SCHOOL SECURITY SYSTEM USING RFID

A PROJECT REPORT

Submitted by

RAHULKUMAR SINGH (A161419)

A.RAJA MATHIYAZHAGAN (A161420)

S.YOGESH (A161431)

H.MOHAMED RIZWAN (A173442)

Under the Guidance of

Dr. V.G. RAVINDHREN

Dissertation submitted to the

STATE BOARD OF TECHNICAL EDUCATION TAMIL NADU

in partial fulfillment for the award

of

DIPLOMA IN COMPUTER ENGINEERING

SESHASAYEE INSTITUTE OF TECHNOLOGY :: TRICHY-10

(AUTONOMOUS)

APRIL 2019

SCHOOL SECURITY SYSTEM USING RFID

A PROJECT REPORT

Submitted by

RAHULKUMAR SINGH (A161419)

A.RAJA MATHIYAZHAGAN (A161420)

S.YOGESH (A161431)

H.MOHAMED RIZWAN (A173442)

Under the Guidance of

Dr. V.G. RAVINDHREN

Dissertation submitted to the

STATE BOARD OF TECHNICAL EDUCATION TAMIL NADU

in partial fulfillment for the award

of

DIPLOMA IN COMPUTER ENGINEERING

SESHASAYEE INSTITUTE OF TECHNOLOGY :: TRICHY-10

(AUTONOMOUS)

APRIL 2019

SESHASAYEE INSTITUTE OF TECHNOLOGY :: TRICHY-10

(AUTONOMOUS)

BONAFIDE CERTIFICATE

Certified that this project report "**SCHOOL SECURITY SYSTEM USING RFID**" is the bonafide work of

"RAHULKUMAR SINGH (A161419)

A.RAJA MATHIYAZHAGAN (A161420)

S.YOGESH (A161431)

H.MOHAMED RIZWAN (A173442)"

who carried out the project work under my guidance.

SIGNATURE
Dr. V.G. RAVINDHREN
HEAD OF THE DEPARTMENT

Department of Computer Engineering

SIGNATURE
Dr. V.G. RAVINDHREN
GUIDE
HOD, Lecturer(S/G)
Department of Computer Engineering

Submitted for the Project Viva – Voice Held on: _____

EXTERNAL EXAMINER **INTERNAL EXAMINER**

ABSTRACT

The main objective of developing this project is to ensure the security of the students and give the parents relief. School Security System (SSS) is a software which is helpful for students as well as the parents. In this present situation the parents remain unknown whether their ward have reached schoolor not. It makes the parents unsecured, uncomfortable and worried. Our School Security System informs the parents if their child have reached the school premises or not.

This project is beneficial for both the students and parents in keeping themselves safe and free from negative thoughts. The system designed is meant to detect the child's presence in the school premises and inform their parents the in-time of the child. The parents will be informed by the means of SMS and they also can generate the attendance report by accessing the parents interface. This system also sends an alert message to their parents in case the students are not still present at the school start time. The system will also record all IN and OUT details of the student, which will also render a great help to the school's administration.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The recent scenario shows that humans are discovering new methods which will reduce child kidnapping and increase comfort. In developing countries child security and monitoring child activity is difficult task to accomplish. Nowadays, parents are worried about their children because of the high rate of kidnapping. Moreover, Parents are having long working hours, so they simply do not have as much time to spend for their children Moreover, they will be proceeded by kidnapper before they enter the school. So, it is the responsibility for the school to take care of their students and they also know In-time and able to send an alert message to their parents if the students are not at the school start time. However, it is not easy to do manually. The school authorities cannot check their students individually and cannot send an alert message to their parents. So, the suitable solution for this problem is by designing a system that will alert the parents automatically.

Latest Technologies enabled advancement in mobile device, hand full of devices available to track and monitor a child. But cost effective systems are very few. The RFID fulfills the technology need of combining cost effective and richer features for general consumers. There are several applications available addressing the security for children's and women's. But with access privilege with smart ID card is unique. The combination of this RFID will serve the society to build a bridge between Parents-Student-Security

CHAPTER 2
LITERATURE REVIEW

2.1 EXISTING SYSTEM

A literature review has shown that there are many studies that made use of Radio Frequency identification (RFID) as a system that transmits the identity of an object using radio waves. This identity is transmitted in the form of serial number that distinguishes each object from others. The RFID system consists of an RFID reader and an RFID tag. The tag consists of the microchip that is connected to an antenna; microchip can store a maximum of 2 KB of data, which may include data and information about the product, manufacturing date, and destination further, the author also observed that the ability of the reader field decreases quickly with increasing distance, which defines the area of reading to 4-5 meter distance using VHF 860-930MHz.

Another research introduced a system that monitors children inside the bus in a safe manner. Each student carries a unique RFID card. The card is embedded in each of the student's school bags. Whenever a student enters or exits from the bus, the reader records the time, date, and location and then transfer the data into a secure database and this does not require any action from the drivers and students.

Radio Frequency Identification (RFID) has been used in a number of practical applications, such as improving supply chain management, tracking household pets, accessing office buildings etc. RFID is used to automatically identify people, objects, and animals using short range radio technology to communicate digital information between a stationary locations (reader) and proposed to communicate between the server and the user present at the remote location. Various other technologies such as ZigBee, Bluetooth, etc. can also be used. It is also seen that RFID Card and RFID Scanner are being used in metro stations.

2.2 PROPOSED SYSTEM

In this system, we are mainly focusing on the security of students when they are bound to reach the school premises. Each students of the school are given with an RFID Card which they would be using as their Identity Card. A small and slim radio frequency wave emitter is present inside the RFID Card. At the school gate, some hardware components are required to be assembled and fixed. We would be using a micro-controller called NODE MCU which is embedded with a Wi-Fi module called ESP8266 and we would also be using an RFID scanner. When the students reaches the school gate, they are to scan their identity card towards the RFID Scanner from a maximum distance of 5mm. Then the RFID Scanner scans the radio wave and converts it into a unique RFID value.

The NODE MCU fetches the RFID value also known as RFID number from the scanner and sends it to the computer via Wi-Fi module using the HTTP protocol. Then the RFID number gets stored in the local database of the school administration and alert message is sent to the parents regarding the arrival of their child. In case the student doesn't scan his/her card before the school start time, the parent of the student will be alerted with a message informing that their child is absent.

The proposed system also provides a very interactive User-Interface for the three main users of the system, i.e. Institute Interface, Parents Interface and Developer Interface. Under the institute interface, the institute members of the respective school is allowed to control database information as well as the attendance information of each and every students. They are also given the option to generate attendance report based on different scenario provided by them. Under the parents interface, they are given the ability to check their child's summary report and attendance report. Lastly, under developer interface, the developers of the system are given with all the rights in the modification of the system. All the error or fault handling tools are provided under this interface.

CHAPTER 3
SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

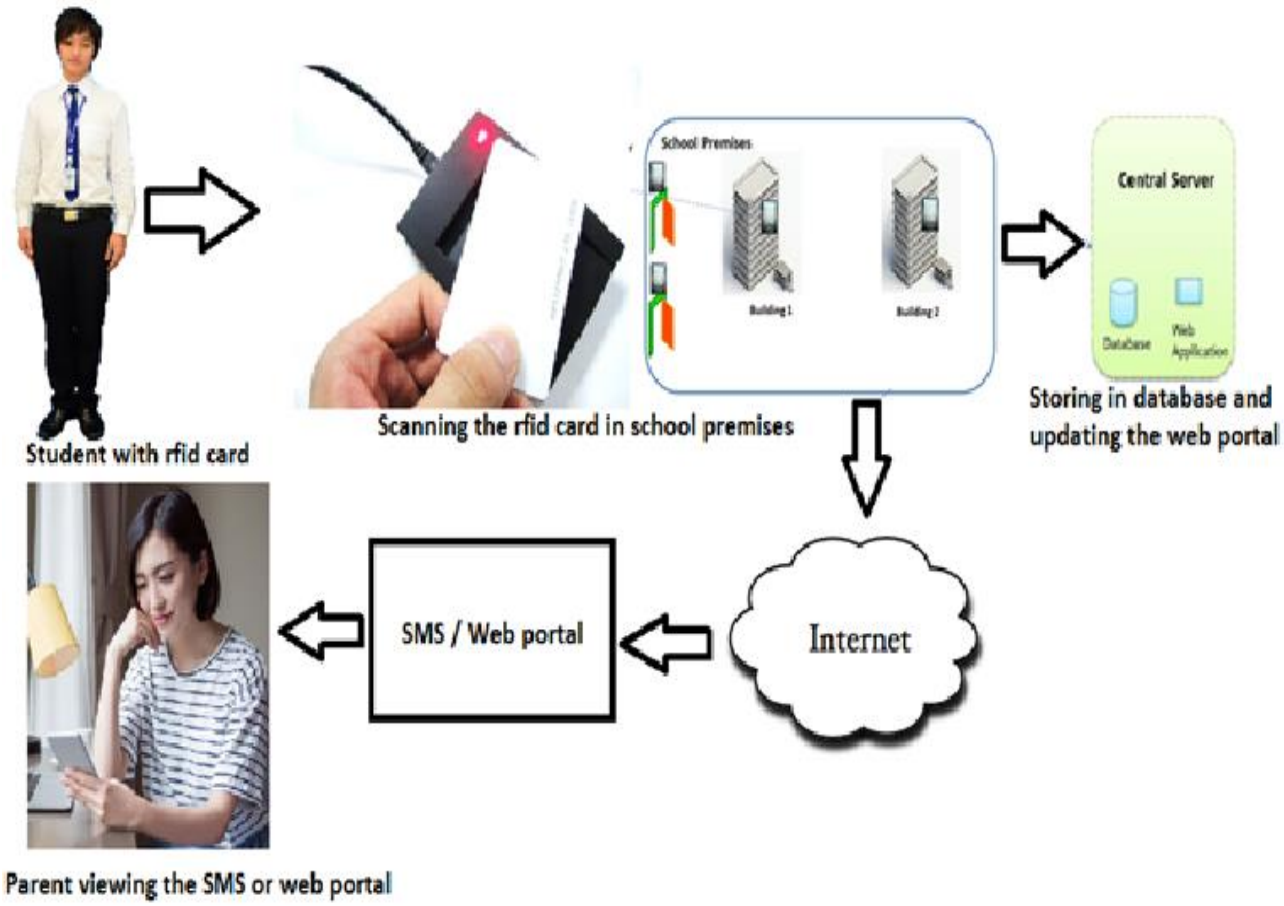
Sl. No.	Component	Description	Quantity
1.	RFID Scanner (RC522)	Used to scan the RFID card or RFID tag.	1
2.	NodeMcu V3 with ESP8266	Used to fetch RFID number from the scanner and sends the number to computer.	1
3.	RFID Card	Contains the RFID number.	1
4.	RFID Tag	Contains the RFID number.	1
5.	Jumper Wire (Female to Female)	Used to connect the micro-controller and the scanner.	7
6.	USB Cable	Used to give power supply to NodeMcu.	1

3.2 SOFTWARE REQUIREMENTS

Sl. No.	All-Inclusive	Software Name	Description
1.	IDE	Arduino IDE	Used to code NodeMcu and RFID module.
2.	Programming and Scripting Language	PHP, C language, HTML5, CSS3, JavaScript, Bootstap	Used for Server Side Scripting, Client Side Scripting and programming NodeMcu
3.	Database and Web Server	MySQL, Apache (XAMPP)	Used as local Database and local web server
4.	Framework	WordPress	Used for Interface designing and code integration.
5.	API	TextLocal	Used to send SMS.

CHAPTER 4
IMPLEMENTATION

4.1 SYSTEM ARCHITECTURE



SHORT DESCRIPTION:

- The student scans its identity card to the RFID Scanner when he reaches the school gate.
- The scanner scans the value and stores it to the database.
- SMS is sent to the parents and the user interface are updated with the newly arrived data.

4.2 MODULES

4.2.1 INSTITUTE INTERFACE

It is an interface specially designed for the use of the institute members of the school. In order to gain access to this interface, the staff or any of the institute member are to provide their respective login credentials. After the user is verified he/she is an authorized user then all the interface's facilities are provided for his/her use. The main use of the module is to control the security and attendance system from them institute side.

The Institute interface module itself contains a number of sub-modules with different functionalities. The sub-modules are listed below:

STUDENT DATABASE VIEWER AND EDITOR

Under this sub-module, the user is given the option to either add new student(s), edit the details of a student or delete a student from the database.
If the user wishes to add new student(s) then he is to specify the number of students he wishes to add. Therefore, based on the users' input the form in the next page will be iterated. If the user wants to edit some of the details of a particular student then he have to enter the registration and search for the student. If the student in available in the database then his details are all displayed in the screen.

The user can now edit all the details as necessary except for the RFID Number which can only be edited from them developer side. The final option available under this sub-module is the delete student option. If the user wishes to delete the details of a student who is no longer part of the institute then he has to search for the student's registration number and press the delete option displayed. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

DISPLAY ALL STUDENTS

This is a special sub-module only available to the Institute interface. This sub-module is used to display all the details of the students that are registered with the specific institute. But the institute is not given with the option to edit the student's details under this sub-module. It is only used for viewing purpose. It also displays the total number of students that are part of the institute. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

SPECIFIC STUDENT REPORT GENERATOR

This is the third sub-module available to the Institute interface. The functionality provided under this sub-module is that the user can generate the attendance report of a specific student. Firstly, the user is to specify the registration number of the student. Then in the following page, several options are listed out for the user to choose. The attendance report of the student can be generated based on three different parameters under this sub-module.

The first parameter asks for a specific date to generate the report. The other parameter asks for a range of date to generate the report. Lastly, it gives the option to generate the full RFID card scanned report of the student.

The details that is being displayed under this sub-module may include RFID no., date of scan, time of scan, SMS status etc. This module gives the user the freedom to choose the way of report generation from the vast predefined options. The above wide field of functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

ATTENDANCE GENERATOR

This is the fourth sub-module provided under the Institute Interface module. The functionality provided under this sub-module is that the user can generate the attendance report. There are various options through which the report

can be generated. If the user only has the RFID number of the student then this module provides the user with the option to search the student using the RFID number.

The attendance report contains many informative details such as entry and exit date and time, SMS status and many more. If the user wishes to view the attendance report of a whole class then he can select the class and generate the report. There are a number of different other options given under this module. The attendance log of the whole school also can be generated under a single click by specifying the date.

The attendance log between the ranges of two dates can also be generated under this multi-purpose sub-module. There are also other options that it provides. The user can also find out the students who are coming late to the school by specifying the class and date. The attendance log report between two different times can also be generated under this sub-module. All the above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

ATTENDANCE EDITOR

This is the fifth sub-module under the Institute Interface module. Here, the user is given the option to make changes to the attendance log. If a teacher wants to add a student PRESENT in the attendance log for some reason then the teacher should search for the student's registration number and select the "ADD STUDENT PRESENT" button. After adding, the teacher is given with an option to send an alert message to the parent of the student informing that their child have reached the school premises.

If the user wishes to edit the attendance log of a particular student then the user must search for the students' registration number and edit accordingly. Under this option, all the attendance log of the student is displayed. The user can edit the details that are begin displayed or delete a specific log. The above

functionalities could be achieved by using PHP, TextLocal API, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

SMS SENDER

This is also a special sub-module available only to the Institute interface. This sub-module allows the user to send SMS to the parent or guardian of a specific student. The user can specify the student by either giving the registration number of the student or the RFID number. If the user knows the contact number of the recipient then he is allowed to enter the number and directly send the SMS. The user can enter the message in a textbox before sending it. After entering the message content the user can press the send button, which will send an SMS to the recipient along with the message content. The above functionalities could be achieved by using PHP, TextLocal API, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

INSTITUTE EMERGENCY CONTACTS

This sub-module is a simple yet important module available under the Institute interface. This sub-module contains a list of emergency numbers which might render great help in times of need. The emergency numbers listed under this sub-module are child helpline number, police number, ambulance number, and firefighter number. The simple yet effective list could be prepared by using the client side scripting languages.

INSTITUTE REQUEST ASSISTANCE

This is the last sub-module under the Institute Interface module. While the faculty members uses the Institute interface there might arise a number of difficulties and doubts regarding the working and performance of the interface. Therefore, we created this sub-module to solve that very problem. This sub-module gives the user a way to communicate with us through E-Mail. If a user is

in dilemma on what to do next, he/she can directly open this sub-module, press the send EMAIL button and email us. And our 24*7 instant support team will help the user out by one of the means of communication or if necessary, by visiting the user's location. The above functionalities could be achieved by using PHP, PHPMailer, Apache, WordPress, HTML5, CSS3 and JavaScript.

4.2.2 PARENTS INTERFACE

The Parents Interface module is specifically designed for parents use. There are many functionalities provided under this module and in order to gain access of these functionalities the parents must be first verified as authorized user by providing their login credentials. The parents can view many informative details about their child under this section.

There are many sub-modules defined under this module. The sub-modules are listed below along with its description respectively:

STUDENT SUMMARY

This is the first sub-module provided under the Parents Interface module. The main use of the module is to generate the short summary reports about his/her child. The summary report may contain some profile details of the child like name, address, class etc. The summary report also contains the child's attendance percentage, most recent attendance log, days present, days absent and many more. The sub-module doesn't provide the option to edit the details of the ward. This sub-module is only meant for viewing purpose only. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

REPORT GENERATOR

The Report Generator is one of the most useful sub-module provided under the Parents Interface module. Here, the parents can manipulate with their child's

attendance log. There are a number of options that we have provided on how to generate the attendance log of the student. All details like time-in, time-out, date and many other information can be accessed by the parents under this sub-module.

The first option provided under this sub-module is to completely generate the full attendance log of the student and display it to the parents. The next option is to display the attendance log of a specific date. Here, the parents will have to select a specific date to generate the log of the date they selected. If the parents want to generate attendance log of their ward between ranges of two dates, then this sub-module also provides them with an option. The parents can select the two ranges of dates and generate the log report.

If the parent wishes to check if their ward reached the school premises on time then the sub-module again turns out smart. It also provides a way on how to generate the number of times their ward are late to school. Attendance log of a specific date and between two different ranges of time can also be generated under this sub-module. But, the parents are not given with the privilege to edit the attendance log. They can use it only for viewing and evaluating purpose. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

PARENTS EMERGENCY CONTACTS

The fourth sub-module provided under the Parents Interface is the Parents Emergency Contacts. If there arises any emergency situation, all necessary emergency details are provided under this sub-module. This sub-module is a simple yet important module available under the Institute interface. This sub-module contains a list of emergency numbers which might render great help in times of need. The emergency numbers listed under this sub-module are child helpline number, police number, ambulance number, and firefighter number. The

simple yet effective list could be prepared by using the client side scripting languages.

PARENTS REQUEST ASSISTANCE

This is the last sub-module under the Parents Interface module. The use of this sub-module is to provide the contact details of the developer when the user is unsure regarding the operation or the functionality of any of the sub-modules. While the parents or guardians uses the Parents interface there might arise a number of difficulties and doubts regarding the working and performance of the interface. Therefore, we created this sub-module to solve that very problem. This sub-module gives the user a way to communicate with us through E-Mail.

If a user is in dilemma on what to do next, he/she can directly open this sub-module, press the send EMAIL button and email us. And our 24*7 instant support team will help the user out by one of the means of communication or if necessary, by visiting the user's location. The above functionalities could be achieved by using PHP, PHPMailer, Apache, WordPress, HTML5, CSS3 and JavaScript.

4.2.3 DEVELOPER INTERFACE

When there is a problem with the Institute Interface or when there is the problem with the Parents Interface, the Developer Interface comes into action. The Developer Interface is only meant for the use by the developers of this security system. All user login details are also controlled under this module. But all this rights cannot be given without any authentication. In order to gain access to the developer login, the user should first provide the highly classified login credentials and only after verification of authenticity the access will be granted by the system to the user.

The Developer Interface module is also made up of a number sub-modules. The sub-modules and its description are given below:

INSTITUTE DATABASE VIEWER AND EDITOR

This is the first sub-module under the Developer Interface module. Under this sub-module the user can add a new institute into the database, edit/view the institute database or even delete the database of an institute if no longer part of the security system. If the user wishes to add a new institute into the database they just have to press the add button and give the necessary details about the institute.

The user also can edit or delete the details of the institute if necessity arises. And for doing so, the user has to select any institute from the given list of institutes and in the following page all the details of the institute will be displayed. Then the user can edit, view or even delete the institute details according to the requirements. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

USER ACCESS DATABASE VIEWER AND EDITOR

This is the second sub-module provided only for the developer. This module controls the entire access control of the security system. With the help of this sub-module the user can add a new user to the system or edit the login details of an existing user or even delete the user. The login details of the school, parents and developers can all be controlled from this sub-module. In order to control it, the user just need to select the option which he wants to edit and the remaining is just the magic of the system. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

STUDENT DB MASTER

Under this sub-module, the user is given the option to either add new student(s), edit the details of a student or delete a student from the database of an institute. If the user wishes to add new student(s) then he is to specify the number of students he wishes to add. Therefore, based on the users' input the form in the

next page will be iterated. If the user wants to edit some of the details of a particular student then he have to enter the registration and search for the student. If the student in available in the database then his details are all displayed in the screen.

Unlike the Institute Interface, the user can edit all the details as required even the RFID Number. The final option available under this sub-module is the delete student option. If the user wishes to delete the details of a student who is no longer part of the institute then he has to search for the student's registration number and press the delete option displayed under the sub-module. The above functionalities could be achieved by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

CONTACT DETAILS GENERATOR

This is also another unique sub-module available only to the developers. This sub-module is used to generate contact details of a student or an institute in a blink of an eye. There are several options given under this sub-module on how to generate the contact details.

To generate the contact details of a student, the user has to either specify the Registration number or the RFID number of the student. Then the user can generate the required contact details. The details may include father or mother name, address, phone number, email address etc.

