px;
    display: flex;
}


#header .social-links a {
    font-size: 16px;
    display: flex;
    justify-content: center;
    align-items: center;
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
    line-height: 1;
    margin-right: 8px;
    border-radius: 50%;
    width: 40px;
    height: 40px;
}

#header .social-links a:hover {
    background: #18d26e;
}

@media (max-width: 992px) {
    #header h1 {
        font-size: 36px;
    }
    #header h2 {
        font-size: 20px;
        line-height: 30px;
    }
    #header .social-links {
        margin-top: 15px;
    }
    #header .container {
        display: flex;
        flex-direction: column;
        align-items: center;
    }
}
#header.header-top {
    height: 80px;
    position: fixed;
    left: 0;
    top: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.9);
}

#header.header-top .social-links,
#header.header-top h2 {
    display: none;
}

#header.header-top h1 {
    margin-right: auto;
    font-size: 36px;
```

```
}


#header.header-top .container {

    display: flex;

    align-items: center;

}


#header.header-top .navbar {

    margin: 0;

}


@media (max-width: 768px) {

    #header.header-top {

        height: 60px;

    }

    #header.header-top h1 {

        font-size: 26px;

    }

}

.navbar {

    padding: 0;

    margin-top: 35px;

}
```

```
.navbar ul {
    margin: 0;
    padding: 0;
    display: flex;
    list-style: none;
    align-items: center;
}

.navbar li {
    position: relative;
}

.navbar li+li {
    margin-left: 30px;
}

.navbar a,
.navbar a:focus {
    display: flex;
    align-items: center;
    justify-content: space-between;
    padding: 0;
    font-family: "Poppins", sans-serif;

    font-size: 16px;

    font-weight: 400;

    color: rgba(255, 255, 255, 0.7);

    white-space: nowrap;

    transition: 0.3s;

}



.navbar a i,

.navbar a:focus i {

    font-size: 12px;

    line-height: 0;

    margin-left: 5px;

}



.navbar a:before {

    content: "";

    position: absolute;

    width: 0;

    height: 2px;

    bottom: -4px;

    left: 0;

    background-color: #18d26e;
    visibility: hidden;

    width: 0px;

    transition: all 0.3s ease-in-out 0s;

}



.navbar a:hover:before,

.navbar li:hover>a:before,

.navbar .active:before {

    visibility: visible;

    width: 25px;

}



.navbar a:hover,

.navbar .active,

.navbar .active:focus,

.navbar li:hover>a {

    color: #fff;

}

.mobile-nav-toggle {

    color: #fff;

    font-size: 28px;

    cursor: pointer;

    display: none;
    line-height: 0;

    transition: 0.5s;

    position: fixed;

    right: 15px;

    top: 15px;

}



@media (max-width: 991px) {

    .mobile-nav-toggle {

        display: block;

    }

    .navbar ul {

        display: none;

    }

}



.navbar-mobile {

    position: fixed;

    overflow: hidden;

    top: 0;

    right: 0;

    left: 0;

    bottom: 0;
    background: rgba(0, 0, 0, 0.9);
    transition: 0.3s;
    z-index: 999;
    margin-top: 0;
}


.navbar-mobile .mobile-nav-toggle {
    position: absolute;
    top: 15px;
    right: 15px;
}


.navbar-mobile ul {
    display: block;
    position: absolute;
    top: 55px;
    right: 15px;
    bottom: 45px;
    left: 15px;
    padding: 10px 0;
    overflow-y: auto;
    transition: 0.3s;
    border: 2px solid rgba(255, 255, 255, 0.2);
```

```
}



.navbar-mobile li {

    padding: 12px 20px;

}



.navbar-mobile li+li {

    margin: 0;

}



.navbar-mobile a,

.navbar-mobile a:focus {

    font-size: 16px;

    position: relative;

}

section {

    overflow: hidden;

    position: absolute;

    width: 100%;

    top: 140px;

    bottom: 100%;

    opacity: 0;

    transition: ease-in-out 0.4s;
    z-index: 2;
}

section.section-show {
    top: 100px;
    bottom: auto;
    opacity: 1;
    padding-bottom: 45px;
}

section .container {
    background: rgba(0, 0, 0, 0.9);
    padding: 30px;
}

@media (max-width: 768px) {
    section {
        top: 120px;
    }
    section.section-show {
        top: 80px;
    }
}
```

```
.section-title h2 {

    font-size: 14px;

    font-weight: 500;

    padding: 0;

    line-height: 1px;

    margin: 0 0 20px 0;

    letter-spacing: 2px;

    text-transform: uppercase;

    color: #aaaaaa;

    font-family: "Poppins", sans-serif;

}


.section-title h2::after {

    content: "";

    width: 120px;

    height: 1px;

    display: inline-block;

    background: #4ceb95;

    margin: 4px 10px;

}


.section-title p {
    margin: 0;

    margin: -15px 0 15px 0;

    font-size: 36px;

    font-weight: 700;

    text-transform: uppercase;

    font-family: "Poppins", sans-serif;

    color: #fff;

}

.about-me .content h3 {

    font-weight: 700;

    font-size: 26px;

    color: #18d26e;

}



.about-me .content ul {

    list-style: none;

    padding: 0;

}



.about-me .content ul li {

    margin-bottom: 20px;

    display: flex;

    align-items: center;
```

```
}

.about-me .content ul strong {
    margin-right: 10px;
}

.about-me .content ul i {
    font-size: 16px;
    margin-right: 5px;
    color: #18d26e;
    line-height: 0;
}

.about-me .content p:last-child {
    margin-bottom: 0;
}
.counts {
    padding: 70px 0 60px;
}

.counts .count-box {
    padding: 30px 30px 25px 30px;
    width: 100%;
    position: relative;
    text-align: center;
    background: rgba(255, 255, 255, 0.08);
}

.counts .count-box i {
    position: absolute;
    top: -25px;
    left: 50%;
    transform: translateX(-50%);
    font-size: 24px;
    background: rgba(255, 255, 255, 0.1);
    padding: 12px;
    color: #18d26e;
    border-radius: 50px;
    line-height: 0;
}

.counts .count-box span {
    font-size: 36px;
    display: block;
    font-weight: 600;
    color: #fff;
```

```
}



.counts .count-box p {

    padding: 0;

    margin: 0;

    font-family: "Raleway", sans-serif;

    font-size: 14px;

}

.skills .progress {

    height: 60px;

    display: block;

    background: none;

    border-radius: 0;