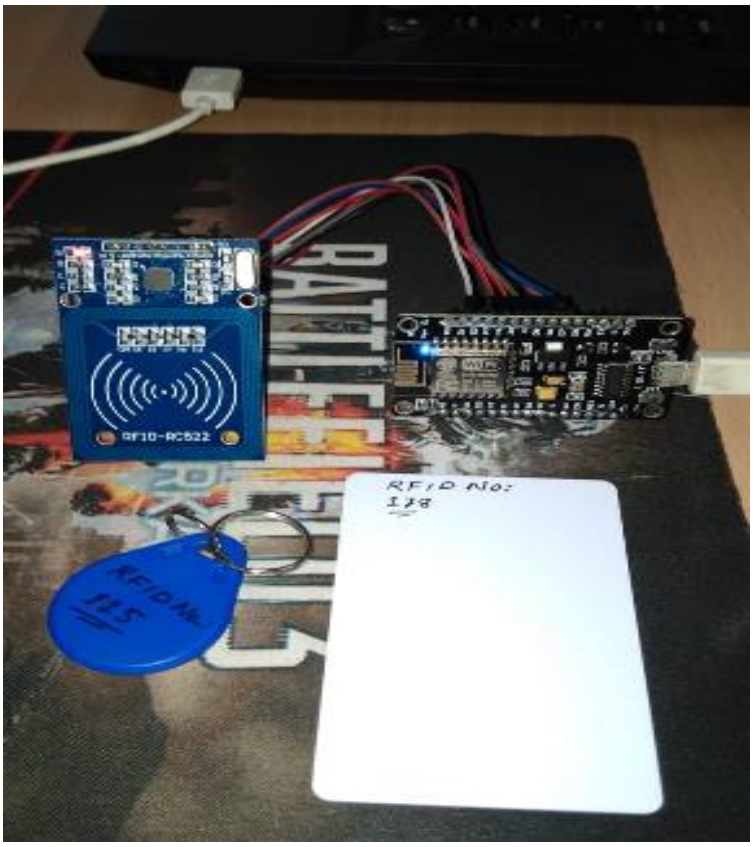
To generate the contact details of an institute, the user has to select the institute from the list of institute displayed in the screen. Then after selecting the institute and pressing the generate button, the contact details will be displayed. The details may include the contact number, address, and email address of the institute respectively.

The contact generator could be developed by using PHP, MySQL, Apache, WordPress, HTML5, CSS3 and JavaScript.

DEVELOPER EMERGENCY CONTACTS

The last sub-module provided under the Developer Interface is the Developer Emergency Contacts. If there arises any emergency situation, all necessary emergency details are provided under this sub-module. This sub-module is a simple yet important sub-module available under the Institute interface. This sub-module contains a list of emergency numbers which might render great help in times of need. The emergency numbers listed under this sub-module are child helpline number, police number, ambulance number, and firefighter number. The simple yet effective list could be prepared by using the client side scripting languages.

4.2.4 RFID, NODE MCU AND SMS



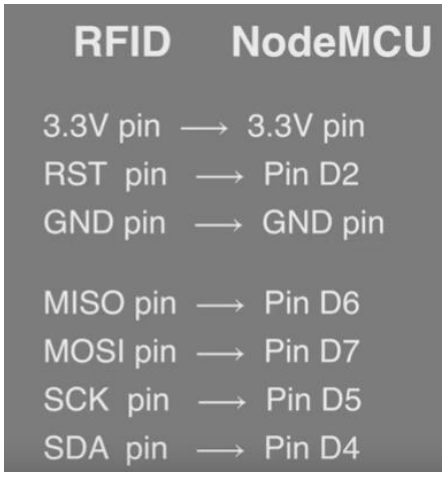
The RFID, NODE MCU and SMS module is the most important module in this system. The heart of the system lies under this trinity i.e. RFID, NODE MCU and SMS.

The RFID, NODE MCU is first of all connected using the female-femal e jumper wires, then 3.3V power supply is provide through the USB cable. The necessary coding for RFID and NODE MCU are done using the Arduino IDE and C programming language. Libraries for RFID Module, ESP8266 and NODE MCU are all used for its working. Then the coding is flashed into the NODE MCU and then tested.

When the RFID Card or RFID Tag is scanned to the RFID scanner, the unique RFID number is received. Then NODE MCU fetches the unique RFID number from the RFID reader and sends the RFID number to the computer with the help of a Wi-Fi module called ESP8266 by using the HTTP protocol. The unique RIFD number is then stored in the database and the contact detail of the student possessing the unique RFID number is noted. Finally, an alert message to the parent of the student saying that your child have reached is sent with the help of the noted contact detail.

The above functionalities could be achieved by using the following software and hardware components: Arduino IDE, C Language, PHP, TextLocal API, MySQL, Apache, NODE MCU, RFID reader, RFID Card/Tag, USB cable, jumper wires.

CONNECTION DIAGRAM



The hardware components required for this project are connected based on the pin diagram shown above. The hardware components were connected using female to female jumper wires. The NodeMCU is connected to the computer using a micro USB cable.

WHAT IS RFID?

RFID is an acronym for "radio-frequency identification" and refers to a technology whereby digital data encoded in RFID tags or smart labels (defined below) are captured by a reader via radio waves. RFID is similar to barcoding in that data from a tag or label are captured by a device that stores the data in a database. RFID, however, has several advantages over systems that use barcode asset tracking software.

The most notable is that RFID tag data can be read outside the line-of-sight, whereas barcodes must be aligned with an optical scanner. If you are considering implementing an RFID solution, take the next step and contact the RFID experts at AB&R (American Barcode and RFID).

HOW DOES RFID WORKS?

RFID belongs to a group of technologies referred to as Automatic Identification and Data Capture (AIDC). AIDC methods automatically identify objects, collect data about them, and enter those data directly into computer systems with little or no human intervention. RFID methods utilize radio waves to accomplish this. At a simple level, RFID systems consist of three components: an RFID tag or smart label, an RFID reader, and an antenna.

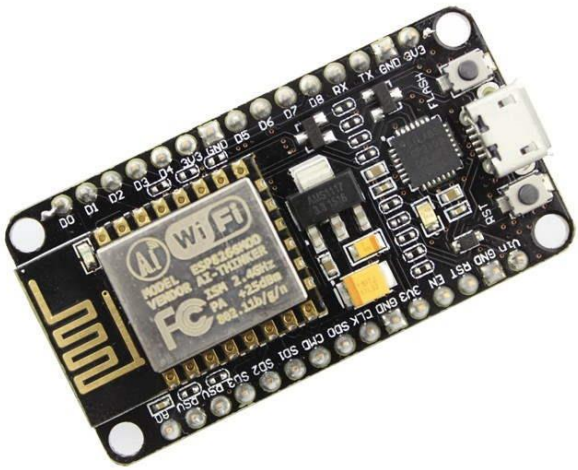
RFID tags contain an integrated circuit and an antenna, which are used to transmit data to the RFID reader (also called an interrogator). The reader then converts the radio waves to a more usable form of data. Information collected from the tags is then transferred through a communications interface to a host computer system, where the data can be stored in a database and analyzed at a later time.

RFID TAGS AND SMART LABELS

As stated above, an RFID tag consists of an integrated circuit and an antenna. The tag is also composed of a protective material that holds the pieces together and shields them from various environmental conditions. The protective material depends on the application. For example, employee ID badges containing RFID tags are typically made from durable plastic, and the tag is embedded between the layers of plastic.

RFID tags come in a variety of shapes and sizes and are either passive or active. Passive tags are the most widely used, as they are smaller and less expensive to implement. Passive tags must be "powered up" by the RFID reader before they can transmit data. Unlike passive tags, active RFID tags have an onboard power supply (e.g., a battery), thereby enabling them to transmit data at all times.

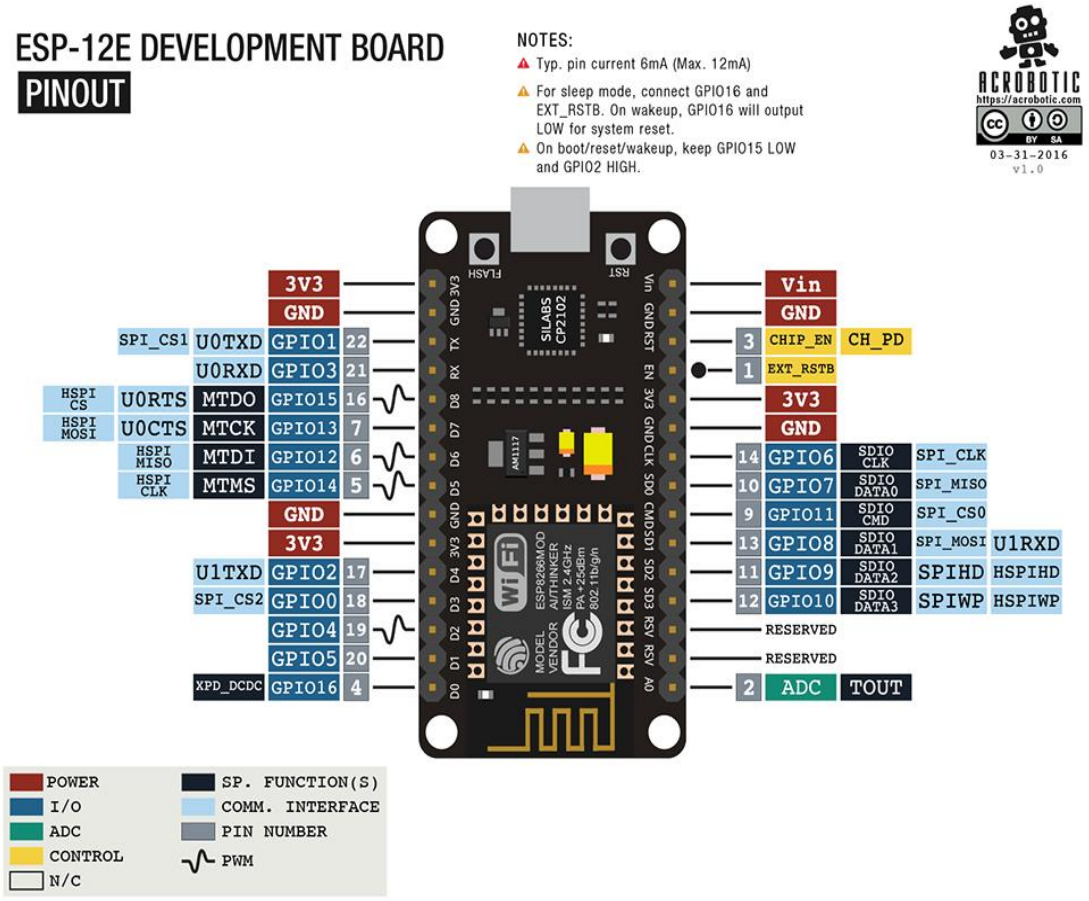
WHAT IS NODEMCU?



The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains all crucial elements of the modern computer: CPU, RAM, networking (wifi), and even a modern operating system and SDK. When purchased at bulk, the ESP8266 chip costs only $2 USD a piece. That makes it an excellent choice for IoT projects of all kinds.

However, as a chip, the ESP8266 is also hard to access and use. You have to solder wires, with the appropriate analog voltage, to its PINs for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. And, you have to program it in low-level machine instructions that can be interpreted by the chip hardware. While this level of integration is not a problem when the ESP8266 is used as an embedded controller chip in mass-produced electronics, it is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

Borrowing a page from the successful playbooks of Arduino or a Raspberry Pi, the NodeMCU project aims to simplify ESP8266 development. The pin-out diagram of NodeMCU is given below:



TEXTLOCAL API

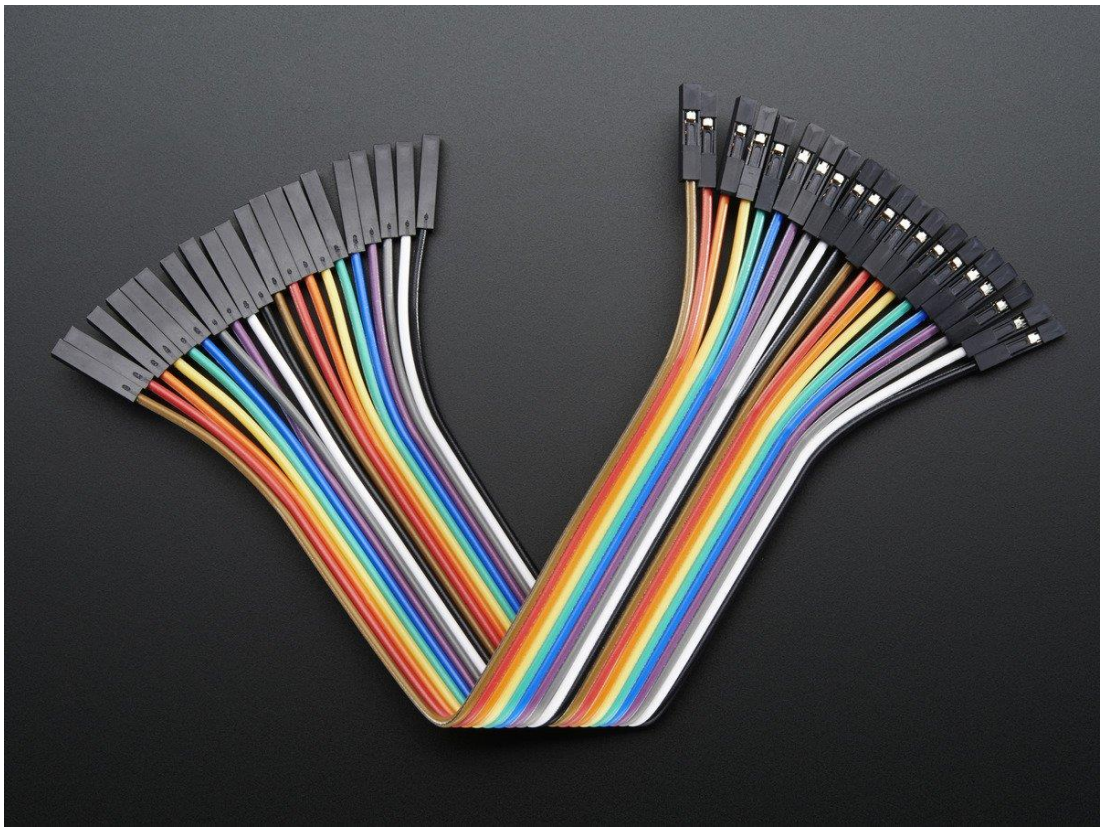
Textlocal is a mobile communications company founded in 2005 with offices based in Malvern and Chester.

In 2005, Alastair Shortland and Darren Daws decided they wanted to transform the way businesses, individuals and groups communicate using mobile. Txtlocal Ltd was then formed.

Since then, with the development of their Messenger product, it has allowed over 165,000 users to communicate via SMS, MMS and Mobile Web. Its services include Inbound and outbound SMS and MMS communications using a web based control panel, SMS Gateway and MMS API, mobile web page creation, mobile forms and surveys, SMS attachments and mobile vouchers and loyalty cards.

JUMPER WIRES (Female to Female**)**

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. What does the color means? Though jumper wires come in a variety of colors, the colors don't actually mean anything. This means that a red jumper wire is technically the same as a black one.



But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power.

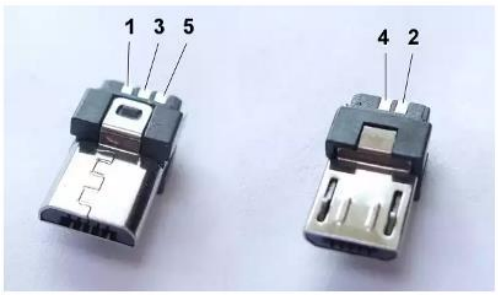
MICRO USB

Micro USB is a miniaturized version of the Universal Serial Bus (USB) interface developed for connecting compact and mobile devices such

as smartphones, MP3 players, GPS devices, photo printers and digital cameras.

Micro USB connectors exist or have existed in three forms: micro A, micro B and micro USB 3. USB 3 micro is much like micro B, but with an additional pin group on the side for twice the wires, enabling USB 3's greater speed. Images of micro USB and pin diagram is shown below:

Pinout



Pin	Name	Cable color	Description
1	VCC	Red	+5 VDC
2	D-	White	Data -
3	D+	Green	Data +
4	ID	n/a	USB OTG ID
5	GND	Black	Ground

Like standard USB, the micro versions are plug-and-play and hot-swappable. The USB peripheral bus standard was developed jointly by Compaq, IBM, DEC, Intel, Microsoft, NEC, and Northern Telecom. The technology is available without charge for all computer and device vendors.

4.3 LIST OF TABLES AND ITS STRUCTURES

The list of tables used in this system are listed below along with its structures respectively.

1.

Table Name: login_parent		
Field Name	**Data type**	**Constraints/key**
inst_id	Varchar(20)	Primary key
stud_id	Varchar(20)	Primary key
password	Varchar(100)	-

The login_parent table is used to store the login details of the parents in the database.

2.

Table Name: login_teacher		
Field Name	**Data type**	**Constraints key**
inst_id	Varchar(20)	Primary key
staff_id	Varchar(20)	Primary key
Pass	Varchar(100)	-

The login_teacher table is used to store the login details of the teachers/faculty in the database.

3.

Table Name: dev_login		
Field Name	**Data type**	**Constraints key**
username	Varchar(30)	Primary key
password	Varchar(100)	-

The dev_login table is used to store the login details of the developers in the database.

4.

Table Name: institute_master		
Field Name	**Data type**	**Constraints key**
institute_id	Varchar(20)	Primary Key
Name	Varchar(100)	Not null
contact_no	Varchar(20)	Not null
email_id	Varchar(100)	Not null
address	Varchar(1000)	Not null

The institute_master table is the table where all the details of the institute which are part of the school security system is stored.

5.

Table Name: attendance_log		
Field Name	**Data type**	**Constraints key**
Slno	Int	A.I.
institute_id	Varchar(20)	Not null
Rfid	Varchar(20)	Not null
Regno	Varchar(20)	Not null
Class	Varchar(10)	Not null
date1	date	Not null
time1	time	Not null
present_stat	Varchar(10)	Not null
msg_stat	Varchar(20)	Not null

The attendance_log table is the table where the details of every RFID Card scanned is stored.

6.

Table Name: inst_working_days		
Field Name	**Data type**	**Constraints key**
institute_id	Varchar(20)	Primary key
working_days	bigint	Not null

The inst_working_days table is the table where the number of working days of every institute are updated daily.

7.

Table Name: student_master		
Field Name	**Data type**	**Constraints key**
institute_id	Varchar(20)	Primary key
Rfid	Varchar(100)	Primary key
Regno	Varchar(20)	Primary key
Name	Varchar(50)	Not null
Class	Varchar(10)	Not null
section	Varchar(20)	Not null
f_m_name	Varchar(50)	Not null
address	Varchar(100)	Not null
contact_no	Varchar(10)	Not null
email_id	Varchar(100)	Not null

The student_master table is the table where all the necessary details of the student is stored.

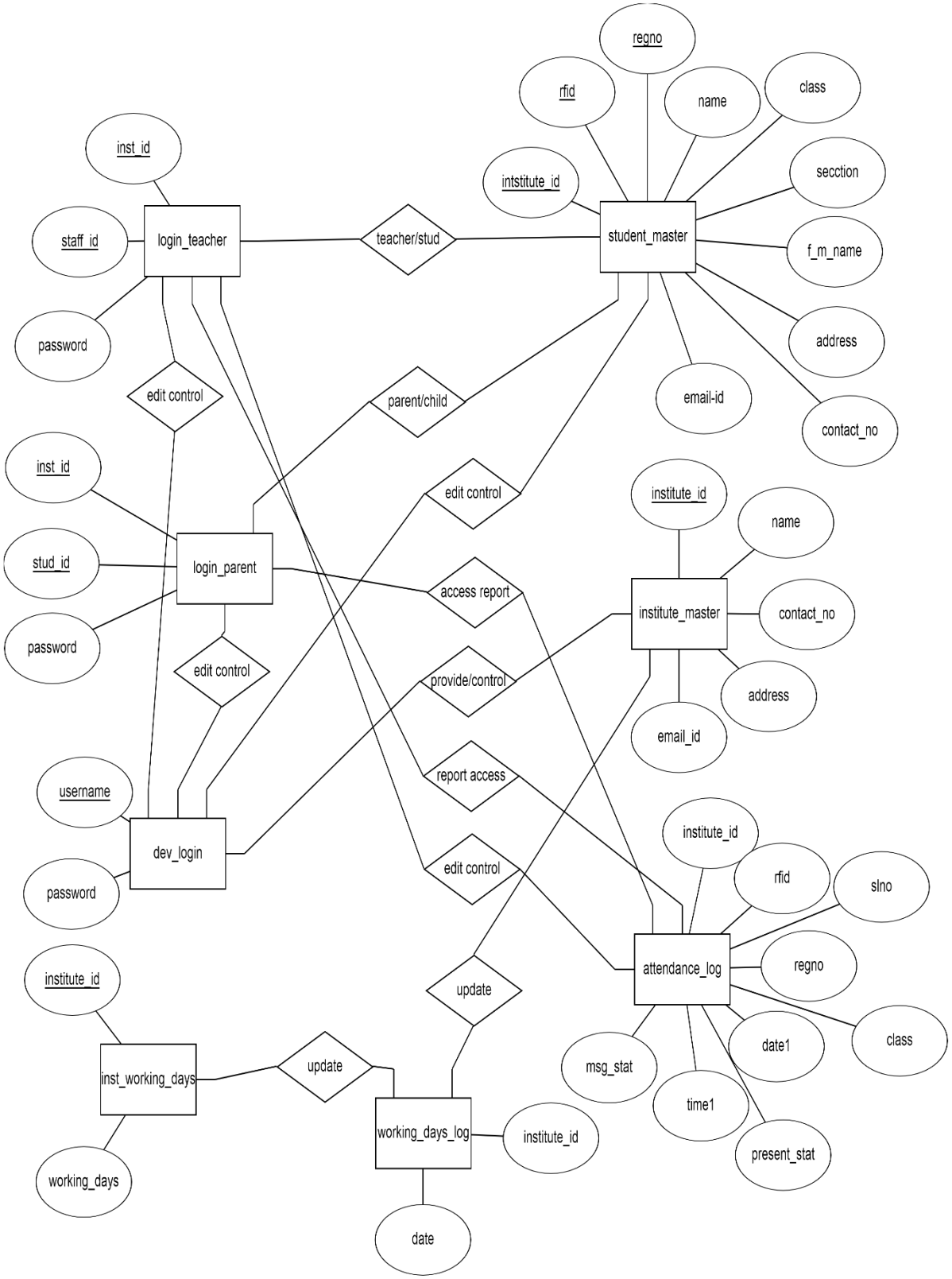
8.

Table Name: working_days_log		
Field Name	**Data type**	**Constraints key**
institute_id	Varchar(20)	Primary key
Date	Date	Not null

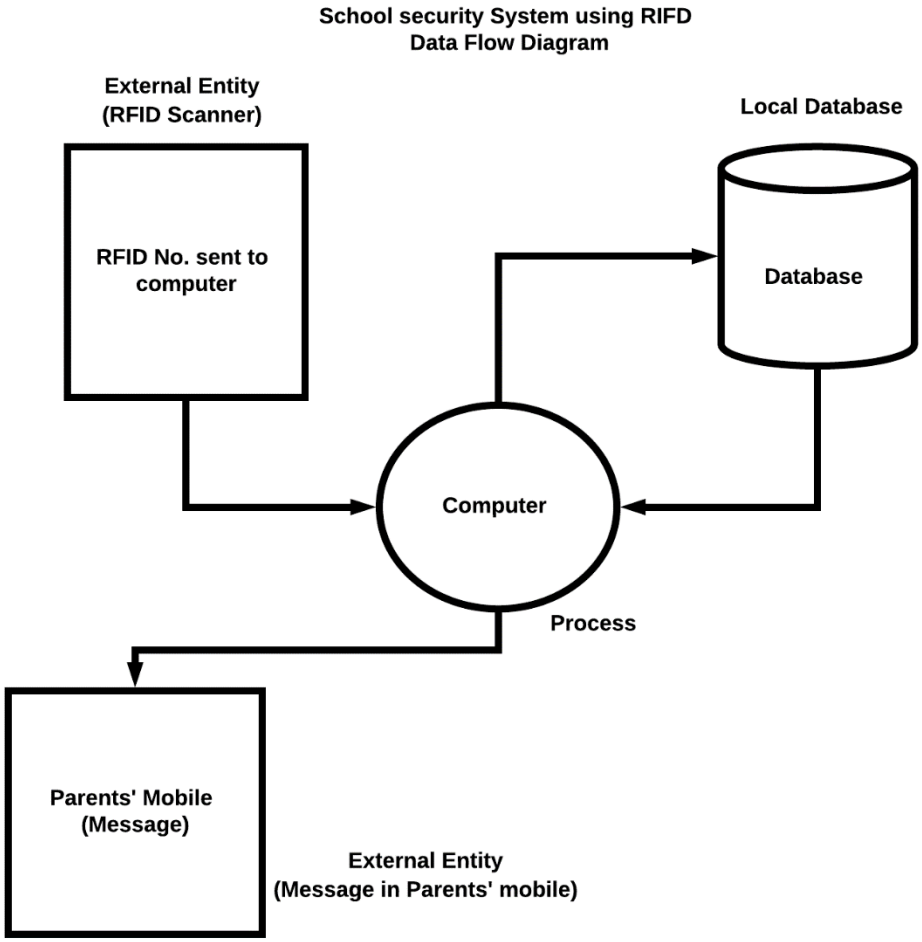
When a faculty member first log in to the Institute Interface he/she is asked if today is working day. If the user says yes, then it is recorded in working_days_log table.

4.4 ER-DIAGRAM

The ER-DIAGRAM of all the tables is given below:



4.5 DATA FLOW DIAGRAM



CHAPTER 5
CONCLUSION

In this project, we have implemented a concept of intelligent security system for school children with the help of RFID. Identification has become a necessary process in almost all fields. Now days, all works are done by automation. Automation invention all are concentrated on the main theme that is to reduce human effort and effective utilization of time. Our project is also implemented to reduce the man power in the identification. Due to this, accuracy of identification is maintained. School Security System makes the parent free from being reluctant from sending their child to school. They feel relief knowing that their child have reached school safely. School Security System can be used by schools and colleges to not only inform the parents that their child reached safely but also to keep record of the students' going out and coming in. Other methods of communicating can result to waste of time and money, and can be expensive and tiresome, therefore School Security Systems proves to be more efficient than the primitive methods. Surely, this identification system will enhance the needs of identification in other fields in our developing modern world. We, a team of four members took a step by step approach in order to reach our goal.

CHAPTER 6

APPENDIX

6.1 SOURCE CODE

NODE MCU AND RFID MODULE CODING

```
//rfid include
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN D4
#define RST_PIN D2
//wifi include
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
//wifi
const char* ssid     = "Baby";
const char* password = "baby1437";
ESP8266WebServer server(80);
WiFiClient client;
MDNSResponder mdns; //multicast Domain Name System
//rfid
MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class
String no_rfid ="";
void setup() {
   Serial.begin(115200);
  delay(1000);
 Serial.println("Hii  ");
  // Connect to WiFi network
  WiFi.begin(ssid, password);
  Serial.print("\n\r \n\rWorking to connect");
  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

```
  }
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  if (mdns.begin("esp8266", WiFi.localIP())) {
    Serial.println("MDNS responder started");
  }
   SPI.begin();       // Init SPI bus
   mfrc522.PCD_Init(); // Init MFRC522
   Serial.println("RFID reading UID");
}
void loop() {
if ( mfrc522.PICC_IsNewCardPresent())
    {
        if ( mfrc522.PICC_ReadCardSerial())
        {
           Serial.print("Tag UID:");
           for (byte i = 0; i < mfrc522.uid.size; i++) {
                 Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                //Serial.print(mfrc522.uid.uidByte[i], HEX);
                 no_rfid = mfrc522.uid.uidByte[i];
           }
                 Serial.print(no_rfid);
       server.handleClient();
if (client.connect("192.168.43.177", 80)) {
    client.print("GET /write_data.php?"); // This
    client.print("value=");  // This
    client.print(no_rfid);
    client.println(" HTTP/1.1"); // Part of the GET request
    client.println("Host: 127.0.0.1");
    client.println("Connection: close");
    client.println();  // Empty line
    client.println();  // Empty line
    client.stop();    // Closing connection to server
```

```
    }
    else {
    // If Arduino can't connect to the server (your computer or web page)
    Serial.println("--> connection failed\n");
  }
          Serial.println();
          mfrc522.PICC_HaltA();
       }
}
}
```

PHP Snippets:

1. **final_student_add**

```
$con=$_SESSION['con'];
$no=$_SESSION['add_no'];
for ($i=1; $i<=$no; $i++)
{
$rfid_no=$_POST['rfid_no'.$i];
$reg_no=$_POST['reg_no'.$i];
$inst_id=$_POST['inst_id'.$i];
$class=$_POST['class'.$i];
$sname=$_POST['sname'.$i];
$sec=$_POST['sec'.$i];
$f_m_name=$_POST['f_m_name'.$i];
$address=$_POST['address'.$i];
$connum=$_POST['connum'.$i];
$email=$_POST['email'.$i];
$qry="insert into student_master values('$inst_id', '$rfid_no', '$reg_no',
'$sname', '$class', '$sec', '$f_m_name', '$address', '$connum', '$email')";
$result=mysqli_query($con, $qry);
if($result)
{
 echo '<script language="javascript">';
  echo 'alert("The details were inserted to the Database successfully!! Return to
the Viewer/Editor Page");';
echo 'window.location.href = "http://localhost/wordpress/student-db-master/";';
```

```
  echo '</script>';
}   }
```

2. view_edit_student

```
$con=$_SESSION['con'];
if(isset ($_POST['but1']))
{
 $but1=$_POST['but1'];
 $regno=$_POST['regno'];
 if($but1=="update")
 {
 $rfid_no=$_POST['rfid'];
  $class=$_POST['class'];
 $name=$_POST['sname'];
 $section=$_POST['section'];
 $f_m_name=$_POST['fname'];
 $address=$_POST['address'];
 $con_no=$_POST['no'];
 $mail=$_POST['email'];
 $qry="update student_master set rfid='$rfid_no', name='$name', class='$class',
section='$section', f_m_name='$f_m_name', address='$address',
contact_no='$con_no', email_id='$mail'  where regno='$regno'";
 $result=mysqli_query($con,$qry);
 if($result)
 {
  echo '<script language="javascript">';
echo 'alert("Successfully Updated the Student Details")';
echo '</script>';
 }
  }
 else if($but1=="delete")
 {
   $qry="delete from student_master where regno='$regno'";
    $result=mysqli_query($con,$qry);
 if($result)
 {
   echo '<script language="javascript">'; echo 'alert("Successfully Deleted the
Student Details!! Going back to the Student Selection Page!!");';
```

```
echo 'window.location.href = "http://localhost/wordpress/student-db-master/";';
echo '</script>';
  }
 }
}
$regno=$_POST["regno"];
$qry="select * from student_master where regno='$regno'";
$result=mysqli_query($con,$qry);
  if($result)
  $row=$result->fetch_assoc();
  $rfid_no=$row['rfid'];
  $inst_id=$row['institute_id'];
  $class=$row['class'];
  $name=$row['name'];
  $section=$row['section'];
  $f_m_name=$row['f_m_name'];
  $address=$row['address'];
  $con_no=$row['contact_no'];
  $mail=$row['email_id'];
?>
  <form method="POST">
 <table>
<tr><td>Reg No:</td><td><input type="text" name="regno" readonly
value='<?php echo $regno; ?>'>
</td></tr>
<tr><td>RFID No:</td><td><input type="text" name="rfid" value='<?php
echo $rfid_no;?>'>
</td></tr>
<tr><td>Name:</td><td><input type="text" name="sname" value='<?php echo
$name; ?>' autofocus required></td></tr>
<tr><td>Class:</td><td><input type="text" name="class"  value='<?php echo
$class; ?>' required></td></tr>
<tr><td>Section:</td><td><input type="text" name="section" value='<?php
echo $section; ?>' required></td></tr>
<tr><td>Father/Mother Name:</td><td><input type="text" name="fname"
value='<?php echo $f_m_name; ?>' required></td></tr>
```

```
<tr><td>Contact No:</td><td><input type="tel" name="no" value='<?php echo
$con_no; ?>' required></td></tr>
<tr><td>Address:</td><td><input type="text"  name='address' value='<?php
echo $address; ?>' required></td></tr>
<tr><td>Email:</td><td><input type="email" name="email" value='<?php
echo $mail;  ?>' required></td></tr>
<tr>
<td coldspan="2"><center><button type="submit" value="update"
name="but1">Update</button>
<a href="http://localhost/wordpress/student-db-master/"><button type="button"
value="goback" name="but1">Go Back</button></a>
<button type="submit" value="delete"  name="but1">Delete</button>
  </td>
  </tr></center></table>
</form>
```

3. **working_days_log**

```
$con=$_SESSION['con'];
if(isset($_POST['but']))
{
  if($_POST['but']=='yes')
  {
  $inst_id=$_SESSION['inst_id'];
  $date=date('Y/m/d');
  $qry1="insert into working_days_log values('$inst_id', '$date')";
  $result=mysqli_query($con,$qry1);
  if($result)
  {
  echo '<script language="javascript">';
  echo 'window.location.href = "http://localhost/wordpress/student-database-
viewer-editor/";';
  echo '</script>';
  }
  else echo "not inserted";
  }
  else if ($_POST['but']=='no')
{
    echo '<script language="javascript">';
```

```
    echo 'window.location.href = "http://localhost/wordpress/student-database-viewer-editor/";';
    echo '</script>';
}
  }
?>
 <form method='post'>
  <center>
  <button value='yes' name='but' type="submit">Yes Mr.Computer</button>
  <button value='no' name='but' type="submit">No Mr.Computer</button>
  </center> </form>
```

4. display_all_students_of_a_school

```
$con=$_SESSION['con'];
$institute_id=$_SESSION['inst_id'];
$qry1="select * from student_master where institute_id='$institute_id'";
$result1=mysqli_query($con,$qry1);
echo "<center><h1 style='font-size:50px;'>".mysqli_affected_rows($con)."</h1></center>";
?>
<table>
        <tr><th>RFID No.</th><th>Regd. No.</th><th>Name</th><th>Class</th></tr>
<?php
while ($row=mysqli_fetch_array($result1))
{
echo"<tr><td>".$row['rfid']."</td><td>".$row['regno']."</td><td>".$row['name']."</td><td>".$row['class']."</td></tr>";
}
?>
</table>
```

5. User_access_2

```
$con=$_SESSION['con'];
        if(isset($_POST['reg']) && isset($_POST['select']))
{
        $un=$_POST["reg"];
        $_SESSION['reg']=$un;
        $sel1=$_POST["select"];
```

```
        $_SESSION['sel']=$sel1;
        $con=mysqli_connect('localhost','root','','worry_less');
}
$reg=$_SESSION['reg'];
$sel=$_SESSION['sel'];
        switch ($sel)
   {
                case 'Parent':
      if(isset($_POST['but']))
      {
      $but_value=$_POST['but'];
                        if($but_value=='update') {
       $stud_id=$_POST["stud_id"];
                                $pass=$_POST["password"];
$sql1="update login_parent  set password='$pass' where stud_id='$stud_id'";
                                $res1=mysqli_query($con,$sql1);
                                if($res1)
                                {
echo '<script language="javascript">';
echo 'alert("Successfully Updated the Parent Access Details!!")';
echo '</script>';
                                }
                                }
     else if($but_value=='delete')
      {
                                $stud_id=$_POST["stud_id"];
$sql1="delete from login_parent  where stud_id='$stud_id'";
                                $res1=mysqli_query($con,$sql1);
                                if($res1)
                                {
                                        echo '<script language="javascript">';
echo 'alert("Successfully Deleted  Parent Access Details!!")';
echo '</script>';
              echo '<script language="javascript">';
```

```
        echo 'window.location.href = "http://localhost/wordpress/user-access-database-viewer-editor/";';
        echo '</script>';
                                }
      }
      }
$sql="SELECT * FROM login_parent  WHERE stud_id='$reg'";
                        $result=mysqli_query($con,$sql);
                        $row=mysqli_fetch_assoc($result);
                        ?>
                        <form method="POST">
                        <table border="1">
                                <tr>
                                        <th>Student ID</th>
                                        <th>Password</th>
                                        <th colspan="2">Update/Delete</th>
                                </tr>
                                <tr>
<td><input type="text" name="stud_id" value="<?php echo $row['stud_id']; ?>" readonly></td>
<td><input type="text" name="password" value="<?php echo $row['password']; ?>"></td>
<td colspan="2"><button type="submit" name="but" value="update">Update</button>
<button type="submit" name="but" value="delete"  >Delete</button></td>
                                </tr>
                        </table>
                </form>
<?php
break;
      //case for staff
 case 'Staff':
                if(isset($_POST['but']))
      {
      $but_value=$_POST['but'];
```

```
                    if($but_value=='update') {
                        $staff_id=$_POST["staff_id"];
                        $pass=$_POST["pass"];
        $sql1="update login_teacher set pass='$pass' where staff_id='$staff_id'";
                        $res1=mysqli_query($con,$sql1);
                        if($res1)
                        {
                            echo '<script language="javascript">';
echo 'alert("Successfully Updated the Staff Access Details!!")';
echo '</script>';
                        }
                        }
      else if($but_value=='delete')
       {
                        $staff_id=$_POST["staff_id"];
$sql1="delete from login_teacher where staff_id='$staff_id'";
                        $res1=mysqli_query($con,$sql1);
                        if($res1)
                        {
                            echo '<script language="javascript">';
echo 'alert("Successfully Deleted Staff Access Details!!")';
echo '</script>';
              echo '<script language="javascript">';
        echo 'window.location.href = "http://localhost/wordpress/user-access-
database-viewer-editor/";';
        echo '</script>';
        }
      }
      }
        $sql="SELECT * FROM login_teacher WHERE staff_id='$reg'";
                    $result=mysqli_query($con,$sql);
                    $row=mysqli_fetch_assoc($result);
                    ?>
                    <form method="POST">
                    <table border="1">
```

```
                        <tr>
                                <th>Institute ID</th>
                                <th>Staff ID</th>
                                <th>Password</th>
                                <th colspan="2">Update/Delete</th>

                        </tr>
                        <tr>
<td><input type="text" name="inst_id" value="<?php echo $row['inst_id']; ?>"
readonly></td>
<td><input type="text" name="staff_id" value="<?php echo $row['staff_id'];
?>" readonly></td>
<td><input type="text" name="pass" value="<?php echo $row['pass'];
?>"></td>
<td colspan="2"><button type="submit" name="but"
value="update">Update</button><button type="submit" name="but"
value="delete">Delete</button></td>
                        </tr>
                </table>                </form>
                <?php
        break;
            //case for developers
      case 'Developer':
              if(isset($_POST['but']))
      {
      $but_value=$_POST['but'];
                if($but_value=='update') {
                        $username=$_POST["username"];
                        $password=$_POST["password"];
$sql1="update dev_login set password='$password' where
username='$username'";  $res1=mysqli_query($con,$sql1);
                        if($res1)
                        {
                                echo '<script language="javascript">';
echo 'alert("Successfully Updated the Developer Access Details!!")';
echo '</script>';
```

```
                                    }
                                        }
      else if($but_value=='delete')
      {   $username=$_POST["username"];
$sql1="delete from dev_login where username='$username'";
                                    $res1=mysqli_query($con,$sql1);
                                    if($res1)
                                    {
echo '<script language="javascript">';
echo 'alert("Successfully Deleted Developer Access Details!!")';
echo '</script>';
echo '<script language="javascript">'; echo 'window.location.href =
"http://localhost/wordpress/user-access-database-viewer-editor/";';
        echo '</script>';
                                    }
      }
      }
$sql="SELECT * FROM dev_login WHERE username='$reg' ";
                        $result=mysqli_query($con,$sql);
                        $row=mysqli_fetch_assoc($result);
                        ?>
                        <form method="POST">
                        <table>
                                <tr>
                                        <th>Developer Id</th>
                                        <th>Password</th>
                                        <th colspan="2">Update/Delete</th>

                                </tr>
                                <tr>

<td><input type="text" name="username" value="<?php echo
$row['username']; ?>" readonly></td>
<td><input type="text" name="password" value="<?php echo
$row['password']; ?>"></td>
```

```
<td colspan="2"><button type="submit" name="but"
value="update">Update</button><button type="submit" name="but"
value="delete">Delete</button></td>
                                    </tr>
                            </table>
                            </form>
                            <?php          break;  }
```

6. User_access_3_add_new_access_details

```
$con=$_SESSION['con'];
$sql='SELECT * FROM institute_master';
$result=mysqli_query($con,$sql);
?><h3>
<form method="post" action="http://localhost/wordpress/add-new-user-access-
details/">
        <table><tr>
                <td>
                Field:<br><br>
<select name="select">
<option  value="Parent">Parent</option>
<option value="Staff">Staff</option>
<option value="Developer">Developer</option>
</select><br><br>
Institute  Name:
*(Only  For Staff & Parent)<br><br>
<select name="inst_name">
<?php
        while  ($row=mysqli_fetch_array($result))
                                    {
                             ?>
        <option  value='<?php echo $row['name'];?>'>
<?php echo $row['name']; ?>
</option>
<?php }
?>
</select><br><br>
<button type="submit" name="" value="Search">Add</button>
</td></tr></table></form></h3>
```

7. User_access_4

```
?> <h3>
<?php
  $con=$_SESSION['con'];
if(isset($_POST['select']) && isset($_POST['inst_name']))
{
$_SESSION['sel']=$_POST["select"];
$_SESSION['inst_name']=$_POST["inst_name"];
}
$sel=$_SESSION['sel'];
$inst_name=$_SESSION['inst_name'];
$sql1="SELECT * FROM institute_master  WHERE name='$inst_name'";
        $result1=mysqli_query($con,$sql1);
        $row1=mysqli_fetch_assoc($result1);
        $inst_id=$row1['institute_id'];
//group of conditional statements
if(isset($_POST['but']))
{
        $but_value=$_POST['but'];
}
//end of conditional statements
switch ($sel)
        {
        //if adding new access details of a parent
case 'Parent':
         if($but_value=='add_parent')
        {
                $regd=$_POST['regd'];
                $parent_pass=$_POST['parent_pass'];
$qry1="insert into login_parent values('$inst_id', '$regd', '$parent_pass')";
        $result2=mysqli_query($con,$qry1);
                if($result2)
                {
                echo '<script language="javascript">';
                echo 'alert("Successfully Inserted New Parent Access Details")';
```

```
                    echo '</script>';
        echo '<script language="javascript">';
        echo 'window.location.href = "http://localhost/wordpress/user-access-
database-viewer-editor/";';
        echo '</script>';
                }}
  ?>
<form method="POST">
        <table>
                <tr>
                        <td>
                                Students Regd.No.:
                        </td>
                        <td><input type="text" name="regd"></td>
                </tr>
<tr><td>Password:</td><td><input type="Password" name="parent_pass">
</td></tr>
<tr><td colspan="2"><button type="submit" name='but'
value="add_parent">Add Details</button></td></tr>
        </table>
</form> <?php
break; case 'Staff':
          if($but_value=='add_staff')
        {
                $id=$_POST['staff_id'];
                $staff_pass=$_POST['staff_pass'];
        $qry2="insert into login_teacher values ('$inst_id', '$id', '$staff_pass')";
                $result3=mysqli_query($con,$qry2);
                if($result3)
                {
echo '<script language="javascript">';
echo 'alert("Successfully Inserted New Staff Access Details")';
        echo '</script>';
        echo '<script language="javascript">';
```

```
echo 'window.location.href = "http://localhost/wordpress/user-access-database-viewer-editor/";';
        echo '</script>';
                }
        }
         ?>
<form method="POST">
        <table>
                <tr>
                        <td>
                                Staff ID Number:
                        </td>
                        <td><input type="text" name="staff_id"></td>
                </tr>
<tr><td>Password:</td><td><input type="Password" name="staff_pass">
</td></tr><tr><td colspan="2"><button type="submit" name='but'
value="add_staff">Add Details</button></td></tr>
        </table>
</form>
<?php
break;
case 'Developer':
           if($but_value=='add_dev')
        {
                $username=$_POST['username'];
                $password=$_POST['dev_pass'];
                $qry3="insert into dev_login values ('$username', '$password')";
                $result4=mysqli_query($con,$qry3);
                if($result4)
                {
                echo '<script language="javascript">';
        echo 'alert("Successfully Inserted New Developer Access Details")';
                        echo '</script>';
        echo '<script language="javascript">';
```