}



.skills .progress .skill {

    padding: 10px 0;

    margin: 0;

    text-transform: uppercase;

    display: block;

    font-weight: 600;

    font-family: "Poppins", sans-serif;

    color: #fff;
```

```
}


.skills .progress .skill .val {
    float: right;
    font-style: normal;
}


.skills .progress-bar-wrap {
    background: rgba(255, 255, 255, 0.2);
}


.skills .progress-bar {
    width: 1px;
    height: 10px;
    transition: 0.9s;
    background-color: #18d26e;
}
.interests .icon-box {
    display: flex;
    align-items: center;
    padding: 20px;
    background: rgba(255, 255, 255, 0.08);
    transition: ease-in-out 0.3s;
```

```
}


.interests .icon-box i {

    font-size: 32px;

    padding-right: 10px;

    line-height: 1;

}


.interests .icon-box h3 {

    font-weight: 700;

    margin: 0;

    padding: 0;

    line-height: 1;

    font-size: 16px;

    color: #fff;

}


.interests .icon-box:hover {

    background: rgba(255, 255, 255, 0.12);

}

.testimonials .testimonial-item {

    box-sizing: content-box;

    min-height: 320px;
```

```
}


.testimonials .testimonial-item .testimonial-img {

    width: 90px;

    border-radius: 50%;

    margin: -40px 0 0 40px;

    position: relative;

    z-index: 2;

    border: 6px solid rgba(255, 255, 255, 0.12);

}


.testimonials .testimonial-item h3 {

    font-size: 18px;

    font-weight: bold;

    margin: 10px 0 5px 45px;

    color: #fff;

}


.testimonials .testimonial-item h4 {

    font-size: 14px;

    color: #999;

    margin: 0 0 0 45px;

}
```

```
.testimonials .testimonial-item .quote-icon-left,
.testimonials .testimonial-item .quote-icon-right {
    color: rgba(255, 255, 255, 0.25);
    font-size: 26px;
}

.testimonials .testimonial-item .quote-icon-left {
    display: inline-block;
    left: -5px;
    position: relative;
}

.testimonials .testimonial-item .quote-icon-right {
    display: inline-block;
    right: -5px;
    position: relative;
    top: 10px;
}

.testimonials .testimonial-item p {
    font-style: italic;
    margin: 0 15px 0 15px;
    padding: 20px 20px 60px 20px;

    background: rgba(255, 255, 255, 0.1);

    position: relative;

    border-radius: 6px;

    position: relative;

    z-index: 1;

}



.testimonials .swiper-pagination {

    margin-top: 20px;

    position: relative;

}



.testimonials .swiper-pagination .swiper-pagination-bullet {

    width: 12px;

    height: 12px;

    opacity: 1;

    background-color: rgba(255, 255, 255, 0.3);

}



.testimonials .swiper-pagination .swiper-pagination-bullet-active {

    background-color: #18d26e;

}
```

```
.portfolio .portfolio-wrap .portfolio-info::before {
    display: block;
    content: "";
    width: 48px;
    height: 48px;
    position: absolute;
    top: 35px;
    left: 35px;
    border-top: 3px solid #fff;
    border-left: 3px solid #fff;
    transition: all 0.5s ease 0s;
    z-index: 9994;
}


.portfolio .portfolio-wrap .portfolio-info::after {
    display: block;
    content: "";
    width: 48px;
    height: 48px;
    position: absolute;
    bottom: 35px;
    right: 35px;
    border-bottom: 3px solid #fff;
    border-right: 3px solid #fff;
    transition: all 0.5s ease 0s;
    z-index: 9994;
}

.portfolio .portfolio-wrap .portfolio-info h4 {
    font-size: 20px;
    color: #fff;
    font-weight: 600;
}

.portfolio .portfolio-wrap .portfolio-info p {
    color: #ffffff;
    font-size: 14px;
    text-transform: uppercase;
    padding: 0;
    margin: 0;
}

.portfolio .portfolio-wrap .portfolio-links {
    text-align: center;
    z-index: 4;
}
```

```
.portfolio .portfolio-wrap .portfolio-links a {
    color: #fff;
    margin: 0 2px;
    font-size: 28px;
    display: inline-block;
    transition: 0.3s;
}

.portfolio .portfolio-wrap .portfolio-links a:hover {
    color: #63eda3;
}

.portfolio .portfolio-wrap:hover::before {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    opacity: 1;
}

.portfolio .portfolio-wrap:hover .portfolio-info {
    opacity: 1;
```

```
}



.portfolio .portfolio-wrap:hover .portfolio-info::before {

   top: 15px;

   left: 15px;




{% extends "base.html" %}



{% block content %}

            <a class="btn btn-primary" href="./edit_users/add/0">

                        <i class="glyphicon glyphicon-plus-sign"></i> Add New Users

            </a>

            <h1>Users</h1>

            <p>This table includes {{table_count}} users</p>

            <table id="sorted" class="table table-striped table-bordered">

                        <thead>

                                    <tr>

                                                <th>Id</th>

                                                <th>Name</th>

                                                <th>Email</th>

                                                <th>Password</th>

                                                <th>Role</th>
```

```
                    <th>Edit</th>

                    <th>Delete</th>

               </tr>

          </thead>

          <tbody>

               {% for row_data in data %}

               <tr>

                    {% for d in row_data %}

                         <td>{{d}}</td>

                    {% endfor %}

                    <td>

                         <a href="./edit_users/edit/{{row_data[0]}}">

                              <i class="glyphicon glyphicon-edit"></i>

                         </a>

                    </td>

                    <td>

                         <a

     href="./save?cat=users&act=delete&modifier=id&id={{row_data[0]}}"

                              onclick="return navConfirm(this.href);">

                         <i class="glyphicon glyphicon-trash"></i>

                         </a>
                    </td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
{% endblock %}

{% extends "base.html" %}

{% block content %}
        <form method="post" action="{{ url_for('save') }}" enctype='multipart/form-data'>
            <fieldset>
                <legend class="hidden-first">Add New Users</legend>
                <input name="cat" type="hidden" value='users'>
                <input name="modifier" type="hidden" value='id'>
                <input name="id" type="hidden" value="{{data[0]}}">
                <input name="act" type="hidden" value="{{act}}">

                <label>Name</label>
                <input
                    class="form-control"
                    type="text"
                    name="name"
                    value="{{data[1]}}" /><br>