```
        echo 'window.location.href = "http://localhost/wordpress/user-access-database-viewer-editor/";';
        echo '</script>';
                }
        }
         ?>
<form method="POST">
        <table>
                <tr>
                        <td>
                                Dev Username:
                        </td>
                        <td><input type="text" name="username"></td>
                </tr>
                <tr><td>Password:</td><td><input type="Password" name="dev_pass"></td></tr>
                <tr><td colspan="2"><button type="submit" name='but' value="add_dev">Add Details</button></td></tr>
        </table>
</form>
<?php
break;
}
?>
</h3>
```

8. **att_summary_1**

```
$con=$_SESSION['con'];
$reg=$_SESSION['uname'];
$qry1="select * from attendance_log where regno='$reg' order by date1 DESC";
$result1=mysqli_query($con,$qry1);
$row1=$result1->fetch_assoc();
//script for working days
$qry2="select * from inst_working_days where institute_id='".$row1['institute_id']."'";
$result2=mysqli_query($con,$qry2);
```

```
$row2=$result2->fetch_assoc();
$working_days=$row2['working_days'];
//script for days present
$qry3="select * from attendance_log where rfid='".$row1['rfid']."' AND
present_stat='present'";
$result3=mysqli_query($con,$qry3);
$present_count=mysqli_affected_rows($con);
//script for absent days
$absent_count=$working_days-$present_count;
//script for attendance percentage
$percentage=($present_count*100)/$working_days;
?>
 <div>
  <h2>
<table>
<tr>
<td>RFID Number:</td>
<td><b><u><lable><?php echo $row1['rfid']; ?></lable></u></b></td>
<td>Date:</td>
<td><b><u><lable><?php echo $row1['date1']; ?></lable></u></b></td>
</tr>
  <tr><td>Time:</td>
<td><b><u><lable><?php echo $row1['time1']; ?></lable></u></b></td>
<td>Message Status:</td>
<td><b><u><lable><?php echo $row1['msg_stat']; ?></lable></u></b></td>
</tr>
  <tr>
        <td>No. of Working Days</td>
        <td><b><u><lable><?php echo $working_days;
?></lable></u></b></td>
        <td>Days Present</td>
        <td><b><u><lable><?php echo $present_count;
?></lable></u></b></td>
        </tr><tr><td>Days Absent</td>
        <td><b><u><lable><?php echo $absent_count;
?></lable></u></b></td>
        <td>Attendance Percentage</td>
```

```
        <td><b><u><lable><?php echo $percentage;
?>%</lable></u></b></td>
</tr>
</table>
 </h2>
    </div>
```

9. spe_stud_report_gen_2

```
?>
 <h2>
        <form name="f1" method="POST">
<?php
        $con=$_SESSION['con'];
        $regno= $_SESSION["regno"];
        $date1=$_POST["date"];
        $sql="SELECT * FROM attendance_log WHERE date1='$date1' AND
regno='$regno'";
        $result=mysqli_query($con,$sql);
?>
        <table border="1">
                <tr>
                        <th>RFID Number:</th>
                        <th>Regd. No.:</th>
                        <th>Date:</th>
                        <th>Time:</th>
                        <th>Entry:</th>
                        <th>SMS Status:</th>
                </tr>
                <?php
                while($row=mysqli_fetch_array($result))
      {
                ?>
                <tr>
                        <td><?php echo $row['rfid']; ?></td>
                        <td><?php echo $row['regno']; ?></td>
                        <td><?php  echo $row['date1']; ?></td>
                        <td><?php echo $row['time1']; ?></td>
```

```
                    <td><?php echo $row['present_stat']; ?></td>
                    <td><?php echo $row['msg_status'];  ?></td>
                </tr>
            <?php
            }
    ?>
            </table>
    </form>
    </h2>
```

10. spe_stud_report_gen _3

```
    $con=$_SESSION['con'];
            $regno= $_SESSION["regno"];
            $fdate=$_POST["fdate"];
            $tdate=$_POST["tdate"];
            $sql="SELECT * FROM attendance_log WHERE date1>='$fdate' AND
    date1<='$tdate' AND regno='$regno' ";
            $result=mysqli_query($con,$sql);
    ?>
            <h2>
    <form name="f1" method="POST" ?>>
            <table border="1">
                <tr>
                    <td>RFID Number:</td>
                    <td>Regd. No.:</td>
                    <td>Date:</td>
                    <td>Time:</td>
                    <td>Entry:</td>
                    <td>SMS Status:</td>
                </tr>
                <?php while  ($row=mysqli_fetch_array($result))
                 {?>
                    <tr>
                    <td><?php echo $row['rfid']; ?></td>
                    <td><?php echo $row['regno']; ?></td>
                    <td><?php  echo $row['date1']; ?></td>
                    <td><?php echo $row['time1'];  ?></td>
                    <td><?php echo $row['present_stat']; ?></td>
```

```
                    <td><?php echo $row['msg_status'];  ?></td>
                </tr>
                <?php } ?>
            </table>
    </form>
    </h2>
```

11. spe_stud_report_gen _4

```
    ?>
    <h2>
            <form name="f1" method="POST">
    <?php
        $con=$_SESSION['con'];
        $regno= $_SESSION["regno"];
           $sql="SELECT * FROM attendance_log WHERE regno='$regno'";
        $result=mysqli_query($con,$sql);
          ?>
        <table border="1">
            <tr>
                <td>RFID Number:</td>
         <td>Regd. No.:</td>
         <td>Date:</td>
         <td>Time:</td>
         <td>Entry:</td>
         <td>SMS Status</td>
            </tr>
            <?php
                while ( $row=mysqli_fetch_assoc($result))
                {?>
                    <tr>
                        <td><?php echo $row['rfid']; ?></td>
                       <td><?php echo $row['regno']; ?></td>
                       <td><?php  echo $row['date1']; ?></td>
                       <td><?php echo $row['time1'];  ?></td>
                       <td><?php echo $row['present_stat']; ?></td>
                       <td><?php echo $row['msg_status'];  ?></td>
                    </tr>
              <?php }
```

```
        ?>
            </table>
    </form>
    </h2>
```

12.Atten_edit_2

```
    $con=$_SESSION['con'];
    if (isset($_POST["regno"])) {
        $regno=$_POST["regno"];
        $sql="SELECT * FROM student_master WHERE regno='$regno'";
        $result=mysqli_query($con,$sql);
        $row=mysqli_fetch_assoc($result);
        if(empty($row))
        {
    ?>          <script type="text/javascript">
    alert('Error: Student with that registration number not found!!!');
    location.href="http://localhost/wordpress/attendance-editor/";
                </script>
                <?php
        }
        else{
    ?>
    <form method="POST">
        <table>
            <tr>
    <td>Institute ID</td>
    <td><input type="text" name="inst_id" value="<?php echo $row['institute_id']
    ?>" readonly></td>
            </tr>
            <tr>
                <td>RFID No.</td>
    <td><input type="text" name="rfid" value="<?php echo $row['rfid'] ?>"
    readonly></td>
            </tr>
            <tr>
                <td>Regd. No.</td>
    <td><input type="text" name="reg" value="<?php echo $row['regno'] ?>"
    readonly></td>
```

```
		</tr>
		<tr>
			<td>Class</td>
			<td><input type="text" name="class" value="<?php echo $row['class'] ?>" readonly></td>
		</tr>
		<tr>
			<td>Date</td>
			<td><input type="Date" name="date"></td>
		</tr>
		<tr>
			<td>Time</td>
			<td><input type="Time" name="time"></td>
		</tr>
		<tr>
<td>Present Status<br>*to be filled only once for the entire day..</td>
<td><input type="text" name="present_stat"></td>
		</tr>
<tr>		<td colspan="2"><center><button type=submit>Add Into Attendance Log & Send SMS</button></center></td>
		</tr>
	</table>
</form>
	<?php
}
}
if (isset($_POST["date"]) && isset($_POST["time"]))
		{
			$inst_id=$_POST["inst_id"];
			$rfid=$_POST["rfid"];
			$reg=$_POST["reg"];
	$class=$_POST['class'];
			$date=$_POST["date"];
			$time=$_POST["time"];
			$present_stat=$_POST["present_stat"];
```

```
$qr="INSERT INTO attendance_log (institute_id, rfid, regno, class, date1, time1,
present_stat,msg_stat) VALUES('$inst_id','$rfid','$reg', '$class',
'$date','$time','$present_stat','')";
        $s=mysqli_query($con,$qr);
        if($s)
        {
            ?>
            <script type="text/javascript">
    alert('Success: Succesfully Added!!!');
    location.href="http://localhost/wordpress/attendance-editor/";
        </script>
        <?php
        }
        else{
            ?>
            <script type="text/javascript">
alert('Error:Unsuccessful!!!');
location.href="http://localhost/wordpress/attendance-editor/";
        </script>
        <?php
        }
        }
```

13. Atten_edit_3

```
if(isset($_POST['regno']))
{
$_SESSION['regno']=$_POST['regno'];
}
$regno=$_SESSION["regno"];
    $con=$_SESSION['con'];
    $sql="SELECT * FROM attendance_log WHERE regno='$regno'";
    $result=mysqli_query($con,$sql);
    ?>
    <form method="POST">
    <table>
        <tr>
            <td>Unique No.</td>
```

```
<td>Institute ID</td>
<td>RFID Number</td>
<td>Regd. Number</td>
<td>Class</td>
<td>Date</td>
<td>Time</td>
<td>Entry Status</td>
<td>SMS Status</td>
<td>Options</td>
</tr>
<?php

 while($row=mysqli_fetch_array($result))
{

?>
<tr>
<td><input type='text' value="<?php echo $row['slno'];?>" name='slno' readonly> </td>
<td><input type="text" value="<?php echo $row['institute_id'] ?>" readonly></td>
<td><input type="text" value="<?php echo $row['rfid'] ?>" readonly></td>
<td><input type="text" value="<?php echo $row['regno'] ?>" readonly></td>
<td><input type="text" value="<?php echo $row['class']?>" readonly></td>
<td><input type="Date" name='date'value="<?php echo $row['date1']?>"></td>
<td><input type="time" name='time' value="<?php echo $row['time1']?>"></td>
<td><input type="text" name='present_stat' value="<?php echo $row['present_stat']?>"></td>
<td><input type="text" name='msg_stat' value="<?php echo $row['msg_stat']?>"></td>
<td><button name="but" value="edit" type="submit">Save Changes</button></td>
<td><button name="but" value="delete" type="submit">Delete Record</button></td>
</tr>
```

```
<?php
}
?></table>
</form><?php
if (isset($_POST["but"]))
                {
                $but=$_POST["but"];
                $slno=$_POST['slno'];
                $date=$_POST["date"];
                $time=$_POST["time"];
                $present_stat=$_POST["present_stat"];
                $msg_stat=$_POST['msg_stat'];
                switch ($but)
                {
                        case 'edit':
$qr="UPDATE attendance_log SET date1='$date', time1='$time',
present_stat='$present_stat', msg_stat='$msg_stat' WHERE slno='$slno'";
                                break;
                case 'delete':
$qr="DELETE FROM attendance_log WHERE slno='$slno'";
break;
                }
                $s=mysqli_query($con,$qr);
if($s)
{
?>
                <script type="text/javascript">
alert('Success: attendance log Edited!!');
location.href="http://localhost/wordpress/edit-attendance-log/";
                </script>
                <?php
                }
                else
?>
                <script type="text/javascript">
```

```
                    alert('Error: Failed to edit!!');
                    location.href="http://localhost/wordpress/attendance-
editor/";
            </script>
            <?php
}
```

14. inst_ag_5

```
if(isset($_POST['class']) && isset($_POST['date']))
  {
  $_SESSION['date']=$_POST["date"];
        $_SESSION['class']=$_POST["class"];
}
$class=$_SESSION['class'];
$date=$_SESSION['date'];

        $con=$_SESSION['con'];
//counting the total number of students in a class
  $qry1="select * from student_master where class='$class'";
  $res1=mysqli_query($con,$qry1);
  $count1=mysqli_affected_rows($con);
//Number of students present
  $qry2="select * from attendance_log where class='$class' AND
present_stat='present' AND date1='$date'";
  $res2=mysqli_query($con, $qry2);
  $row2=$res2->fetch_assoc();
  $count2=mysqli_num_rows($res2);
  $stud_ab=$count1-$count2;
  //querying the attendance log
    $sql="SELECT * FROM attendance_log WHERE class='$class' AND
date1='$date'";
    $result=mysqli_query($con,$sql);
      ?>
      <h1>
        <center>
```

```
    <lable>Total Number of Students: <u><b><?php echo $count1; ?></b></u></lable></br>
    <lable>Number of Students Present: <u><b><?php echo $count2; ?></b></u></lable></br>
    <lable>Number of Students Absent: <u><b><?php echo $stud_ab; ?></b></u></lable></br>
      </center>
   </h1>
    <h2>
    <table>
       <tr>
          <th>Institute ID</th>
           <th>RFID Number</th>
           <th>REGD. No.</th>
           <th>Date</th>
           <th>Time</th>
           <th>Present Status</th>
           <th>SMS Status</th>
       </tr>
       <?php
           while ( $row=mysqli_fetch_assoc($result))
           { ?>

              <tr>
              <td> <?php echo $row['institute_id']; ?></td>
                <td> <?php echo $row['rfid']; ?></td>
                <td> <?php echo $row['regno']; ?></td>
                <td> <?php echo $row['date1']; ?></td>
                <td> <?php echo $row['time1']; ?></td>
                <td> <?php echo $row['present_stat'];?></td>
                <td> <?php echo $row['msg_stat']; ?></td>
              </tr>
         <?php }  ?>

   </table>
```

```
</h2>
```

15.inst_ag_6

```
$time='09:30:00';
if(isset($_POST['class']) && isset($_POST['date']))
  {
  $_SESSION['date']=$_POST["date"];
  $_SESSION['class']=$_POST["class"];
}
$date=$_SESSION['date'];
$class=$_SESSION['class'];
  $con=$_SESSION['con'];
 $sql="SELECT * FROM attendance_log WHERE class='$class' AND
date1='$date' AND time1 > '09:30:00' AND present_stat='present'";
    $result=mysqli_query($con,$sql);
    $count=mysqli_affected_rows($con);
      ?>
     <h1><lable>Number of Students Late: <u><?php echo $count;
?></u></lable></h1>
      <h2>
      <table>
         <tr>
            <th>Institute ID</th>
             <th>RFID Number</th>
             <th>REGD. No.</th>
             <th>Date</th>
             <th>Time</th>
             <th>Present Status</th>
             <th>SMS Status</th>
         </tr>
         <?php
             while ( $row=mysqli_fetch_assoc($result))
             { ?>

                 <tr>
                 <td> <?php echo $row['institute_id']; ?></td>
```

```
                <td> <?php echo $row['rfid']; ?></td>
                <td> <?php echo $row['regno']; ?></td>
                <td> <?php echo $row['date1']; ?></td>
                <td> <?php echo $row['time1']; ?></td>
                <td> <?php echo $row['present_stat'];?></td>
                <td> <?php echo $row['msg_stat']; ?></td>
              </tr>
          <?php }  ?>

    </table>
</h2>
```

16. inst_ag_7

```
if(isset($_POST['min_time']) && isset($_POST['max_time']) && isset($_POST['date'])  )
  {
  $_SESSION['min']=$_POST["min_time"];
  $_SESSION['max']=$_POST["max_time"];
  $_SESSION['date']=$_POST['date'];
}
$min=$_SESSION['min'];
$max=$_SESSION['max'];
$date=$_SESSION['date'];
  $con=$_SESSION['con'];
   $sql="SELECT * FROM attendance_log WHERE date1='$date' AND time1>='$min'  AND time1<='$max'";
    $result=mysqli_query($con,$sql);
    $count=mysqli_affected_rows($con);
     ?>
     <h2>
     <table>
        <tr>
          <th>Institute ID</th>
           <th>RFID Number</th>
           <th>REGD. No.</th>
           <th>Date</th>
```

```
            <th>Time</th>
            <th>Present Status</th>
            <th>SMS Status</th>
        </tr>
        <?php
            while ( $row=mysqli_fetch_assoc($result))
            { ?>

                <tr>
                <td> <?php echo $row['institute_id']; ?></td>
                  <td> <?php echo $row['rfid']; ?></td>
                  <td> <?php echo $row['regno']; ?></td>
                  <td> <?php echo $row['date1']; ?></td>
                  <td> <?php echo $row['time1']; ?></td>
                  <td> <?php echo $row['present_stat'];?></td>
                  <td> <?php echo $row['msg_stat']; ?></td>
                </tr>
          <?php }  ?>

    </table>
</h2>
```

17. sms_1

```
?>
<form method="POST">
        <h2>
                <table>
                        <tr>
                                <td>Enter Regd. No.</td>
                                <td><input type="text" name="regd"></td>
                                <td>OR</td>
                                <td>Enter RFID No.</td>
                                <td><input type="text" name="rfid"></td>
                                <td>OR</td>
                                <td>Enter Contact No.</td>
                                <td><input type="text" name="contact"></td>
```

```
                    </tr>
                    <tr>
        <td colspan="4">Enter the Message Content</td>
        <td colspan="4"><input type="text" name="msg"></td>
                    </tr>
                    <tr>
<td colspan="8"><center><button type="submit">Send SMS</button>
</center></td>
</tr>
</table>
</h2>
</form>
  <?php
$con=$_SESSION['con'];
if(isset($_POST['regd']) && isset($_POST['msg']))
{
  $regd=$_POST['regd'];
  $msg=$_POST['msg'];
        $qry1="select contact_no from student_master where regno='$regd'";
        $res1=mysqli_query($con,$qry1);
        $row1=$res1->fetch_assoc();
        $phone_number=$row1['contact_no'];
  if(!empty($phone_number))
  {
        send_sms($phone_number, $msg);
  }
  $phone_number="";
  $msg="";
}
if(isset($_POST['rfid']) && isset($_POST['msg']))
{
  $rfid=$_POST['rfid'];
  $msg=$_POST['msg'];
        $qry1="select contact_no from student_master where rfid='$rfid'";
        $res1=mysqli_query($con,$qry1);
```

```
        $row1=$res1->fetch_assoc();
        $phone_number=$row1['contact_no'];
          if(!empty($phone_number))
  {
        send_sms($phone_number, $msg);
  }
  $phone_number="";
  $msg="";
}
if(isset($_POST['contact']) && isset($_POST['msg']))
{
  $contact=$_POST['contact'];
  $msg=$_POST['msg'];
    if(!empty($contact))
  {
        send_sms($contact, $msg);
  }
  $contact="";
  $msg="";
}
function send_sms($phone_number, $msg)
{
        echo "Phone Number:::".$phone_number."<br>";
        echo "Message Content::".$msg."<br>";
// Account details
   $apiKey=urlencode('omVQqMD9jUw-WDfGwL1uHDkEtQDhAAhsbMyL
UU9BPe');
     // Message details
   $sender = urlencode('TXTLCL');
   $message = rawurlencode("".$msg);
   $data = array('apikey'  => $apiKey, 'numbers' => $phone_number, "sender" =>
$sender, "message" => $message);
     // Send the POST request with cURL
   $ch = curl_init('https://api.textlocal.in/send/');
   curl_setopt($ch, CURLOPT_POST, true);
```

```
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
        // Process your response here
    echo "displaying:". $response;
}
```

18.Sendmail_1

```
<center>
    <div class="form-group container-fluid" style="width:500px;">
<form method="post" enctype="multipart/form-data">
<h3>
Enter your E-mail ID<br>
<input type="text" name="email" class="form-control" placeholder="your_mail_id@example.com" required><br>
Subject<input type="text" name="sub" class="form-control" placeholder="Doubth!!.. etc.." required><br>
Message Content<br>
<textarea type="textarea" class="form-control" name="msg" placeholder="Enter your message" cols="30" rows="4" required></textarea><br>
<button type="submit" name="send" class="form-control btn-info">Send E-Mail</button>
</h3>
</form>
</div>
</center>
<?php
if(isset($_POST['send']))
{
        $to="worrylessrfid@gmail.com";
        $email=$_POST['email'];
        $sub=$_POST['sub'];
        $msg=$_POST['msg'];
        $full_msg="Sender: ".$email." Message Content: ".$msg;
        require 'PHPMailerAutoload.php';
```

```
        require 'credential.php';
$mail = new PHPMailer(true);
try {
output
    $mail->isSMTP();                                    // Set mailer to use SMTP
    $mail->Host       = 'smtp.gmail.com';  // Specify main and backup SMTP servers
    $mail->SMTPAuth   = true;                           // Enable SMTP authentication
    $mail->Username   = EMAIL;                  // SMTP username
    $mail->Password   = PASS;                           // SMTP password
    $mail->SMTPSecure = 'tls';                          // Enable TLS encryption,
`ssl` also accepted
    $mail->Port       = 587;                            // TCP port to connect to
    $mail->setFrom(EMAIL, 'Worry Less');
    $mail->addAddress($to);     // Add a recipient
    $mail->isHTML(false);                               // Set email format to HTML
    $mail->Subject = $sub;
    $mail->Body    = $full_msg;
    $mail->send();
    echo '<center><label class=alert-success>The EMAIL was sent
successfully!!!</label></center>';
} catch (Exception $e) {
    echo "<center><label class=alert-danger>The EMAIL wasn't sent!!!. Mailer
Error: {$mail->ErrorInfo}</label></center>";
}
}
session_start();
if(!empty($_SESSION['url']))
{
$url=$_SESSION['url'];
}
?>
<center>
<div style="width:300px">
<form action=<?php echo $url; ?> ><button class="form-control alert-
danger">Go Back To Where You Left!!!!</button>
```

```
</form>
</div>
</center>
<footer style="text-align:center">
<p style="padding-bottom:10px;">Developed By: Rahulkumar Singh, A.Raja Mathiyazhagan, S.Yogesh, H.Mohamed Rizwan</p>
</footer>
</body>
</html>
```

19. get_data

```
<?php
$dbname="worry_less";
$host="localhost";
$user="root";
$pwd="";
$con=mysqli_connect($host,$user,$pwd,$dbname);
$url=$_SERVER['REQUEST_URI'];
header("Refresh: 2; URL=$url");  // Refresh the webpage every 5 seconds
$qry1="select * from attendance_log where msg_stat='' limit 1";
$res1=mysqli_query($con,$qry1);
$row1=mysqli_fetch_assoc($res1);
$rfid_No=$row1['rfid'];
$d_arr=$row1['date1'];
$t_arr=$row1['time1'];
$qry2="select * from student_master where rfid='$rfid_No'";
$res2=mysqli_query($con,$qry2);
$row2=mysqli_fetch_assoc($res2);
$phone_number=$row2['contact_no'];
echo "number:".$phone_number;
$name=$row2['name'];
$class=$row2['class'];
if(!empty($phone_number))
{
$apiKey = urlencode('omVQqMD9jUw-WDfGwL1uHDkEtQDhAAhsbMyLUU9BPe');
```

```
    $sender = urlencode('TXTLCL');
  $message = rawurlencode('Your child, '.$name.', studying in class '.$class.' have reached the school premises. Arrival date:'.$d_arr.' Arrival time:'.$t_arr);
    $data = array('apikey'  => $apiKey, 'numbers' => $phone_number, "sender" => $sender, "message" => $message);
    $ch = curl_init('https://api.textlocal.in/send/');
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    echo $response;
$phone_number=NULL;
$qry3="update attendance_log set msg_stat='sent' where rfid='$rfid_No' && date1='$d_arr' && time1='$t_arr'";
$res=mysqli_query($con,$qry3);
}
?>
<html>
<head>
    <title>RFID</title>
</head>
    <body>
        <h1>RFID Numbers</h1>
    <table border="0" cellspacing="0" cellpadding="4">
      <tr>
            <td>RFID NO.</td>
             <td>Message Status</td>
      </tr>
<?php
    $result = mysqli_query($con,'SELECT * FROM attendance_log');
    while($row = mysqli_fetch_array($result))
    {
        echo "<tr>";
        echo "<td>" . $row['rfid'] . "</td>";
```

```
        echo "<td>" . $row['msg_stat']. "</td>";
        echo "</tr>";
    }
    mysqli_close($con);
?>
    </table>
    </body>
</html>
```

20.write_data

```
<?php
    $dbusername = "root"; // enter database username, I used "arduino" in step 2.2
    $dbpassword = ""; // enter database password, I used "arduinotest" in step 2.2
    try{
$dbname="worry_less";
$host="localhost";
$user="root";
$pwd="";
$con=mysqli_connect($host,$user,$pwd,$dbname);
        if(!isset( $_GET['value'])){
            throw new Exception('value not found');
        }
        $value = $_GET["value"];
        $sql = "select * from student_master where rfid='$value'";
     $result=mysqli_query($con,$sql);
$row=mysqli_fetch_assoc($result);
$inst_id=$row['institute_id'];
$regno=$row['regno'];
$class=$row['class'];
date_default_timezone_set("Asia/Kolkata");
$d1=date('Y/m/d');
$t1=date('H:i:s');
$p="present";
$q1="select * from attendance_log where rfid='$value' AND date1='$d1'";
$r1=mysqli_query($con,$q1);
$count=mysql_num_rows($r1);
```

```
$row1=$r1->fetch_assoc();
if(empty($row1))
{
  $sql2="insert into attendance_log (institute_id, rfid, regno, class, date1, time1, present_stat) values ('$inst_id','$value','$regno','$class', '$d1', '$t1', '$p')";
$res2=mysqli_query($con,$sql2);
}
else
{   $sql1="insert into attendance_log (institute_id, rfid, regno, class, date1, time1) values ('$inst_id','$value','$regno', '$class', '$d1', '$t1')";
$res=mysqli_query($con,$sql1);
}    }
    catch (Exception $e){
        echo $e->getMessage();
    }
?>
```

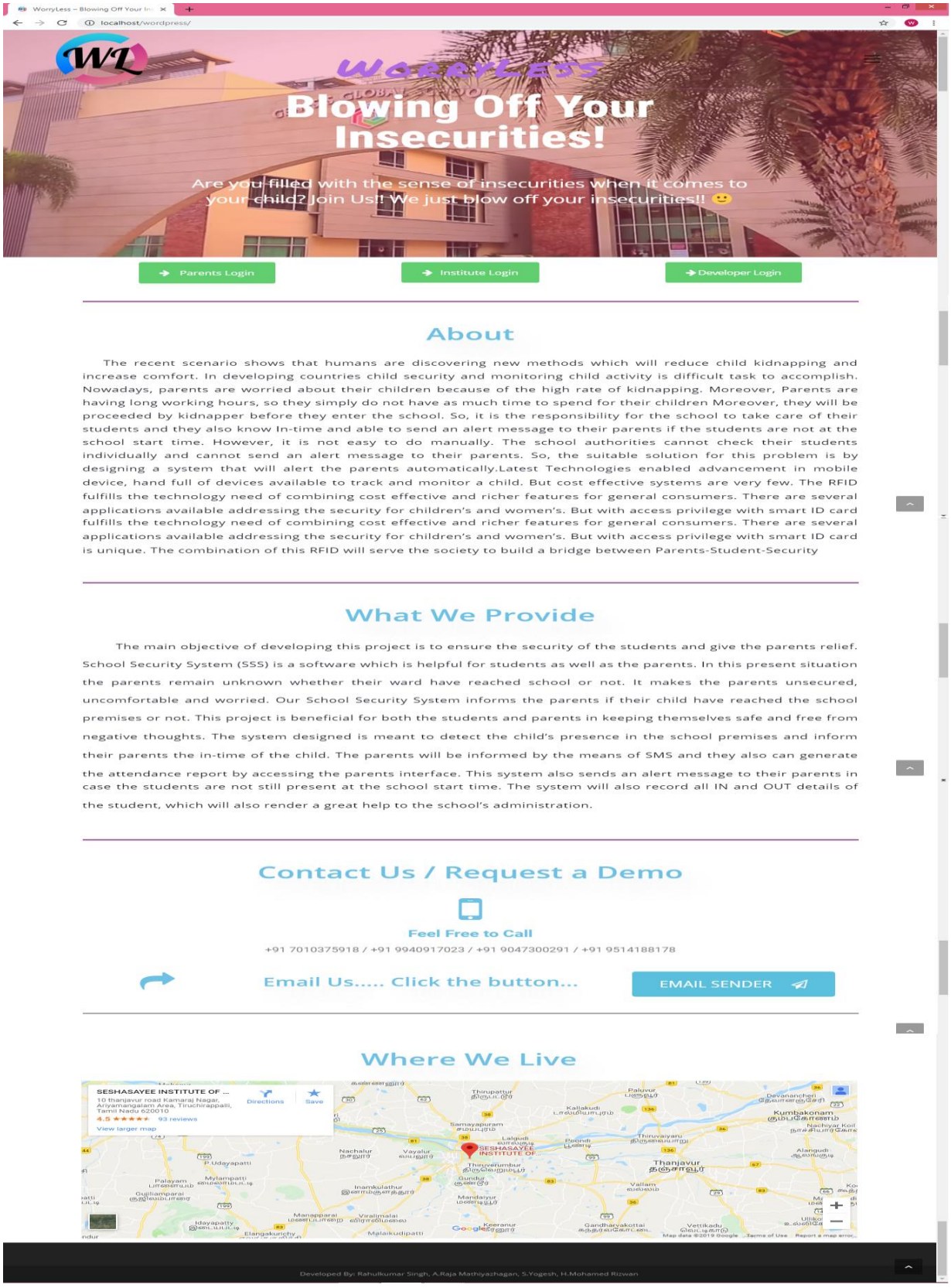
21. send_sms