                <label>Email</label>

                <input

                    class="form-control"

                    type="text"

                    name="email"

                    value="{{data[2]}}" /><br>

                <label>Password</label>

                <input

                    class="form-control"

                    type="text"

                    name="password"

                    value="{{data[3]}}" /><br>

                <label>Role</label>

                <input

                    class="form-control"

                    type="text"

                    name="role"

                    value="{{data[4]}}" /><br><br>

                <input type="submit" value=" Save " class="btn btn-success">

            </fieldset>

        </form>

{% endblock %}
```

6. TESTING AND RESULTS

6.1 TESTING:

6.1.1 UNIT TESTING:

Unit testing is a method by which individual units of source code is tested and sets of one or more computer program modules together with associated control data, usage, procedures and Operating procedures.

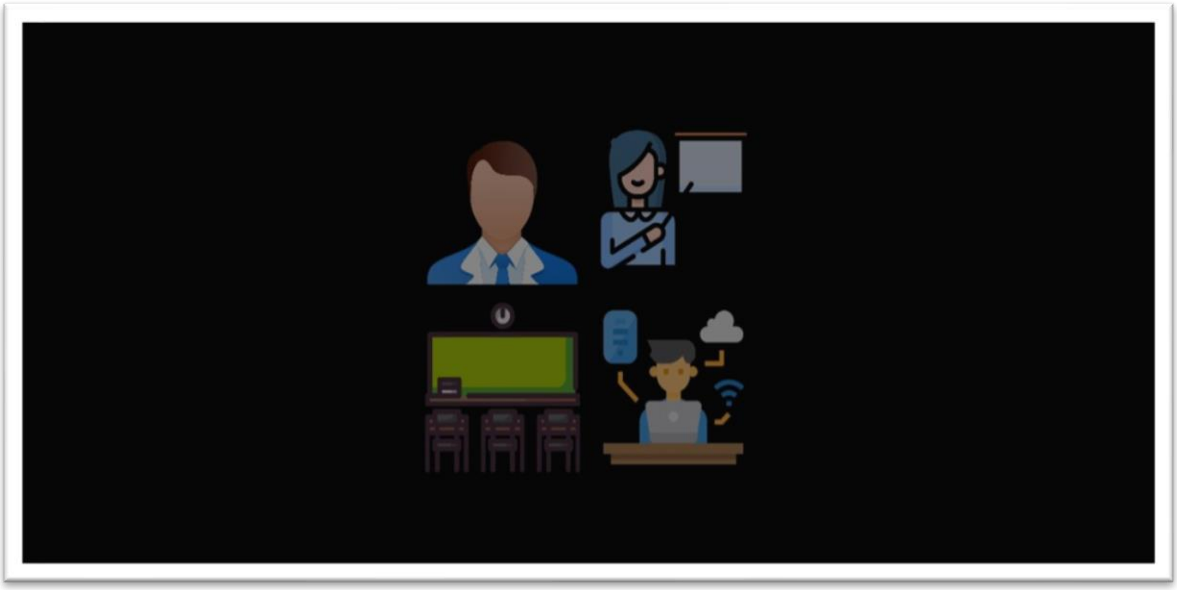
For unit testing we adopted code testing strategy, which examined the logic of program. During the development process itself all the syntax errors get rooted out. For this developed test case the result in executing every instruction in the program verifies the functionalities of the modules under test.

Test id	Test Case	Input	Expected result	Output	Status
1	Input type Gmail, password	Admin	Admin Homepage	Logged in Successfully	Successfully Logged in
2	Input type Gmail, password	Admin	Unsuccessful login	Invalid password or username	Unsuccessful login
3	Input type Gmail, Password, New Device	Student	Student Homepage	Logged in Successfully	Successful login
4	Input type Gmail, Password, New Device	Student	Unsuccessful login	Invalid password or username	Unsuccessful login
5	Input type Gmail, Password, Registered Device	Student	Student Homepage	Logged in Successfully	Successfully Logged in

6	Input type Gmail, Password, Un-registered Device	Student	Unsuccessful login	Invalid password or username	Unsuccessful login
7	Mark Attendance	Student	Decode QR-Code and update to Buffer	Home page	Attendance Marked successfully
8	Input type Gmail, Password	Teacher	Teacher Homepage	Logged in Successfully	Successful login
9	Input type Gmail, Password	Teacher	Unsuccessful login	Invalid password or username	Unsuccessful login
10	Mark Attendance	Teacher	Decode QR-Code and fetch details from Buffer	Home page	Attendance Marked successfully for Students
11	Input type username, Password	Classroom	Classroom Homepage	Logged in Successfully	Successful login
12	Input type username, Password	Classroom	Unsuccessful login	Invalid password or username	Updated successfully
13	QR-Generator	Classroom	Display QR-code	QR-code page	Successful QR-Code generated

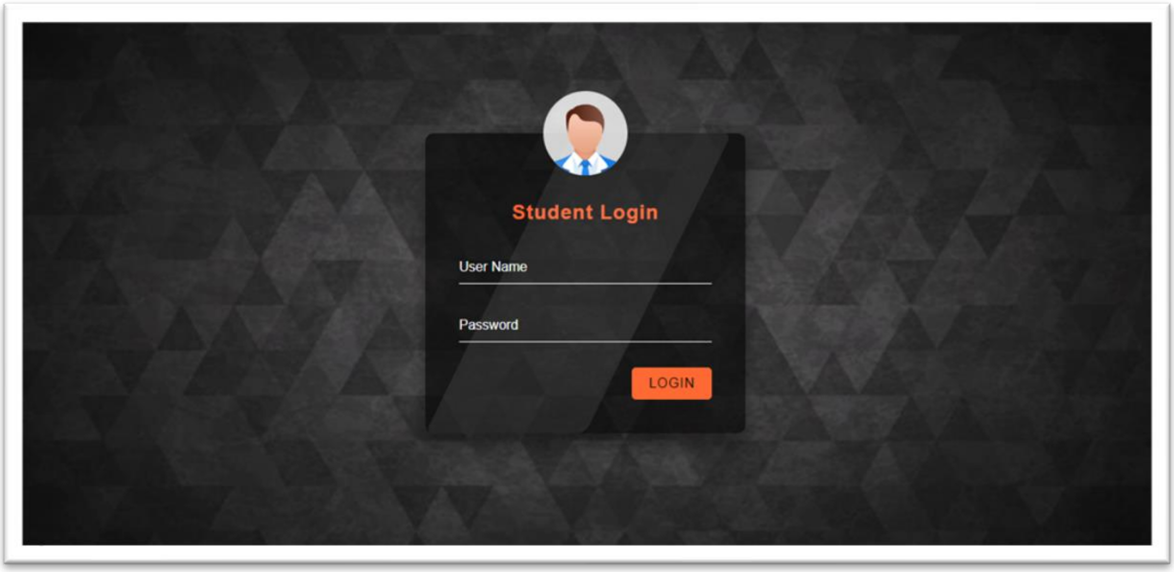
6.2 RESULTS

1. LANDING PAGE:



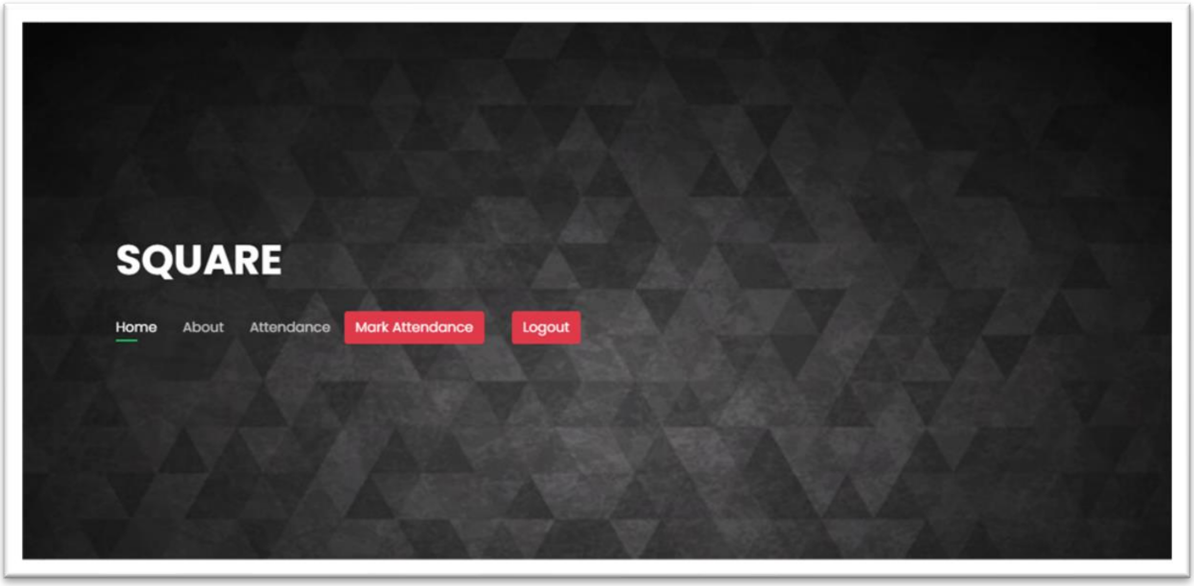
This page lets us select the type of user who would want to login as: i.e, student, teacher, classroom or as admin.

2. STUDENT LOGIN:



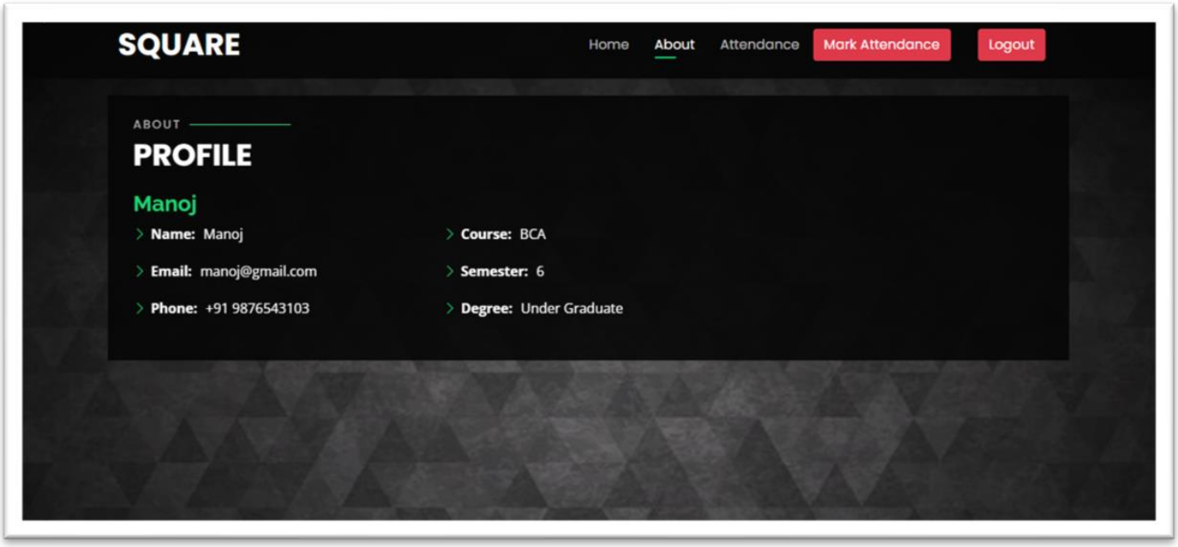
This is the student login page where the student has to type the gmail is and password to log in to the application.

3. STUDENT HOMEPAGE:



This page lets us select the action that the student has to perform.

4. STUDENT PROFILE:



ABOUT

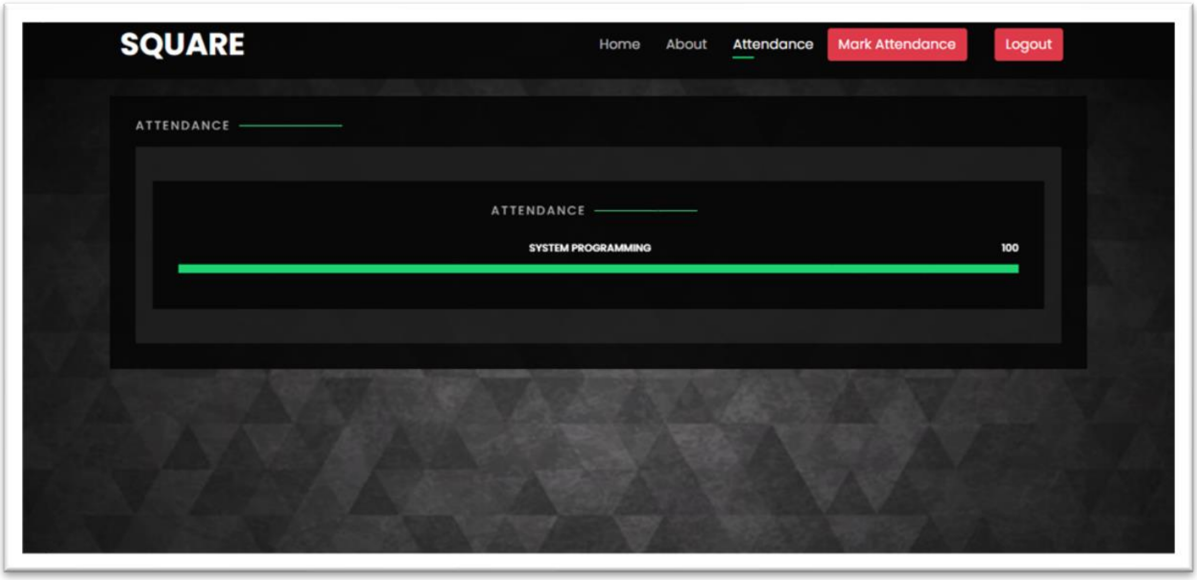
PROFILE

Manoj

- **Name:** Manoj
- **Email:** manoj@gmail.com
- **Phone:** +91 9876543103
- **Course:** BCA
- **Semester:** 6
- **Degree:** Under Graduate

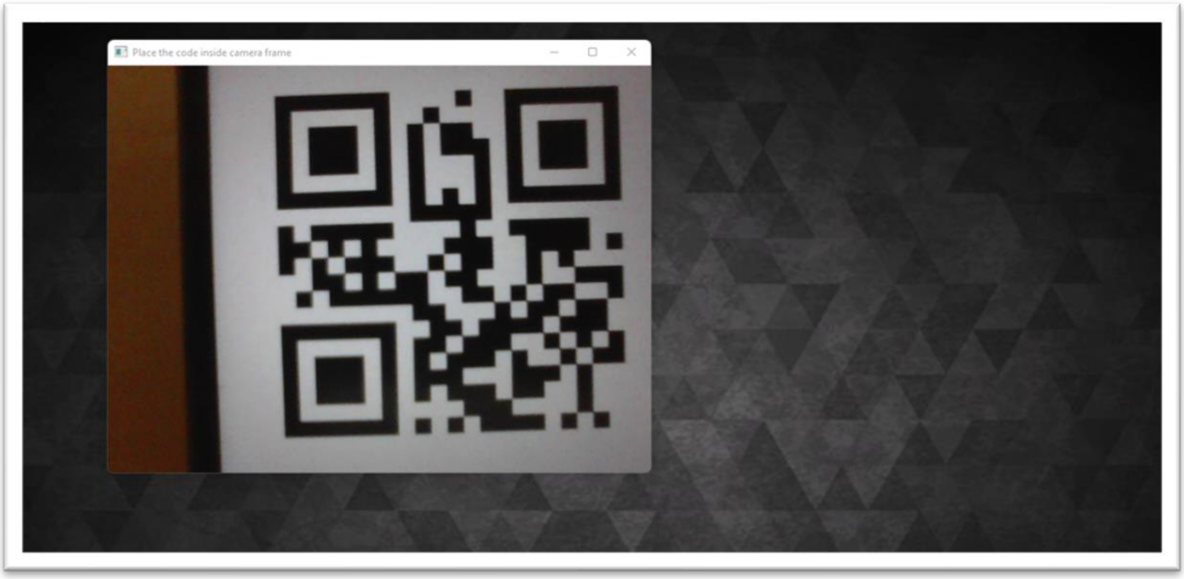