```
<?php
    $dbusername = "root"; // enter database username, I used "arduino" in step 2.2
    $dbpassword = ""; // enter database password, I used "arduinotest" in step 2.2
    try{
$dbname="worry_less";
$host="localhost";
$user="root";
$pwd="";
$con=mysqli_connect($host,$user,$pwd,$dbname);
        if(!isset( $_GET['value'])){
            throw new Exception('value not found');
        }
        $value = $_GET["value"];
        $sql = "select * from student_master where rfid='$value'";
     $result=mysqli_query($con,$sql);
$row=mysqli_fetch_assoc($result);
$inst_id=$row['institute_id'];
$regno=$row['regno'];
```

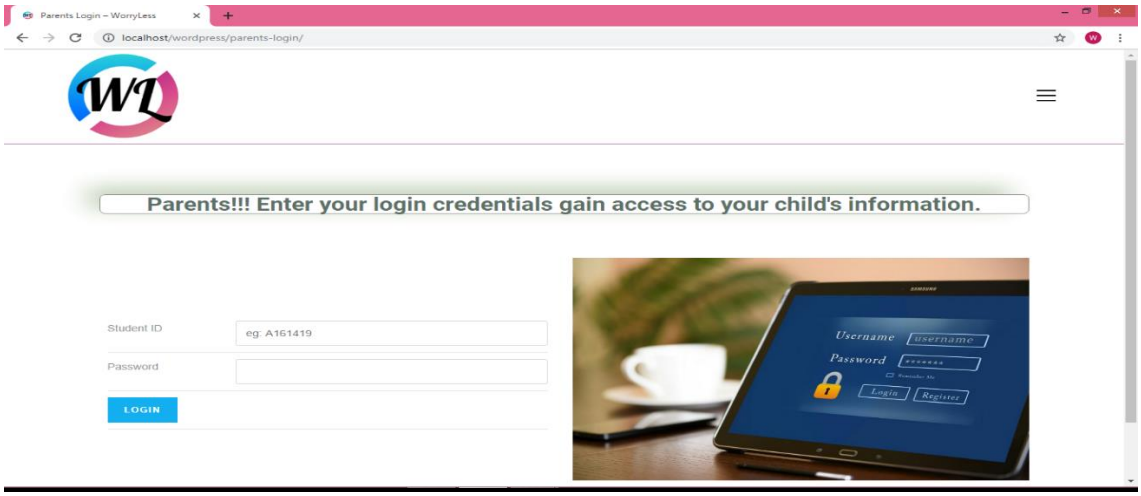
```
$class=$row['class'];
date_default_timezone_set("Asia/Kolkata");
$d1=date('Y/m/d');
$t1=date('H:i:s');
$p="present";
$q1="select * from attendance_log where rfid='$value' AND date1='$d1'";
$r1=mysqli_query($con,$q1);
$count=mysql_num_rows($r1);
$row1=$r1->fetch_assoc();
if(empty($row1))
{  $sql2="insert into attendance_log (institute_id, rfid, regno, class, date1, time1, present_stat) values ('$inst_id','$value','$regno','$class', '$d1', '$t1', '$p')";
$res2=mysqli_query($con,$sql2);
}
else
{
   $sql1="insert into attendance_log (institute_id, rfid, regno, class, date1, time1) values ('$inst_id','$value','$regno', '$class', '$d1', '$t1')";
$res=mysqli_query($con,$sql1);
}    }
   catch (Exception $e){