This page lets us know the details of the student.

5. STUDENT ATTENDANCE DETAILS:



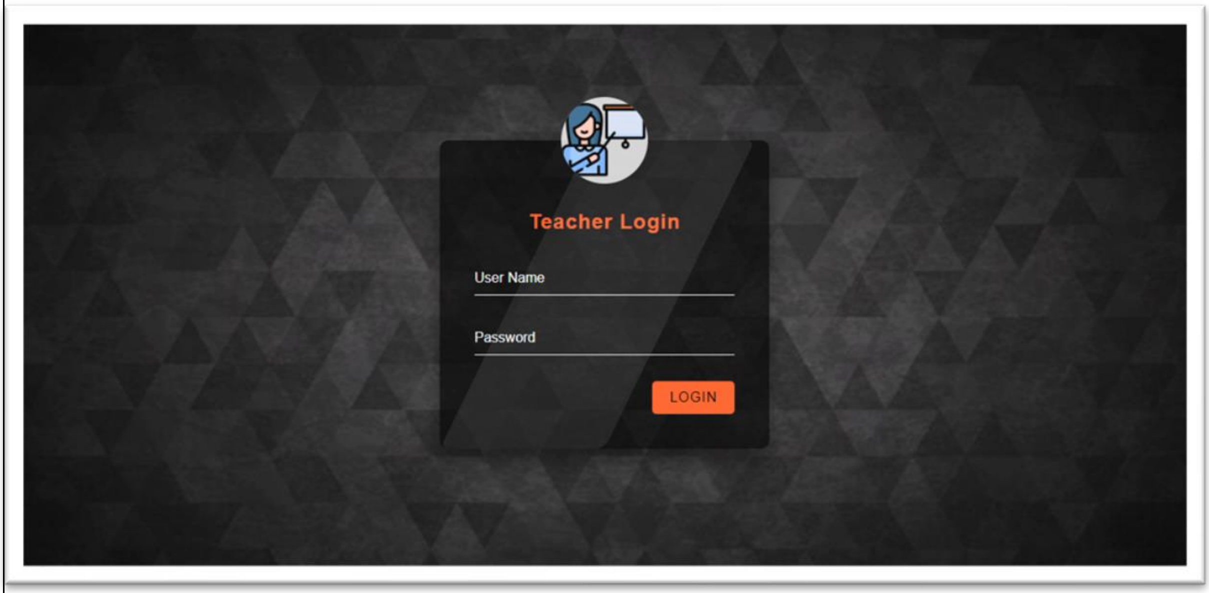
This page lets the student to know the attendance details of the respective subject.

6. STUDENT QR-CODE SCANNER:



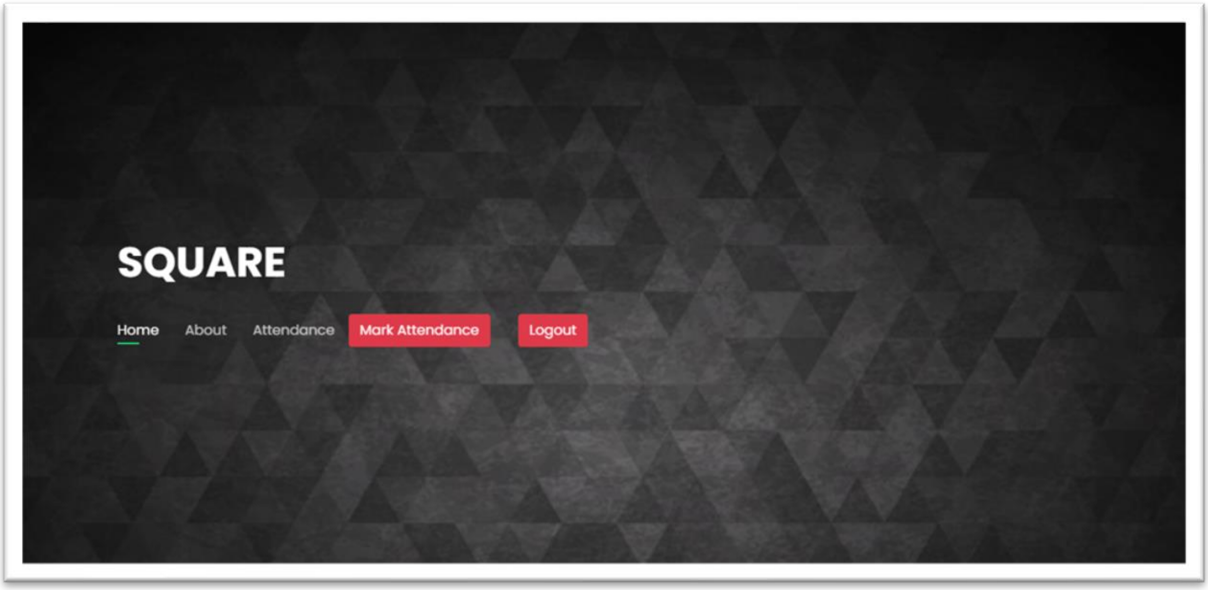
This page lets us to scan the QR code to mark the attendance for the student.

7. TEACHER LOGIN:



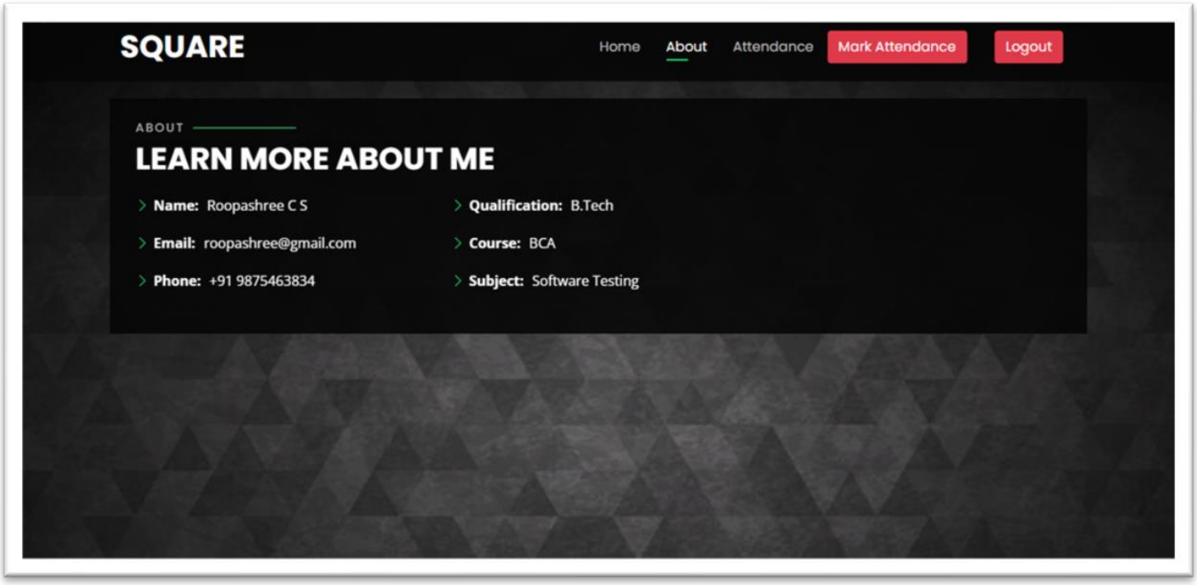
This is the teacher login page where the teacher has to type the gmail is and password to login to the application.

8. TEACHER HOME PAGE:



This page lets us select the action that the student has to perform.

9. TEACHER PROFILE:



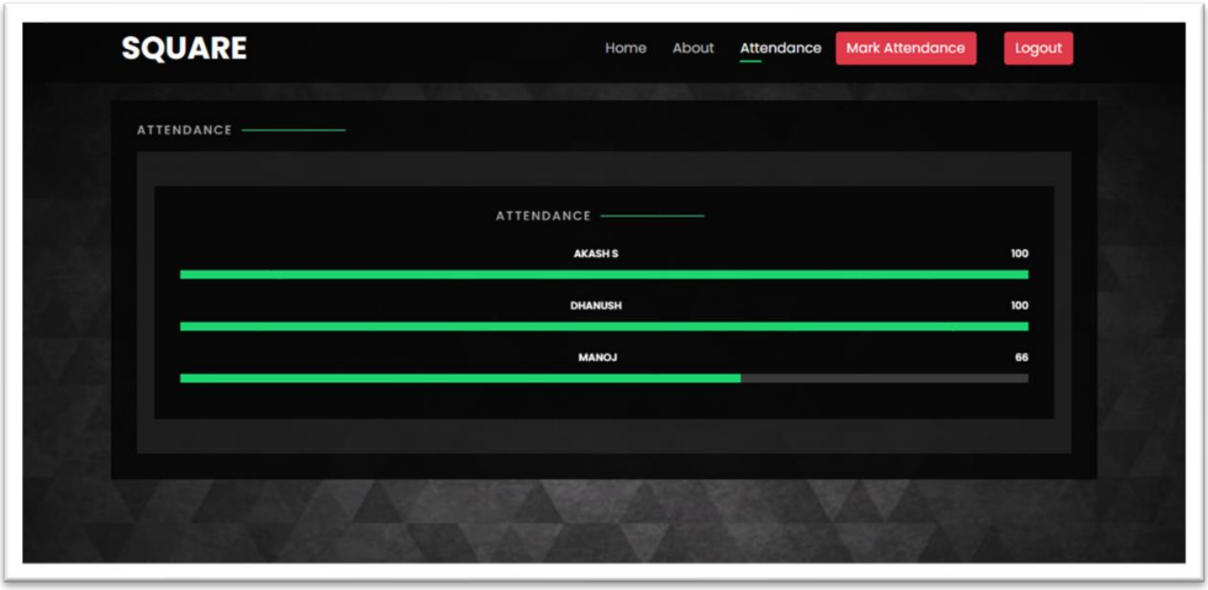
ABOUT

LEARN MORE ABOUT ME

- Name: Roopashree C S
- Email: roopashree@gmail.com
- Phone: +91 9875463834
- Qualification: B.Tech
- Course: BCA
- Subject: Software Testing

This page lets us know the details of the teacher.

10. TEACHER ATTENDANCE DETAILS OF STUDENTS:



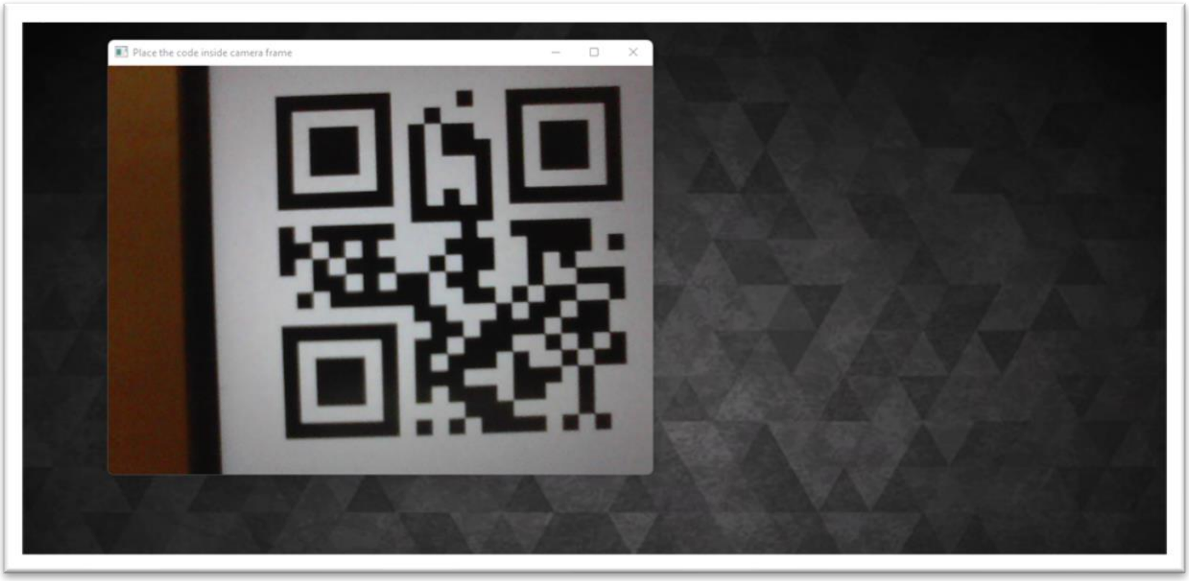
ATTENDANCE

ATTENDANCE

AKASH S	100
DHANUSH	100
MANOJ	66

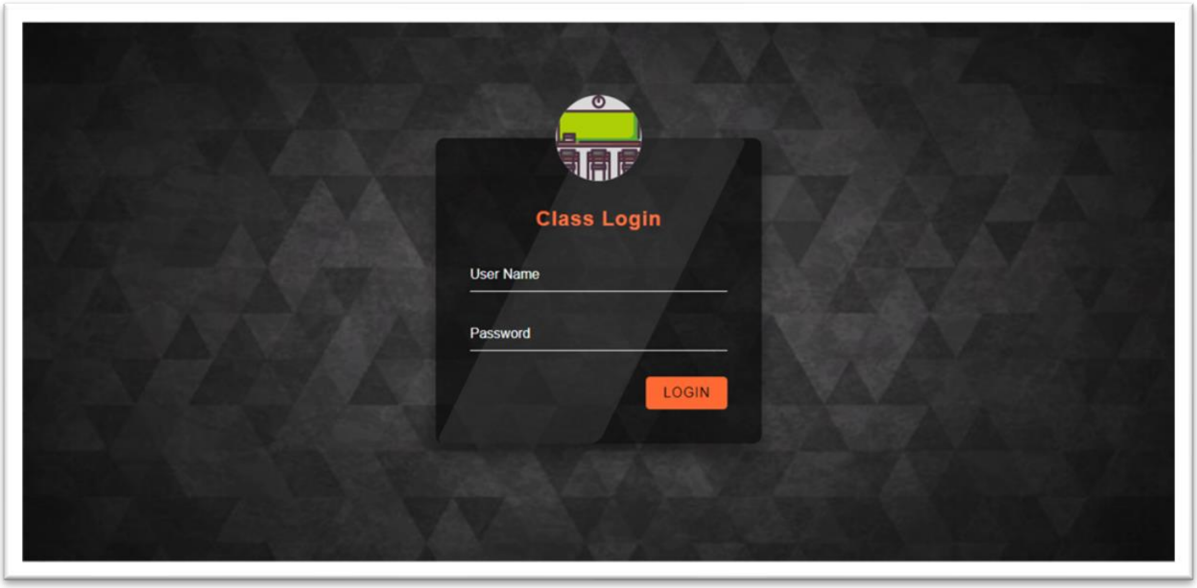
This page lets the teacher to know the attendance details of the students for the respective subject.

11. TEACHER QR-CODE SCANNER:



This page lets us to scan the QR code to mark the attendance for the student.

12. CLASSROOM LOGIN:



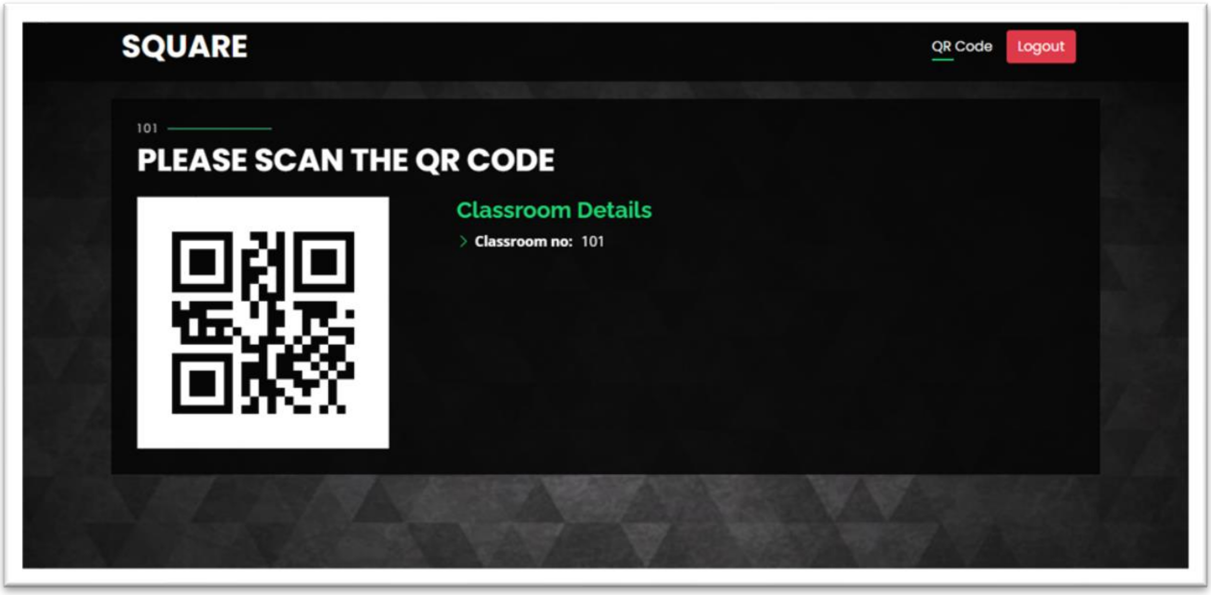
This page lets the user to login to the classroom where the QR code will be generated.

13. CLASSROOM HOME PAGE:

SQUARE
QR Code
Logout

This page lets the user to know what action has to be performed in the page

14. CLASSROOM QR-CODE PAGE:



101

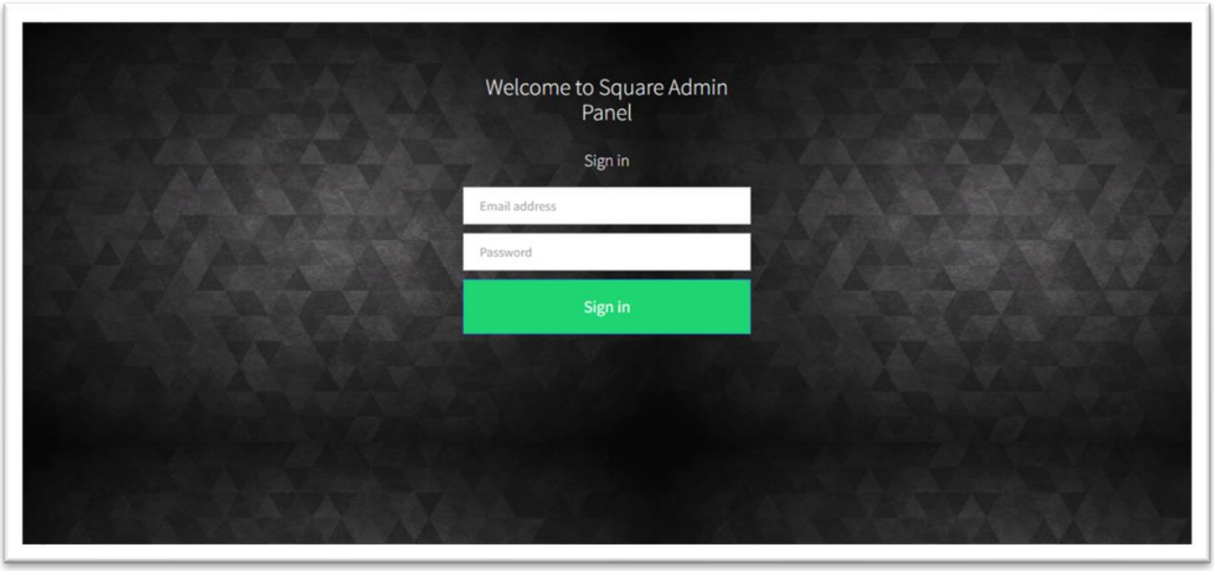
PLEASE SCAN THE QR CODE

Classroom Details

Classroom no: 101

This page displays the QR code using which the student can mark their attendance by scanning the same.

15. ADMIN LOGIN:



This page lets us login as administrator to have complete access to the application information.

14. ADMIN HOME:

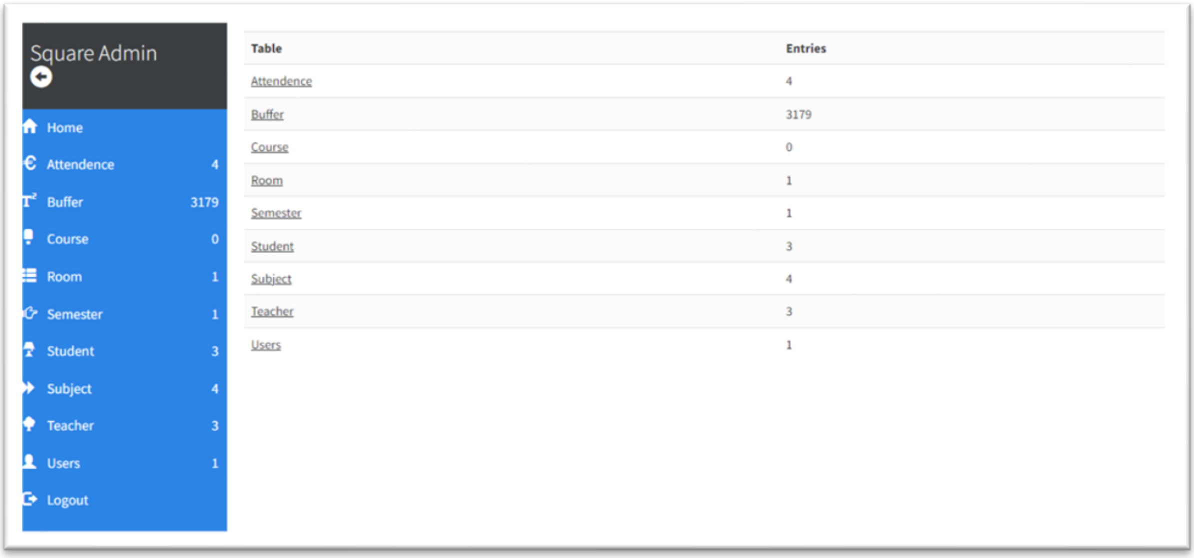


Table	Entries
Attendence	4
Buffer	3179
Course	0
Room	1
Semester	1
Student	3
Subject	4
Teacher	3
Users	1

This page lets the administrator to perform necessary action on the information stored in the database.

7. CONCLUSION AND FUTURE ENHANCEMENT

ATTENDANCE MANAGEMENT SYSTEM USING QR CODE VERIFICATION TECHNIQUE has been developed to overcome the faults present in the existing systems. Through this system the manual work of marking attendance or marking attendance with the devices that comprises of semi be cost-effective, efficient, responsive and feasible from the beginning to the users reducing the risk of proxy marking of attendance.

As future enhancement, some of the additional features could be implemented and integrated into the application code making it reliable and flexible.

The future enhancements could include features where the system could include face recognition which helps in betterment of security against the proxy. And we can include sending message to parents whenever the student is absent. We can also make the application to view the details of subjects such as marks of the tests and more.

REFERENCES

[1]. GATETE Marcel, HARUBWIRA Flaubert, A Development of an Online Student Attendance Management Information System: Case Study “University of Tourism, Technology, and Business Studies”, 2022.

[2]. Xiling Wu, Caihua Zhang, Wei Du, An Analysis on the Crisis of “Chips Shortage” in Automobile Industry--Based on the Double Influence of CIVID-19 and Trade Friction, 2021.

[3]. Hussam Elbehiery, Enhancement of QR code Student’s Attendance Management System using GPS, 2019.