      echo $e->getMessage();
   }
?>
```

6.2 SCREENSHOTS

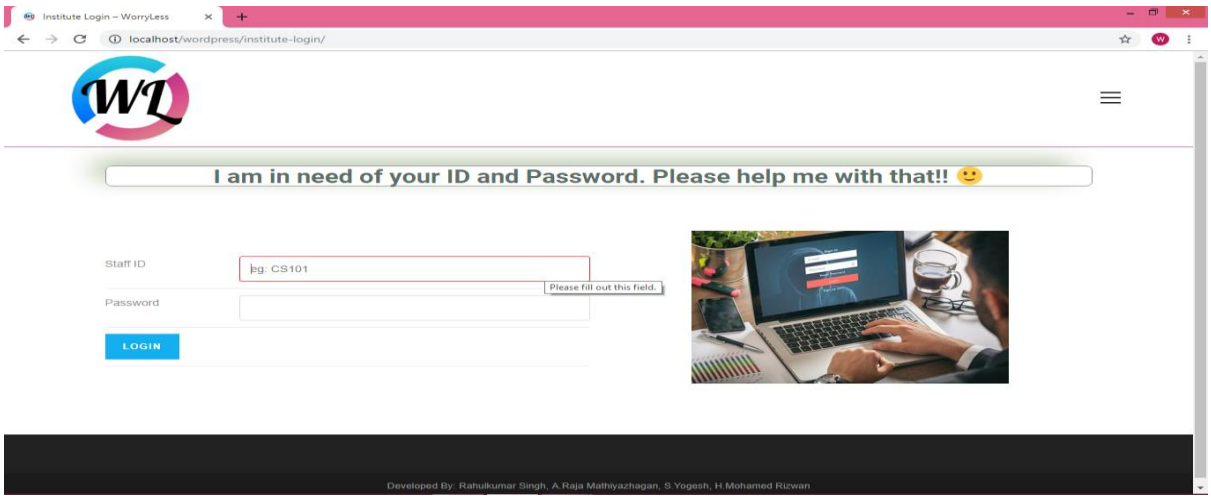
HOME



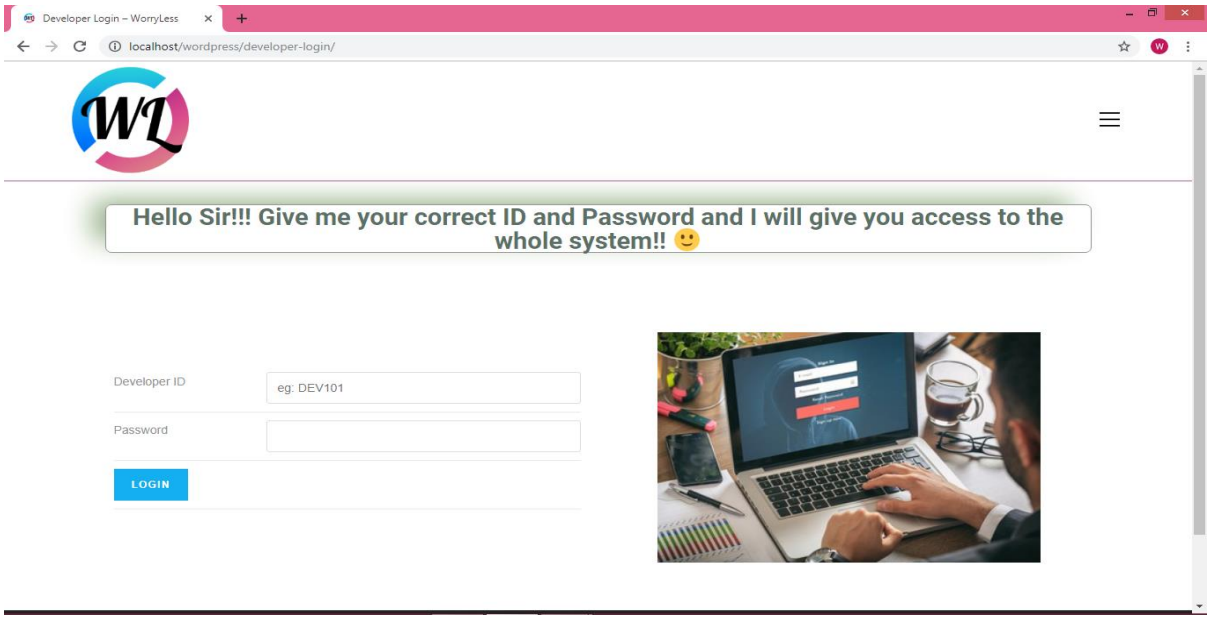
PARENT LOGIN



INSTITUTE LOGIN

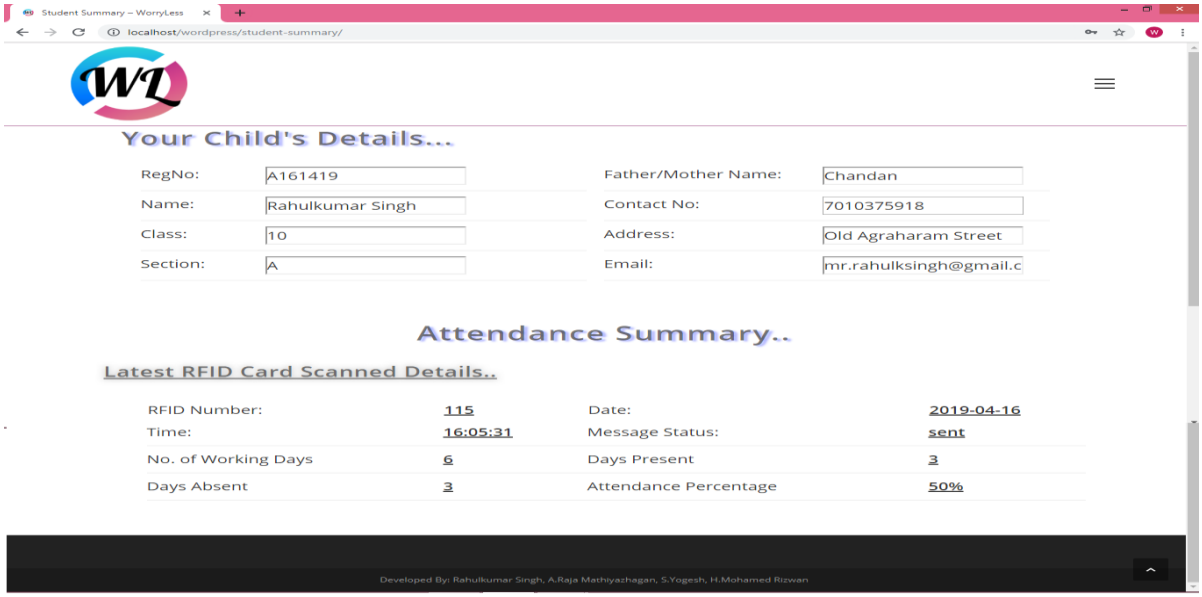


DEVELOPER LOGIN

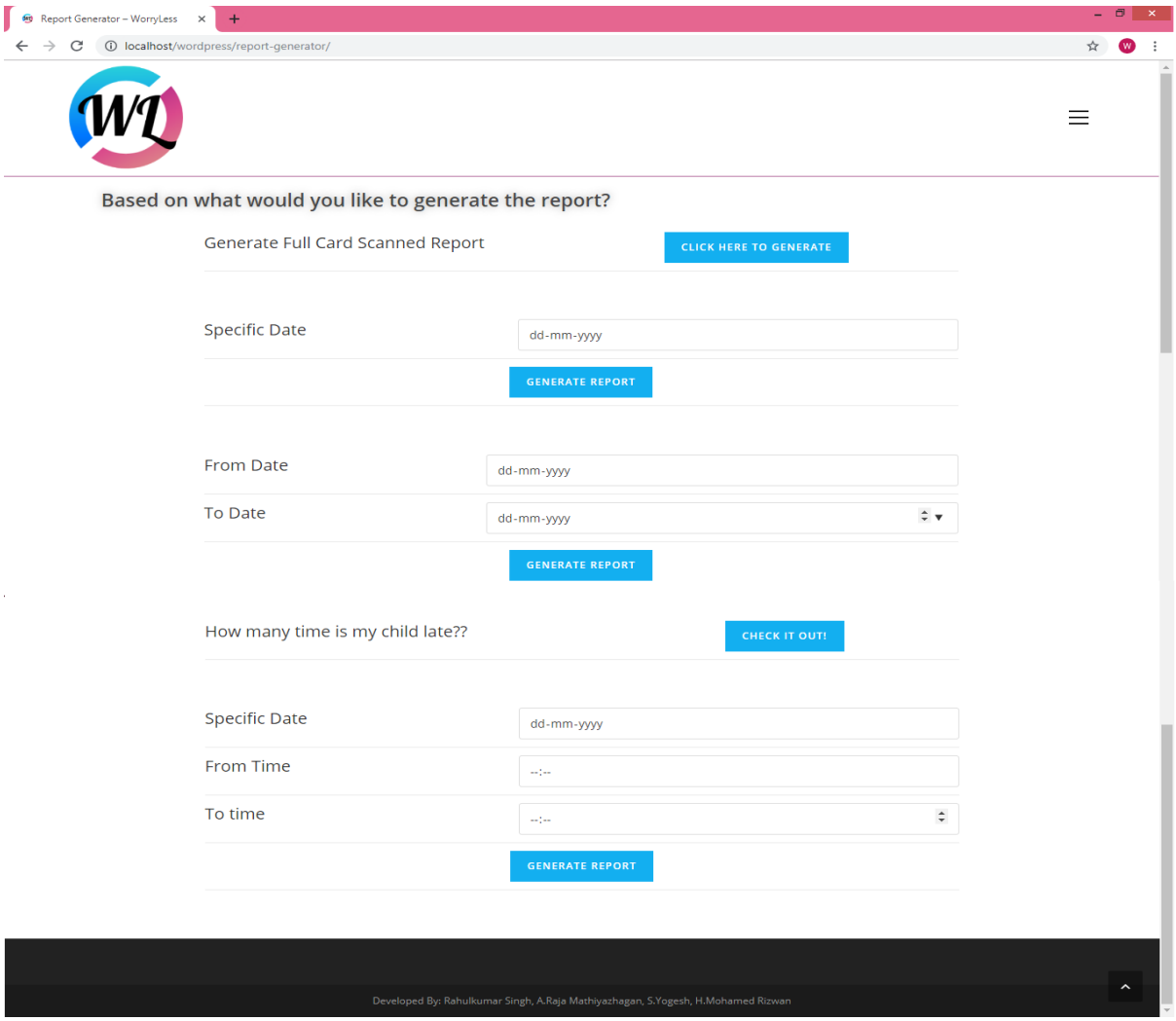


PARENTS' INTERFACE

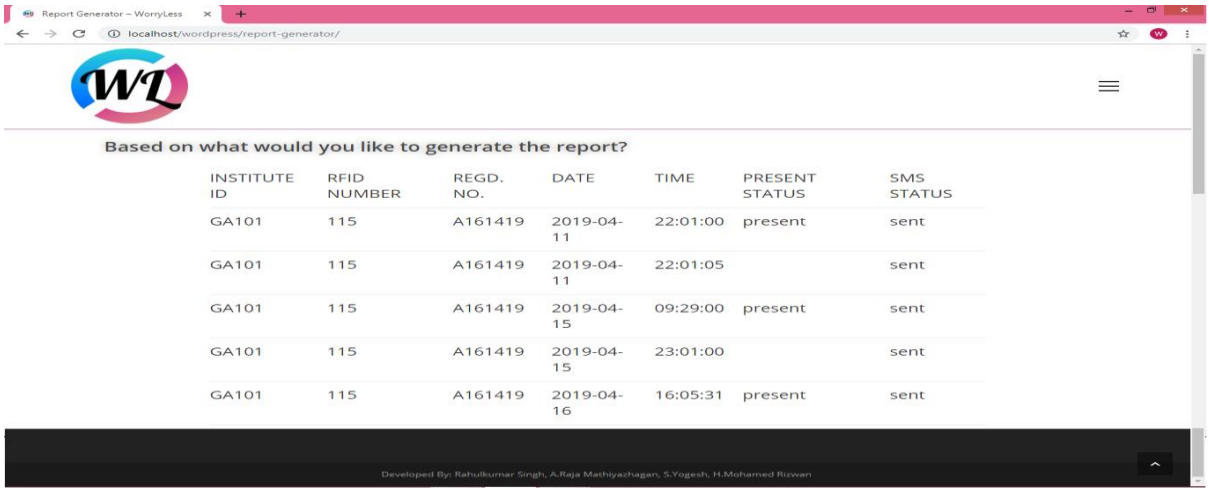
STUDENT SUMMARY



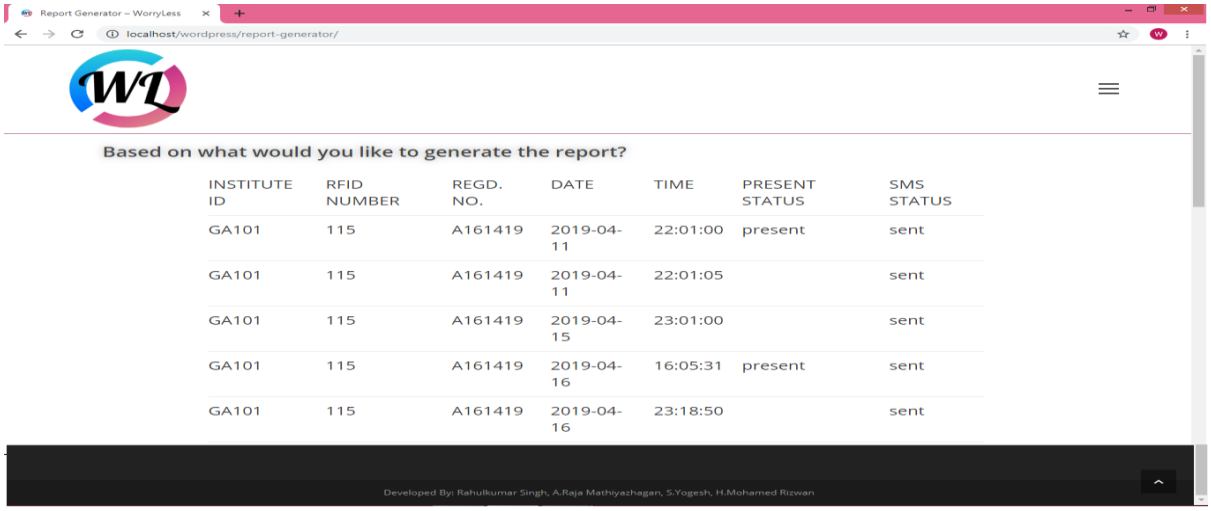
REPORT GENERATOR



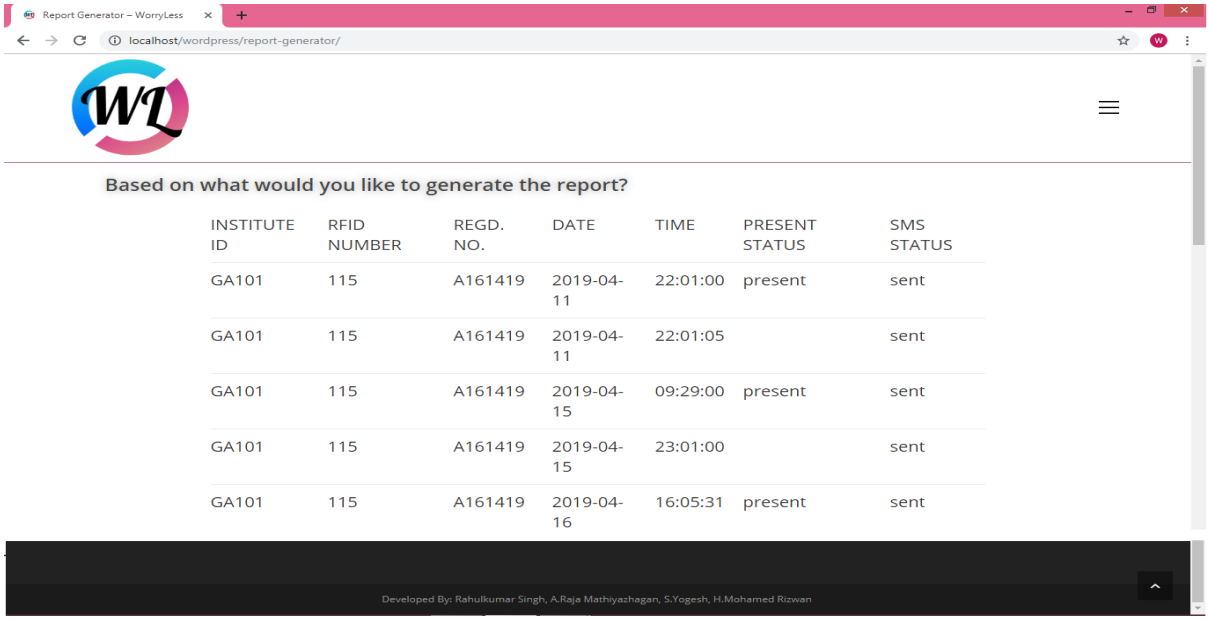
FULL REPORT GENERATOR



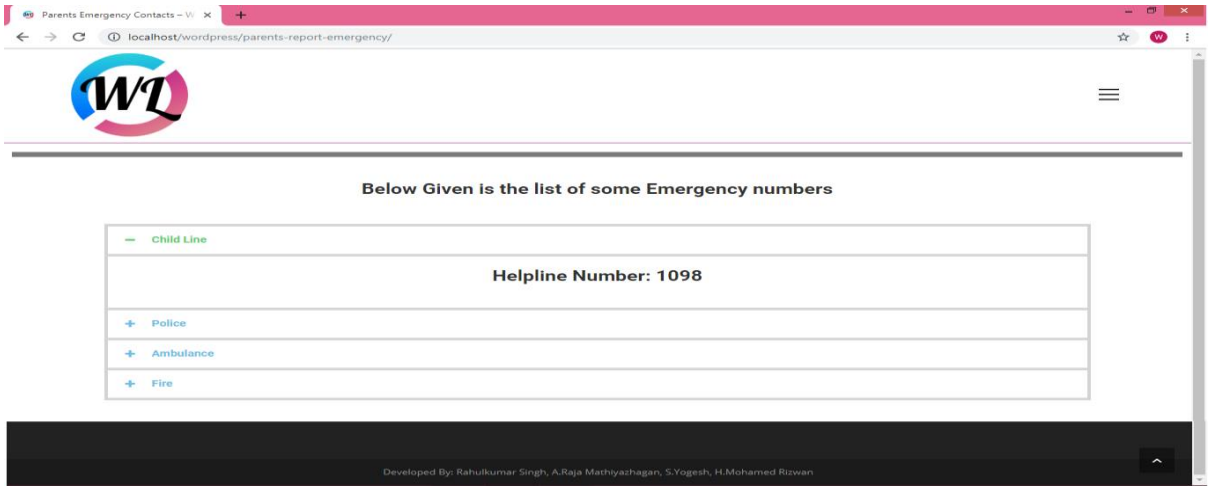
REPORT GENERATOR BASED OF LATE TIMINGS



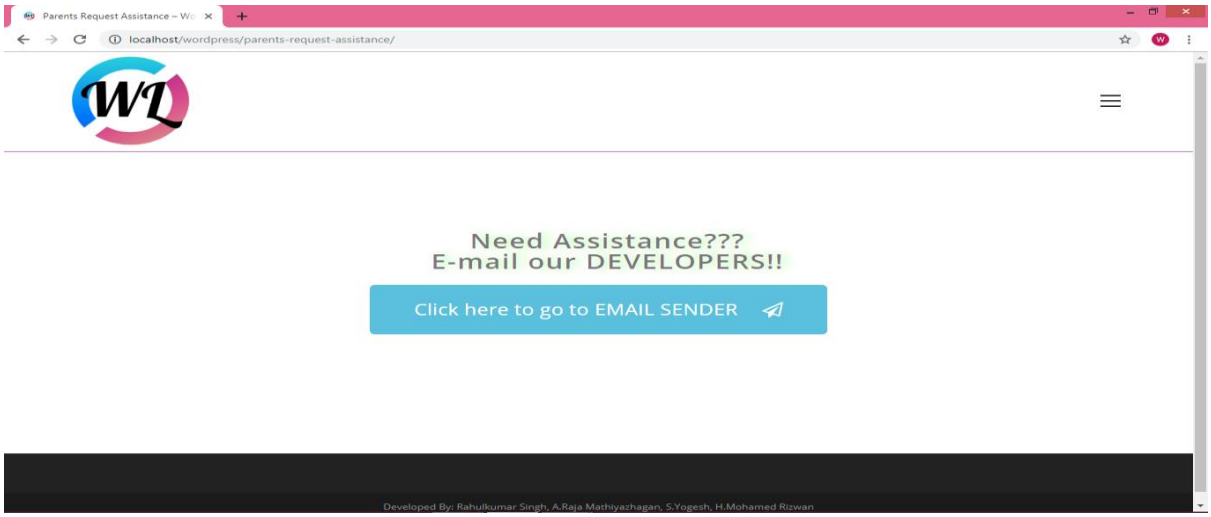
REPORT GENERATOR BASED ON RANGE OF DATES



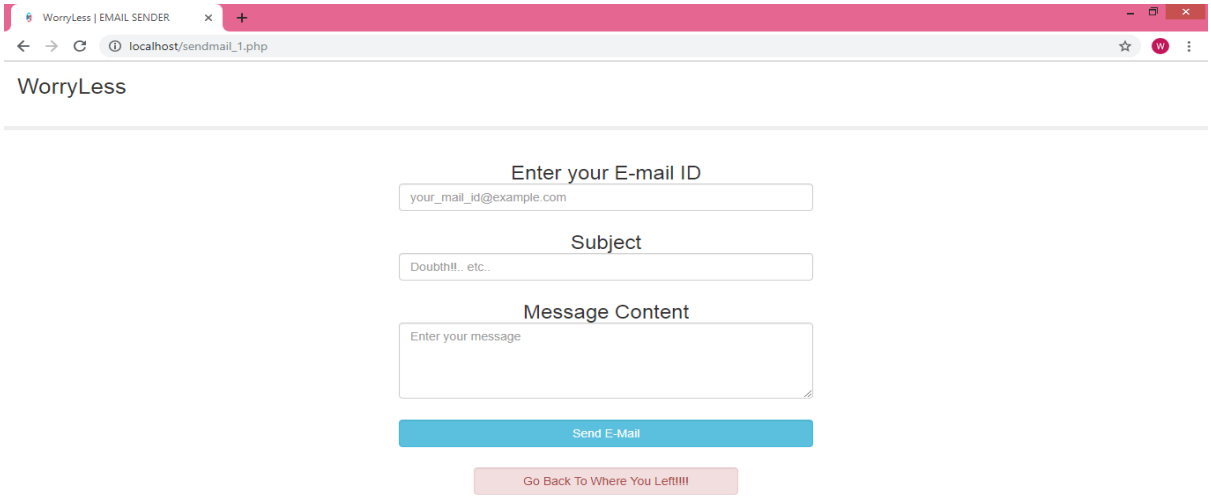
EMERGENCY CONTACTS



REQUEST ASSISTANCE

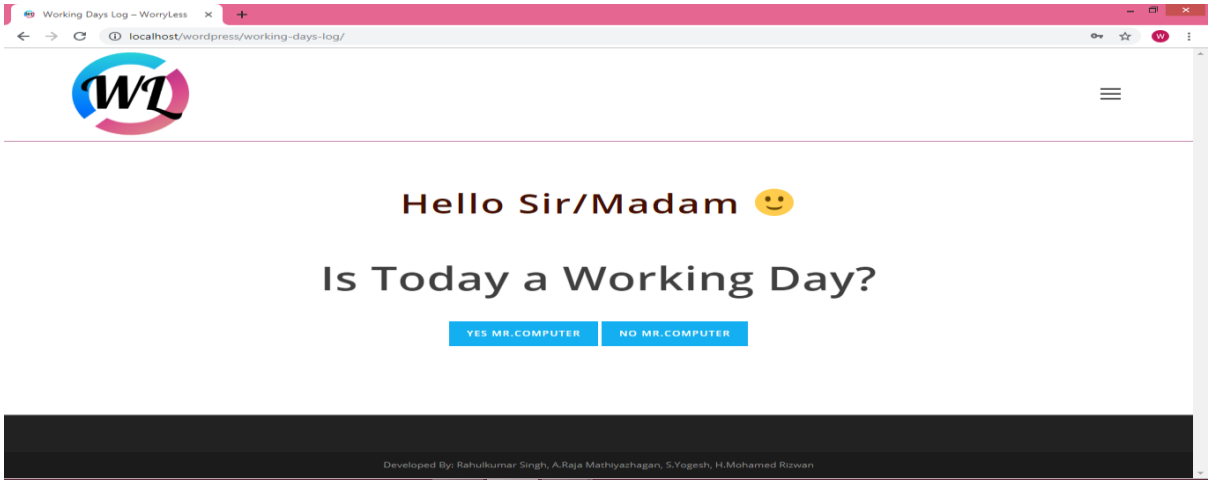


EMAIL SENDER

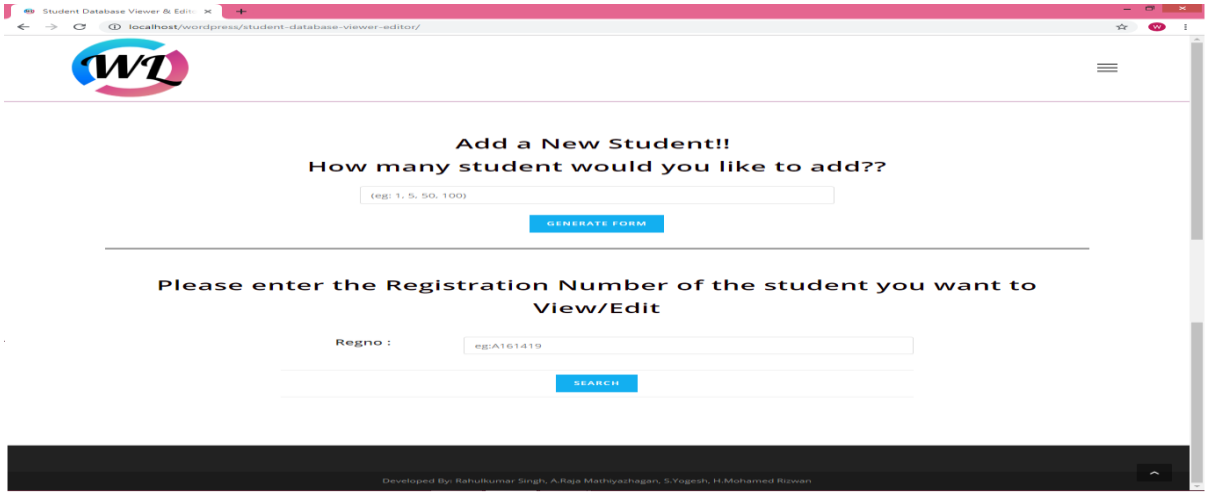


INSTITUTE INTERFACE

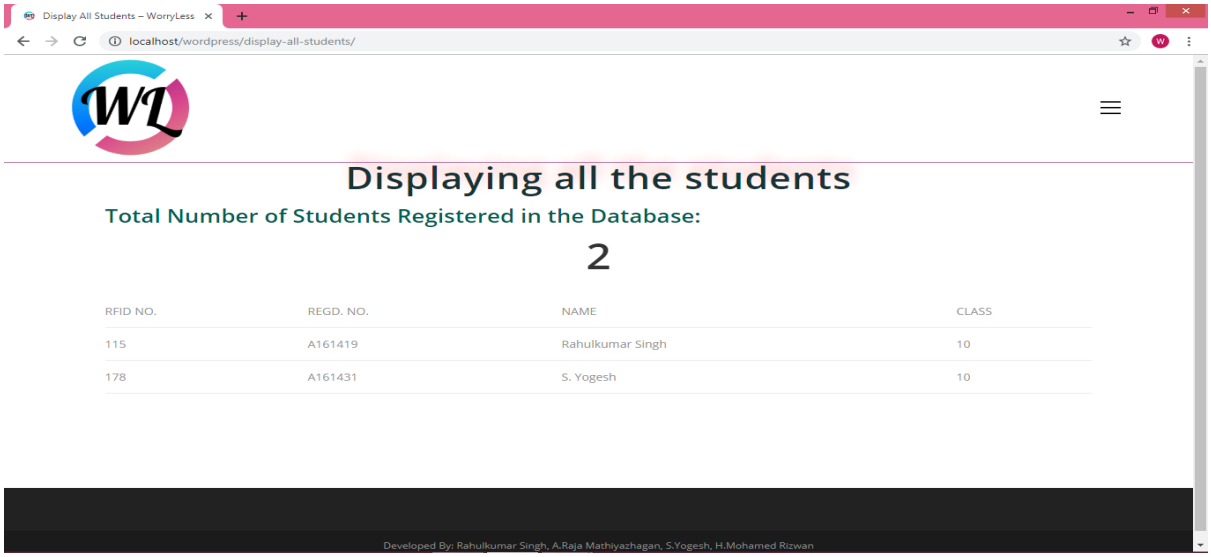
WORKING DAYS LOG



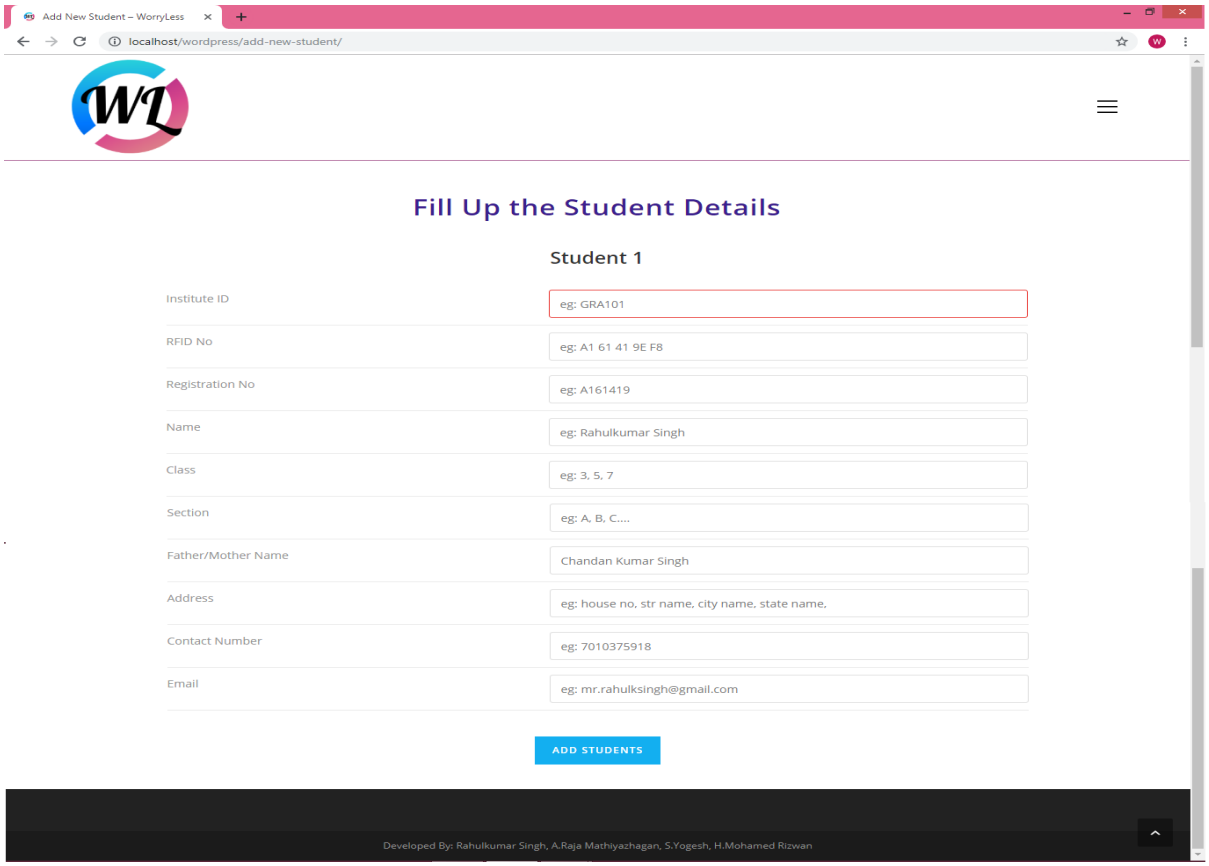
STUDENT DATABASE VIEWER AND EDITOR



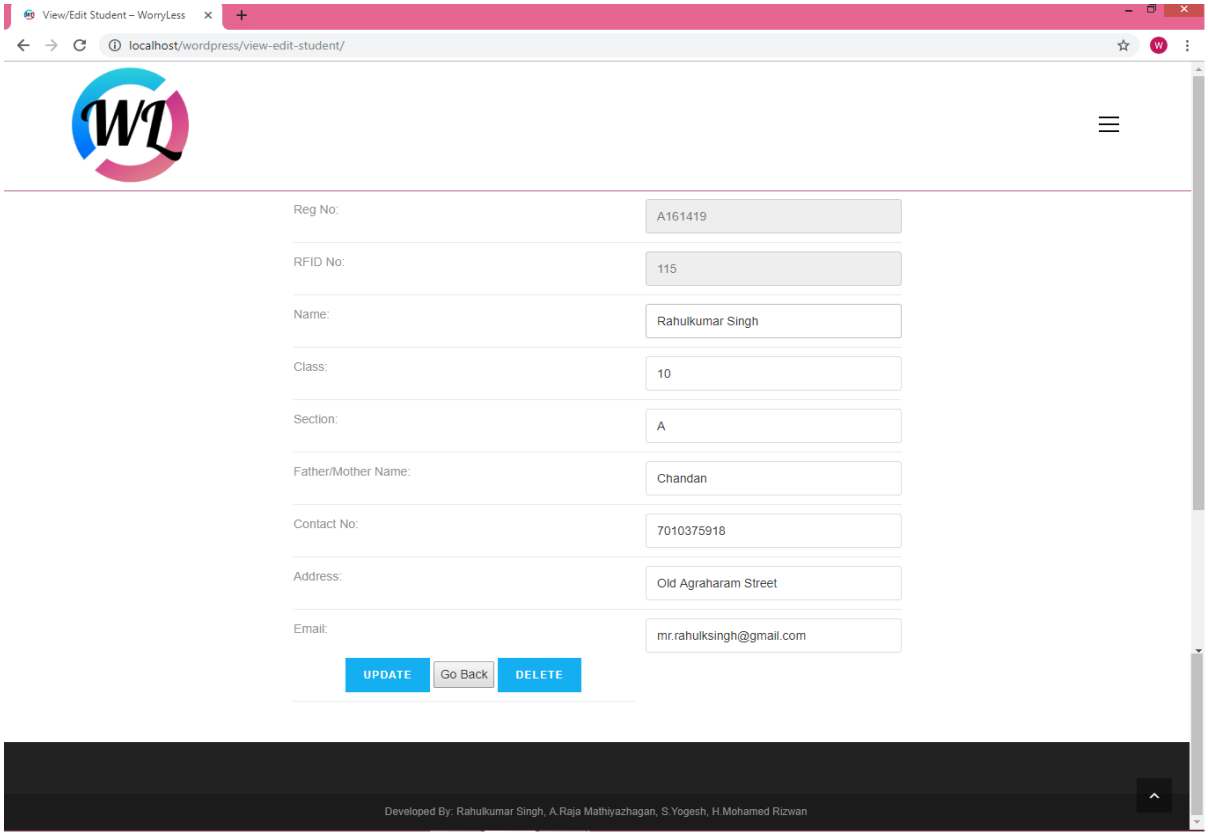
DISPLAYING ALL STUDENTS



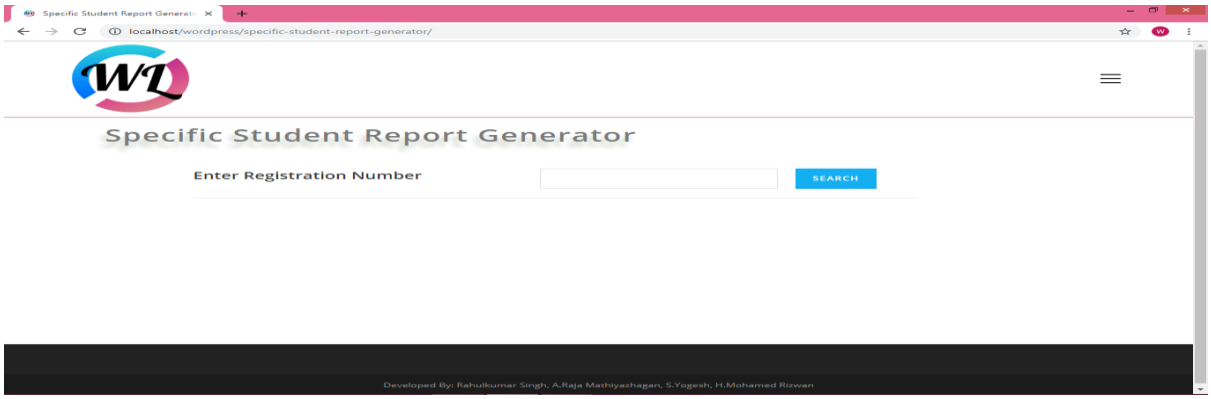
ADD NEW STUDENTS



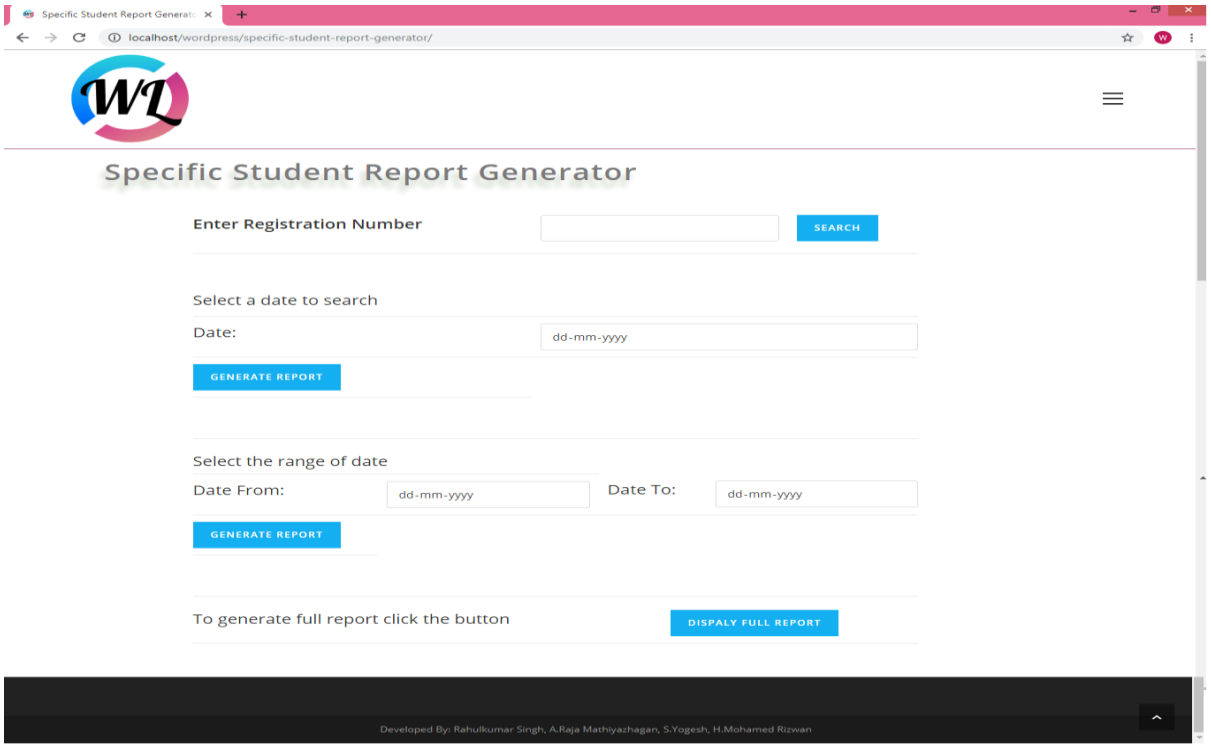
VIEW/EDIT STUDENTS



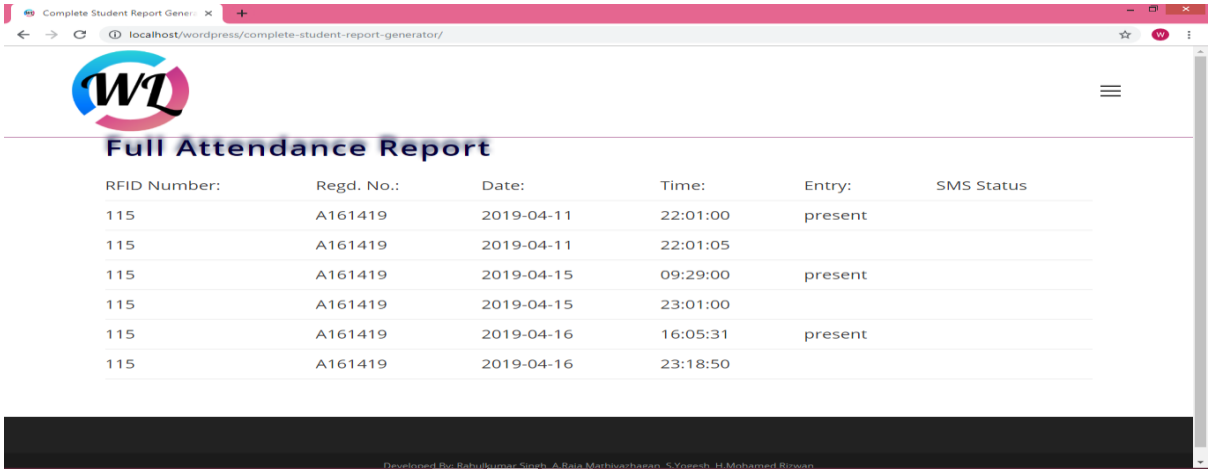
SPECIFIC STUDENT REPORT GENERATOR



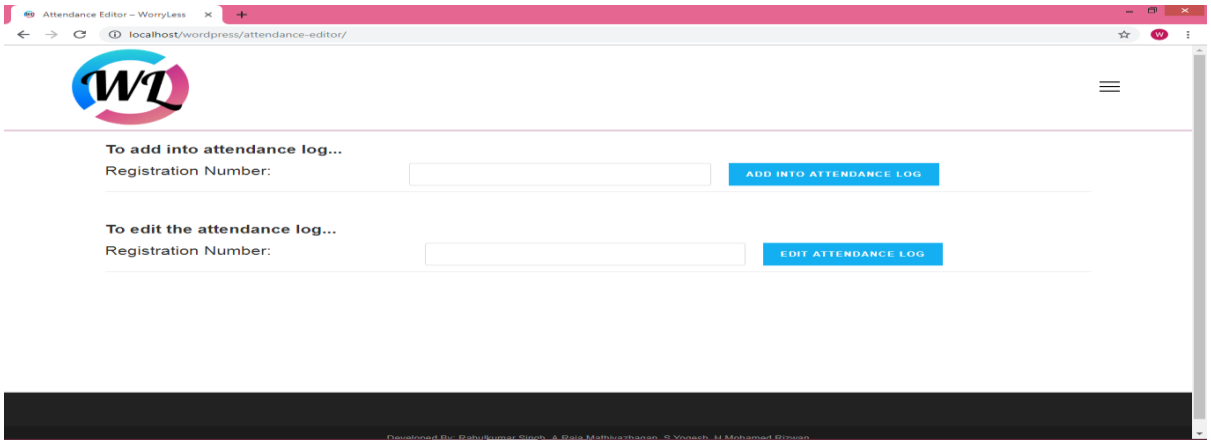
SPECIFIC STUDENT REPORT GENERATOR OPTIONS



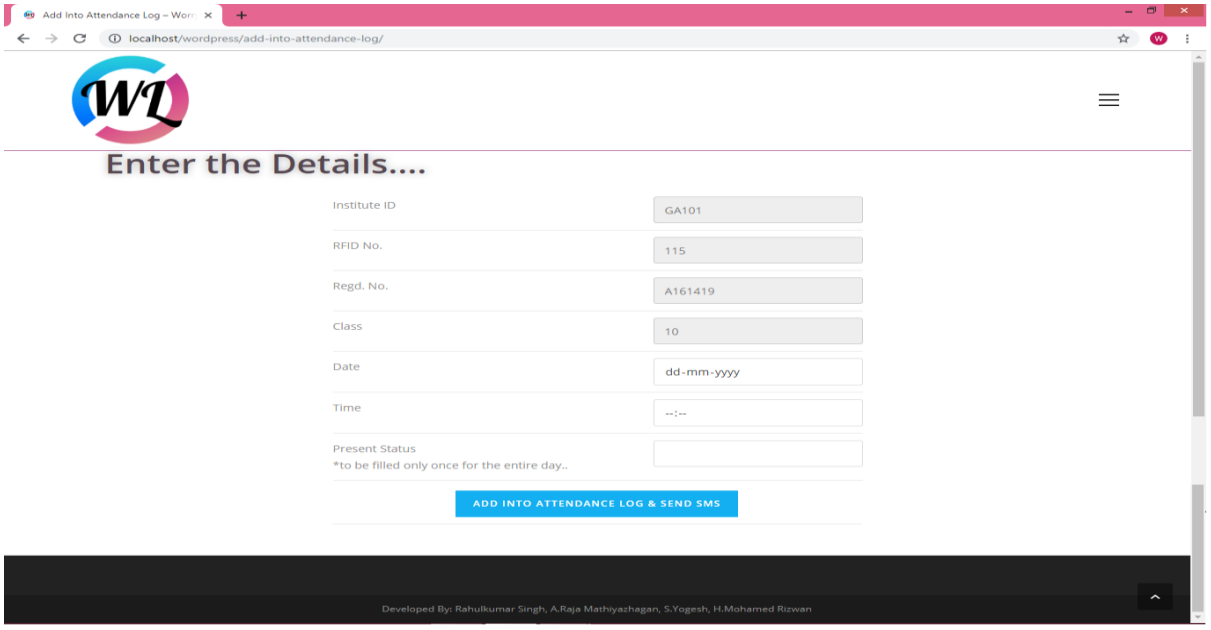
COMPLETE STUDENT REPORT GENERATOR



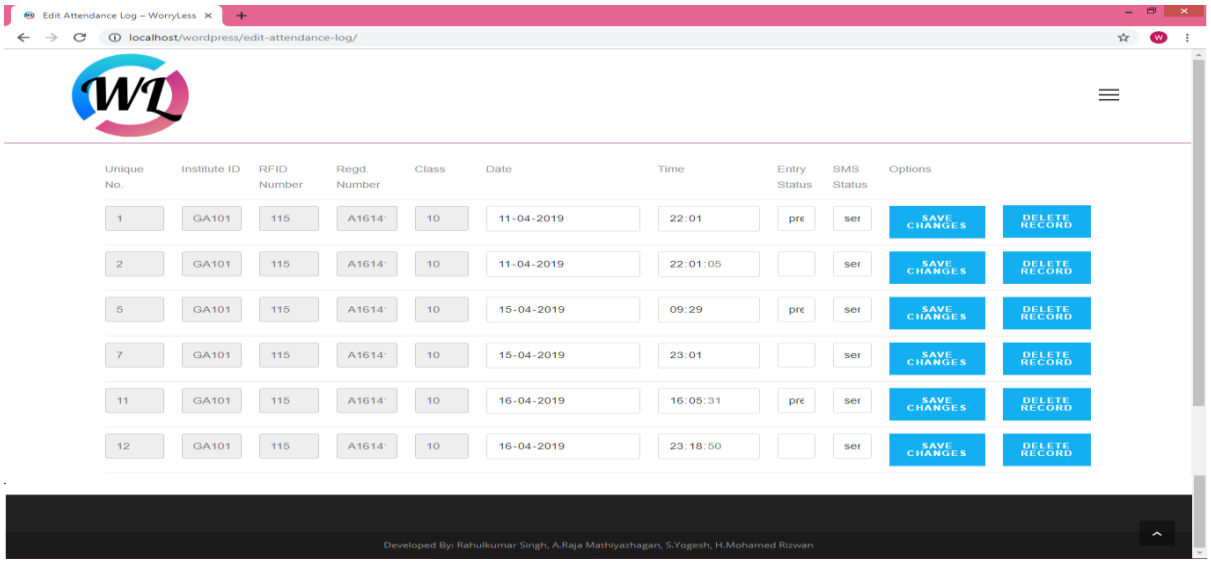
ATTENDANCE EDITOR



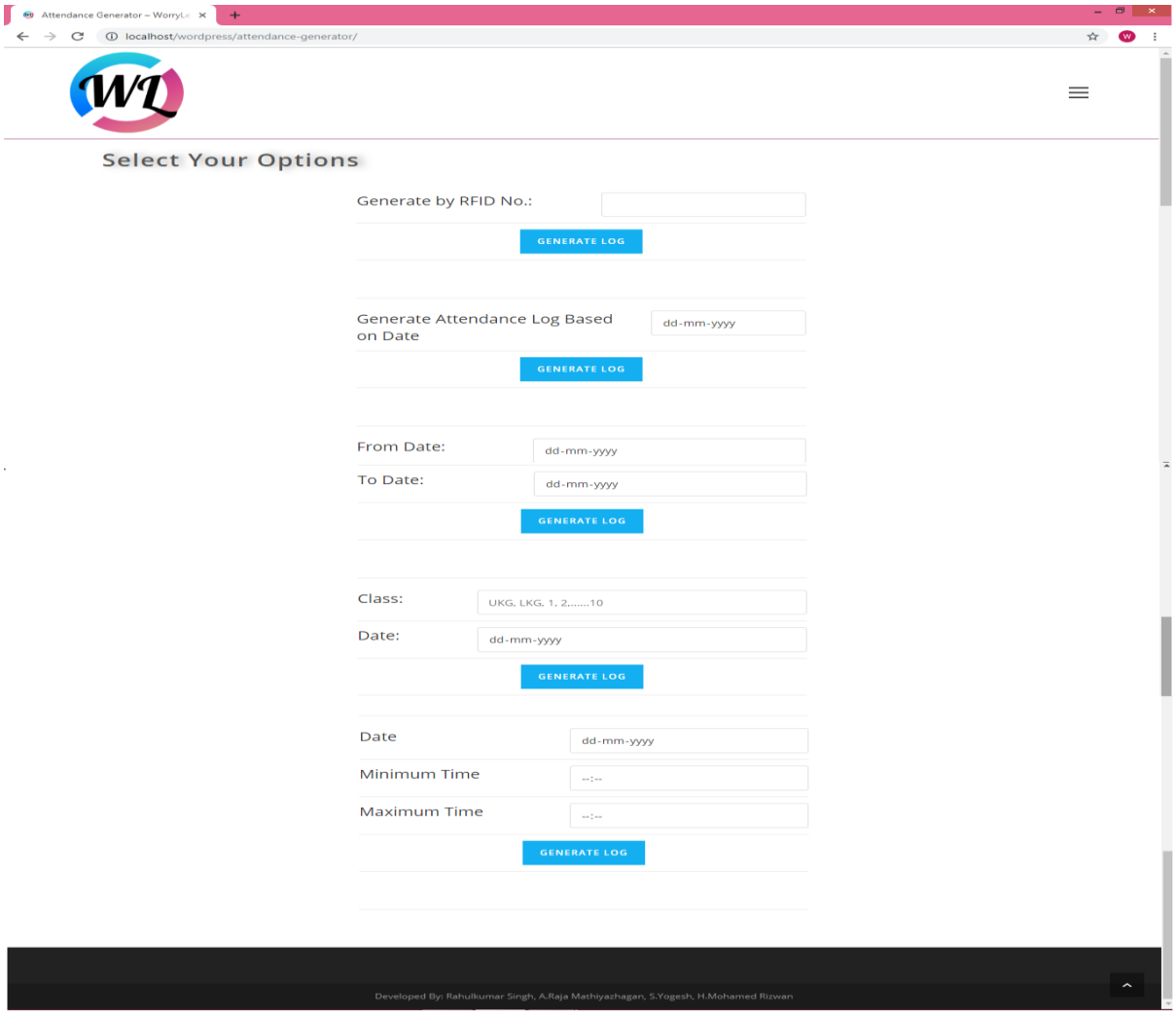
ADD INTO ATTENDANCE LOG



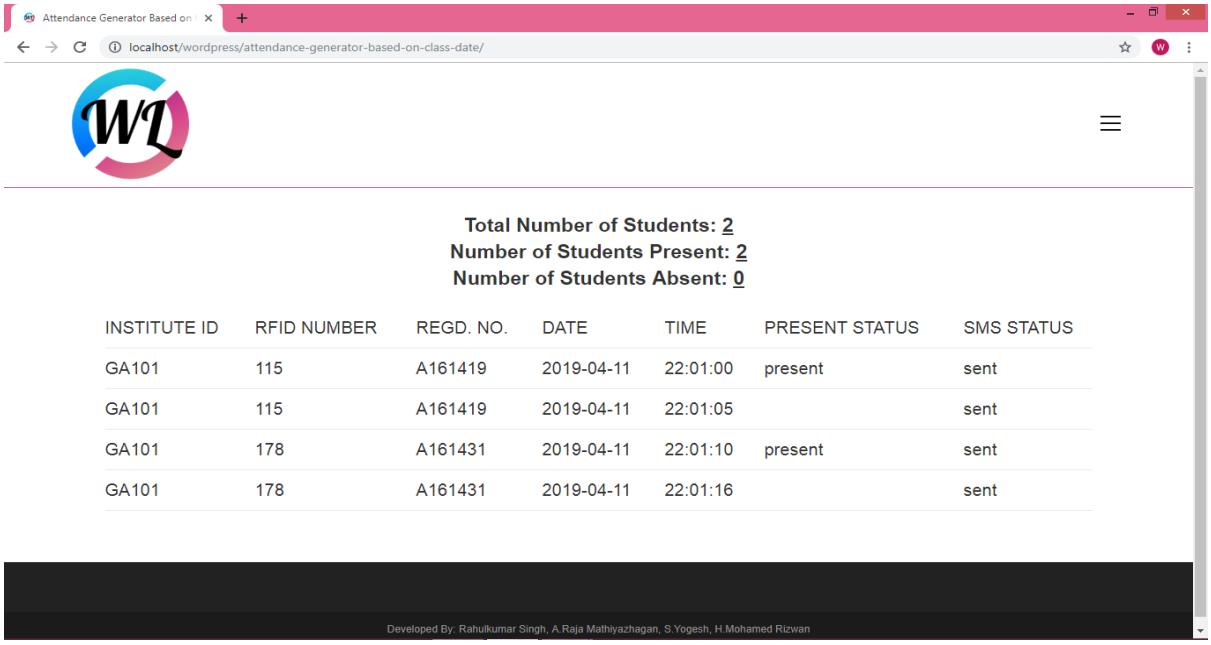
EDIT ATTENDANCE LOG



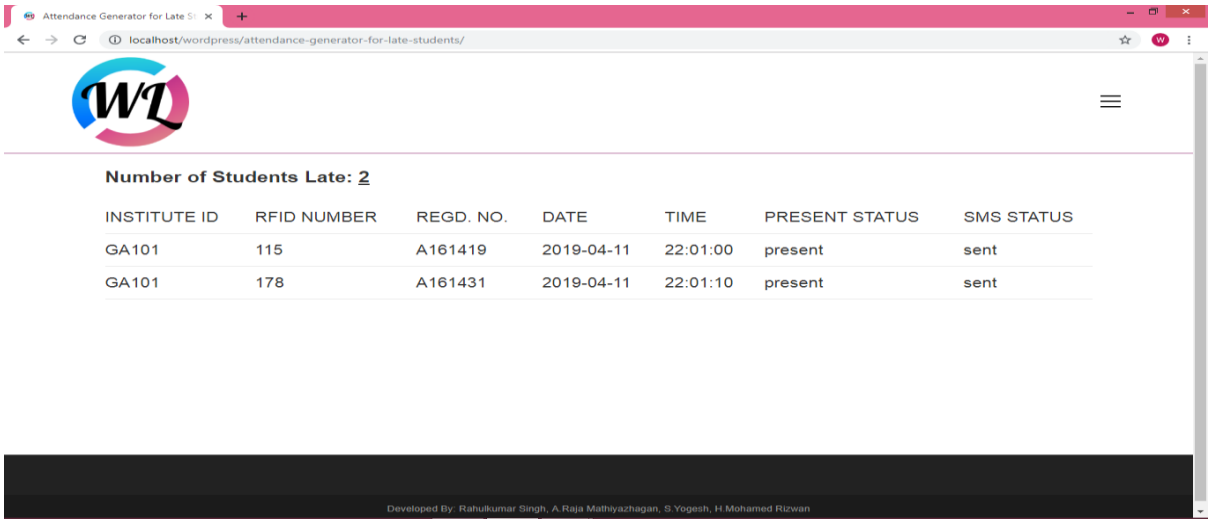
ATTENDANCE GENERATOR



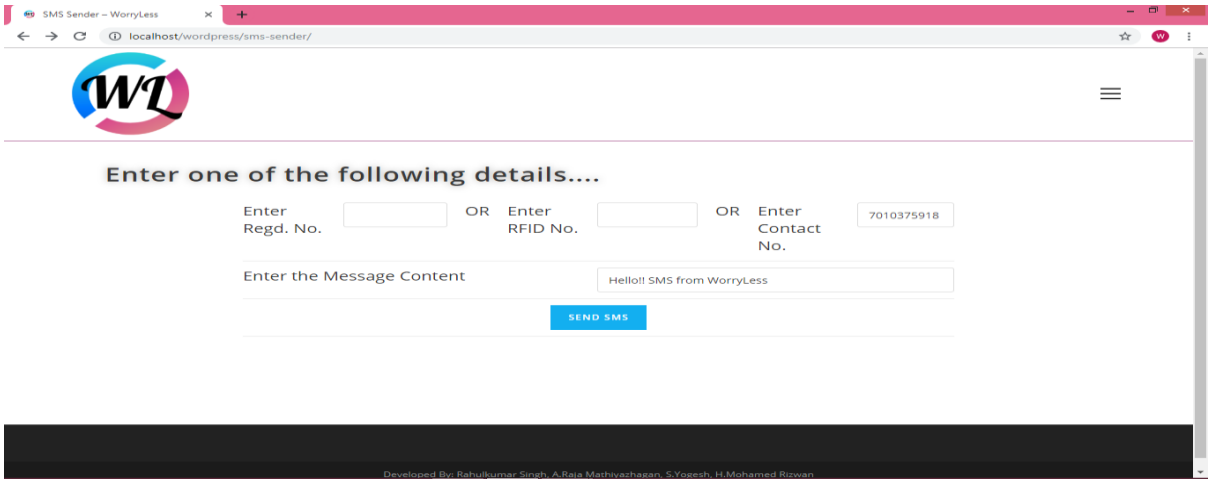
ATTENDANCE GENERATOR BASED ON CLASS AND DATE



ATTENDANCE GENERATOR BASED ON LATE STUDENTS OF CLASS

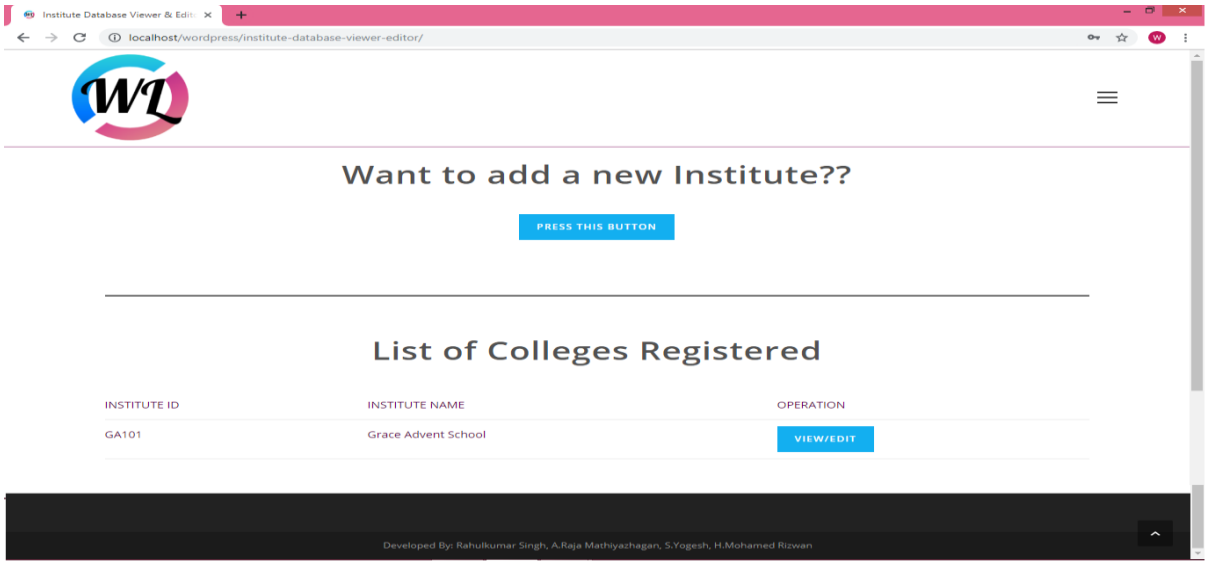


SMS SENDER

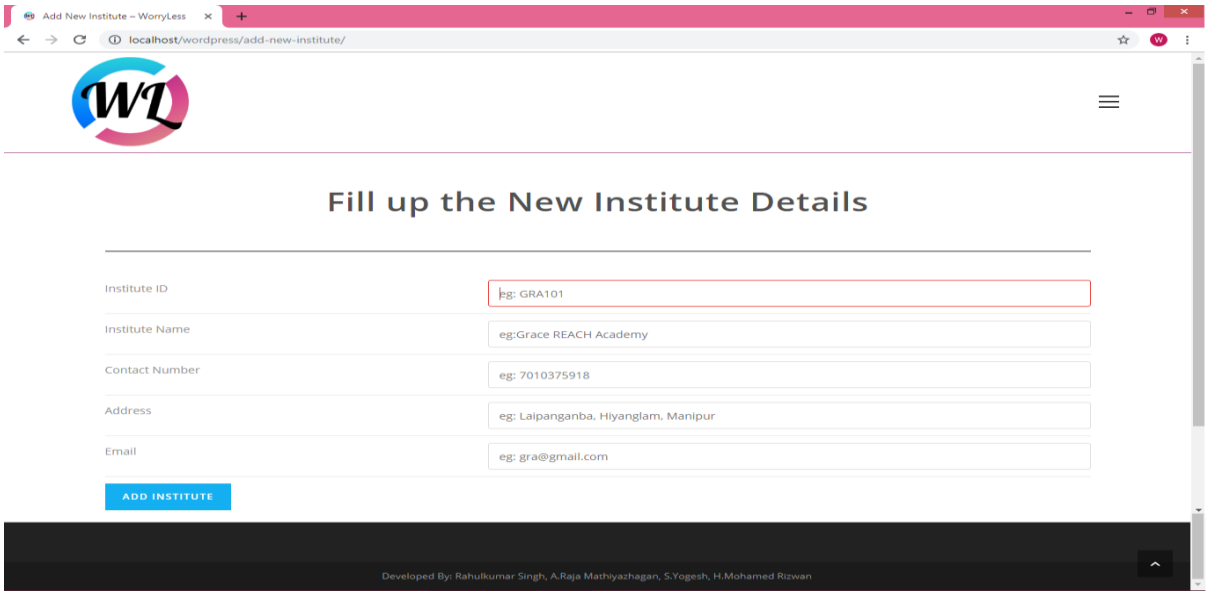


DEVELOPERS' INTERFACE

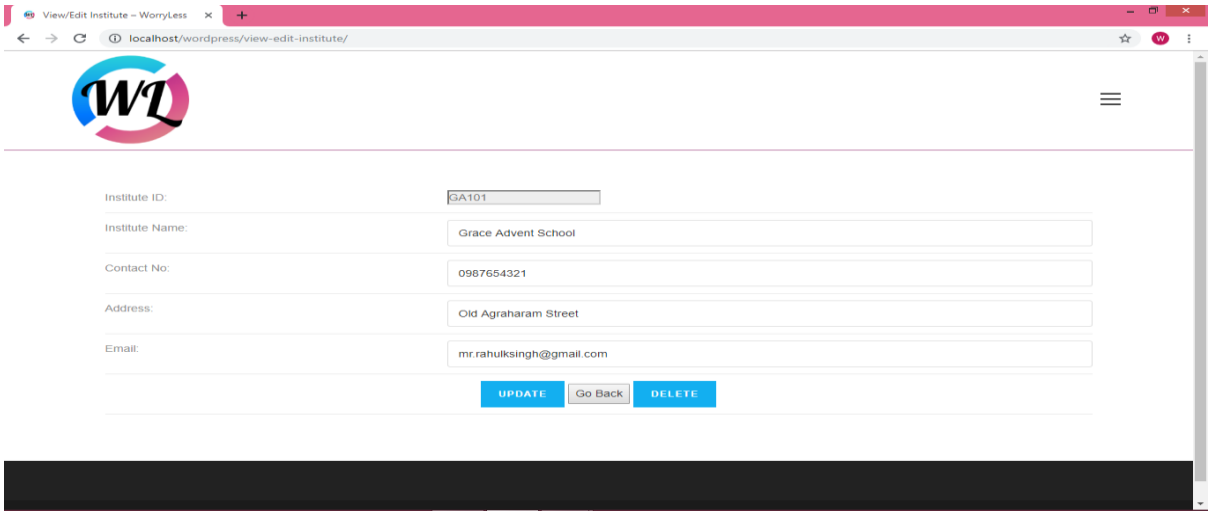
INSTITUTE DATABASE VIEWER AND EDITOR



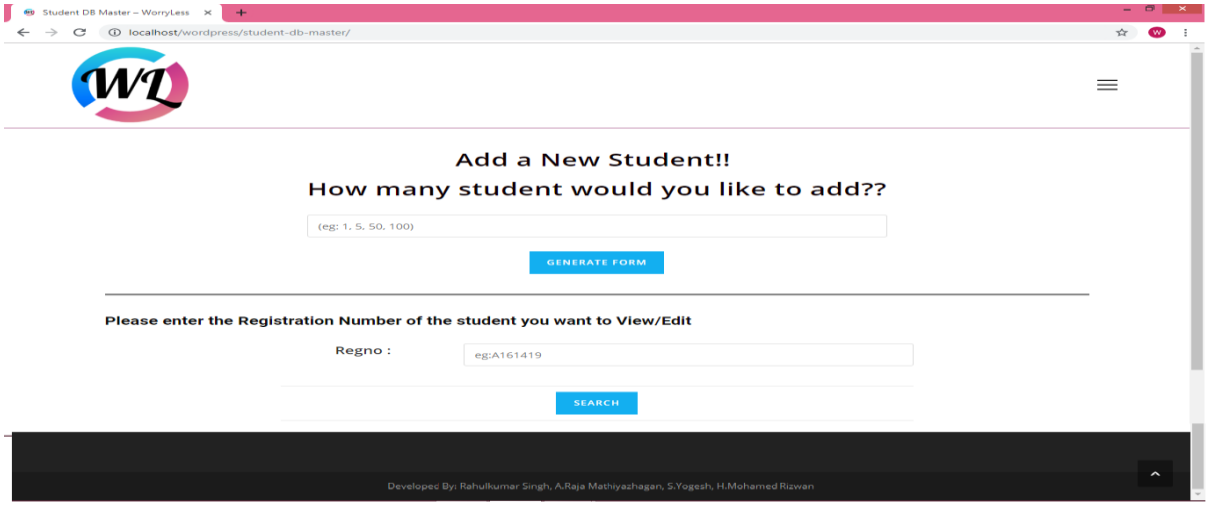
ADD NEW INSTITUTE



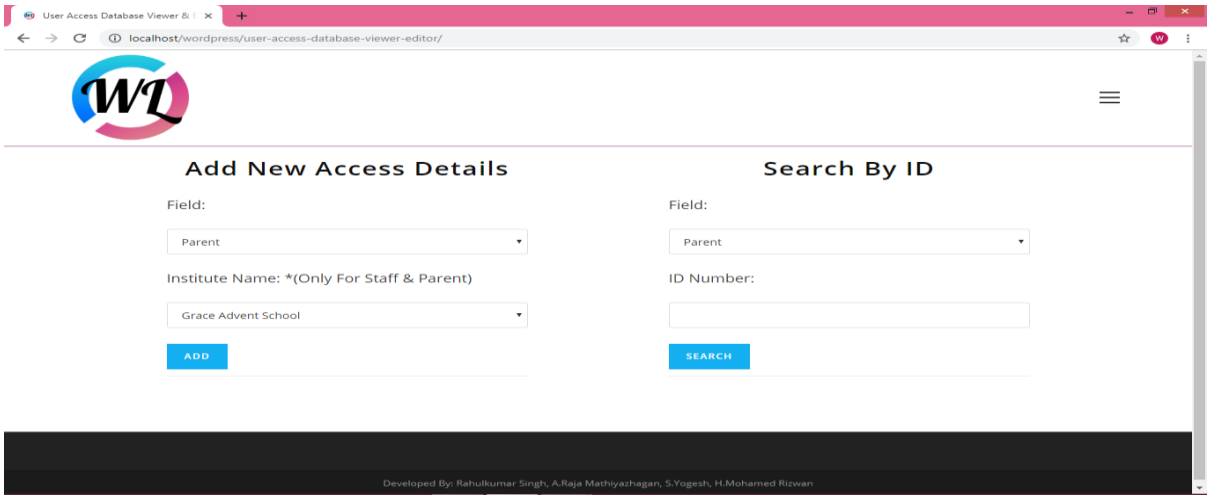
VIEW/EDIT INSTITUTE



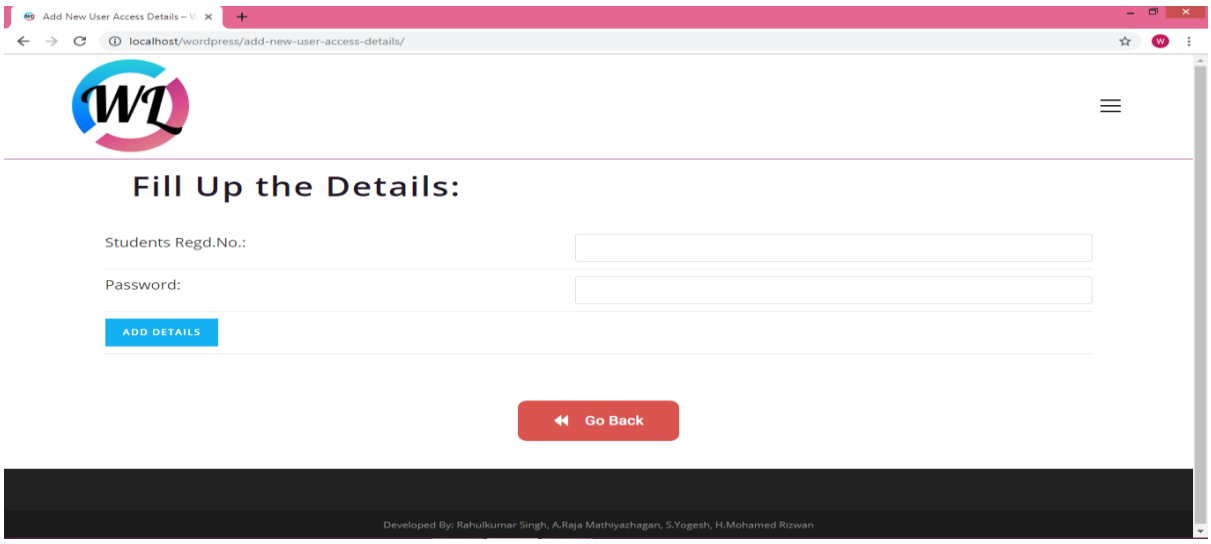
STUDENT DB MASTER



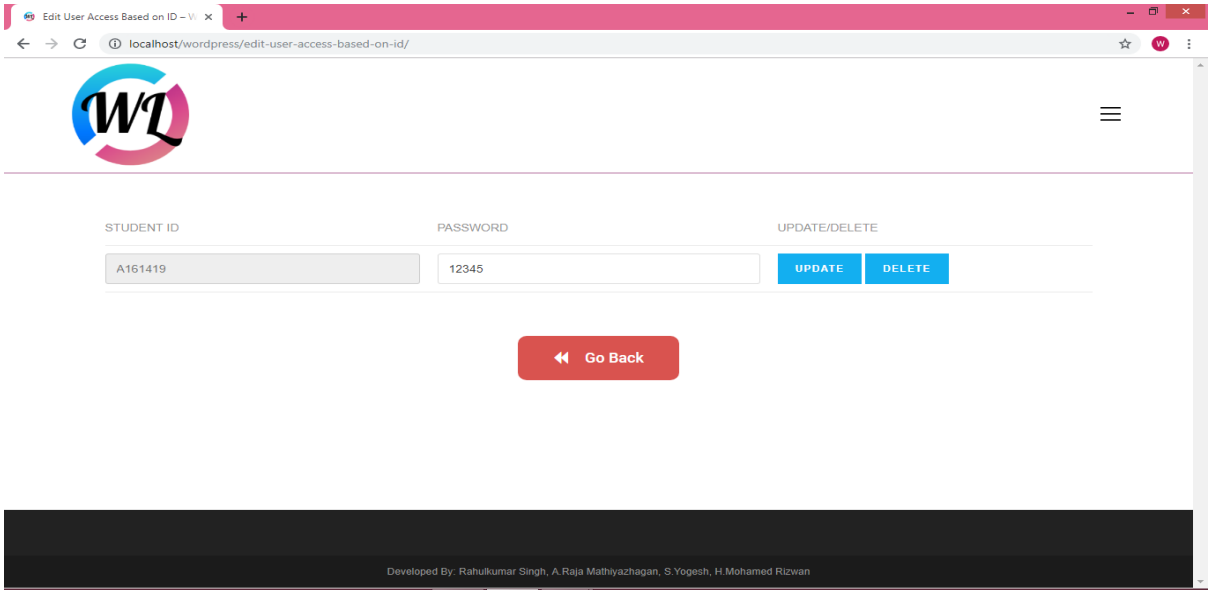
USER ACCESS DATABASE VIEWER AND EDITOR



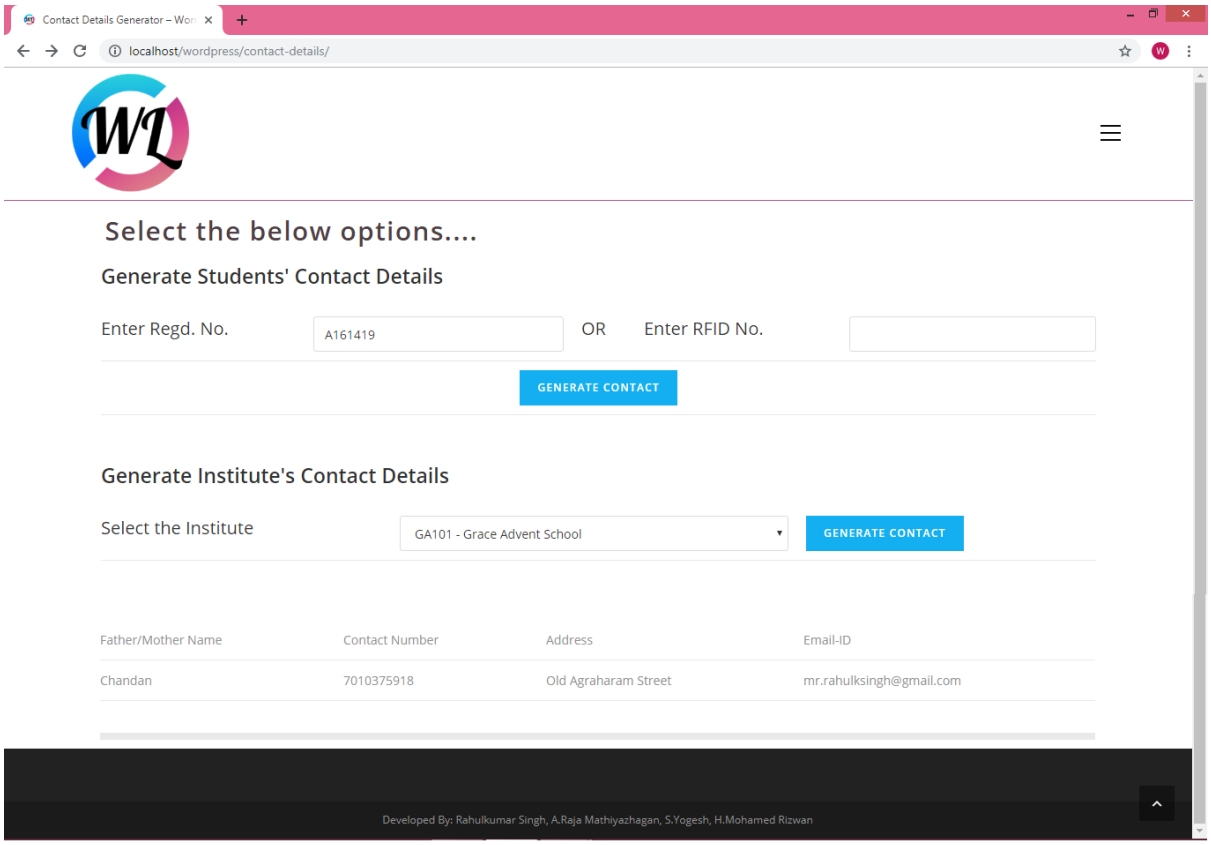
ADD NEW USER ACCESS DETAILS



EDIT USER ACCESS BASED ON ID



CONTACT DETAILS GENERATOR



CHAPTER 7
REFERENCES

[1] Steven Shepard- “RFID: Radio Frequency Identification”

[2] NodeMCU- https://www.nodemcu.com/index_en.html

[3] PHP- https://www.w3schools.com/php/

[4] Wordpress- https://www.youtube.com/watch?v=2cbvZf1jIJM

[5] NodeMCU and RFID- https://www.youtube.com/watch?v=nFMNccptJdU